



UNIVERSITAT POLITÈCNICA DE
CATALUNYA

PH.D. THESIS

Monocular Depth Estimation in Images and Sequences Using Occlusion Cues

Author:

Guillem Palou Visa

Advisor:

Philippe Salembier Clairon

Barcelona, November 2013

Abstract

Quan els humans observen una escena, son capaços de distingir perfectament les parts que la componen i organitzar-les espacialment per tal de poder-se orientar. Els mecanismes que governen la percepció visual han estat estudiats des dels principis de la neurociència, però encara no es coneixen tots els processos biològic que hi prenen part. En situacions normals, els humans poden fer servir tres eines per estimar l'estructura de l'escena. La primera és l'anomenada divergència. Aprofita l'ús de dos punts de vista (els dos ulls) i és capaç de determinar molt acuradament la posició dels objectes ,que a una distància de fins a cent metres, romanen enfront de l'observador. A mesura que augmenta la distància o els objectes no es troben en el camp de visió dels dos ulls, altres mecanismes s'han d'utilitzar. Tant l'experiència anterior com certs indicis visuals s'utilitzen en aquests casos i, encara que la seva precisió és menor, els humans aconsegueixen quasibé sempre interpretar bé el seu entorn. Els indicis visuals que aporten informació de profunditat més coneguts i utilitzats són, per exemple, la perspectiva, les oclusions o el tamany de certs objectes. L'experiència anterior permet resoldre situacions vistes anteriorment com ara saber quins regions corresponen al terra, al cel o a objectes.

Durant els últim anys, quan la tecnologia ho ha permès, s'han intentat dissenyar sistemes que interpretessin automàticament diferents tipus d'escena. En aquesta tesi s'aborda el tema de l'estimació de la profunditat utilitzant només un punt de vista i indicis visuals d'occlusió. L'objectiu del treball es la detecció d'aquests indicis i combinar-los amb un sistema de segmentació per tal de generar automàticament els diferents plans de profunditat presents a una escena. La tesi explora tant situacions estàtiques (imatges fixes) com situacions dinàmiques, com ara trames dins de seqüències de vídeo o seqüències completes. En el cas de seqüències completes, també es proposa un sistema automàtic per reconstruir l'estructura de l'escena només amb informació de moviment. Els resultats del treball son prometedors i competitius amb la literatura del moment, però mostren encara que la visió per computador té molt marge de millora respecte la presició dels humans.

Abstract

When humans observe a scene, they are able to perfectly distinguish the different parts composing it. Moreover, humans can easily reconstruct the spatial position of these parts and conceive a consistent structure. The mechanisms involving visual perception have been studied since the beginning of neuroscience but, still today, not all the processes composing it are known. In usual situations, humans can make use of three different methods to estimate the scene structure. The first one is the so called divergence and it makes use of both eyes. When objects lie in front of the observed at a distance up to hundred meters, subtle differences in the image formation in each eye can be used to determine depth. When objects are not in the field of view of both eyes, other mechanisms should be used. In these cases, both visual cues and prior learned information can be used to determine depth. Even if these mechanisms are less accurate than divergence, humans can almost always infer the correct depth structure when using them. As an example of visual cues, occlusion, perspective or object size provide a lot of information about the structure of the scene. A priori information depends on each observer, but it is normally used subconsciously by humans to detect commonly known regions such as the sky, the ground or different types of objects.

In the last years, since technology has been able to handle the processing burden of vision systems, there has been lots of efforts devoted to design automated scene interpreting systems. In this thesis we address the problem of depth estimation using only one point of view and using only occlusion depth cues. The thesis objective is to detect occlusions present in the scene and combine them with a segmentation system so as to generate a relative depth order depth map for a scene. We explore both static and dynamic situations such as single images, frame inside sequences or full video sequences. In the case where a full image sequence is available, a system exploiting motion information to recover depth structure is also designed. Results are promising and competitive with respect to the state of the art literature, but there is still much room for improvement when compared to human depth perception performance.

Agraïments

Voldria començar donant l'agraïment al meu tutor Philippe ja que, sense els seus consells, les seves idees i recomanacions, aquesta tesi no hagués estat possible. Durant aquest temps, les discussions amb ell, sempre fructíferes, m'han servit per aprendre nous conceptes i per a saber escollir la millor opció. Per això vull agrair-li la seva paciència, la seva llibertat i sobretot la seva exigència en cada moment.

M'agradaria també fer especial menció al Grup de Processat d'Imatge i a tots els seus integrants en conjunt ja que han recolzat totes les activitats realitzades durant la tesi. També donar les gràcies als companys del despatx per fer aquest període més amè i distret.

Especial menció a la meva família, que amb tot el seu suport i dedicació m'han ajudat a tirar endavant tots aquest anys. Finalment, vull agrair de tot cor a la meva parella Alba la seva dedicació, la seva comprensió i afecte durant tot aquest temps i els que vindran.

A tots, moltes gràcies!

Contents

I	Introduction	11
1	Introduction	13
1.1	Motivation	13
1.2	Research Contributions	14
1.3	Thesis Organization	16
2	Depth Perception	17
2.1	Vision: the Early Process	17
2.1.1	The visual system	17
2.1.2	Visual perception	19
2.2	Depth Perception in Humans	24
2.2.1	Multiple View Depth Cues	25
2.2.2	Monocular Depth Cues	26
2.2.3	Dynamic Depth Cues	35
2.2.4	General Cue Combination	36
2.3	Depth Cue Perception in Computer Vision: Proposed Approach	37
II	Monocular Depth Estimation from Occlusion Cues	39
3	Monocular Occlusion Cues	41
3.1	Depth Cues in Static Images	41
3.1.1	T-junction estimation	43
3.1.2	Convexity estimation	53
3.1.3	Probability of ownership - Combining T-junction and Convexity cues	56
3.2	Depth Cues in Dynamic Scenes	59
3.2.1	Optical Flow Estimation	61
3.2.2	Motion Occlusions	65
4	Evaluation Methodology	73
4.1	Integrating detection and classification problems	73
4.1.1	Detection Problems	74
4.1.2	Combining Detection with Binary Classification	76

4.2	Two PRC frameworks on Depth Ordering	78
4.2.1	Local Depth Consistency	78
4.3	Global Depth Consistency	82
5	Depth Ordering in Still Images	87
5.1	State of the Art	87
5.2	Hierarchical Representation of Images	94
5.2.1	State of the Art	94
5.2.2	The Monocular Depth BPT	97
5.2.3	Ultrametric Contour Maps	106
5.3	Tree Cuts	111
5.3.1	A 0-1 Integer Programming Approach	112
5.3.2	Equivalent Graph Cut Problem	114
5.3.3	Tree cuts for general binary trees	116
5.3.4	Dynamic Programming Algorithm	117
5.3.5	Are Tree Cuts Useful?	118
5.3.6	Depth-based Tree Cut	123
5.4	Depth Ordering	126
5.4.1	Occlusion Based Tree Cut	126
5.4.2	The Depth Order Graph	129
5.5	Results	136
5.5.1	Depth Annotated Dataset	136
5.5.2	Quantitative evaluation	137
5.5.3	Qualitative evaluation	141
6	Depth Ordering in Single Frames of Video Sequences	149
6.1	State of the Art	149
6.2	Hierarchical Representation of Frames	153
6.2.1	State of the Art	153
6.2.2	Proposed BPT for Frames	154
6.2.3	The UCM for Frames	156
6.3	Depth Ordering	157
6.3.1	Tree Cut for Parametric Flow Fitting	157
6.3.2	Occlusion relation estimation	158
6.3.3	The Depth Order Graph for Frames	159
6.3.4	Final Depth Ordering	159
6.4	Results	160

6.4.1	Quantitative Evaluation	160
6.4.2	Qualitative Evaluation	163
7	Depth Ordering of Video Sequences	167
7.1	State of the Art	167
7.2	Hierarchical Representation of Video Sequences	172
7.2.1	State of the Art	172
7.2.2	Proposed Hierarchical Video Representation	175
7.2.3	Trajectory Estimation	176
7.2.4	Trajectory Binary Partition Tree	179
7.2.5	Results on Early Segmentations	184
7.3	Relative Depth Ordering	189
7.3.1	Tree Cut for Motion Segmentation	190
7.3.2	Occlusion relations for video	192
7.3.3	Depth ordering	192
7.3.4	Results	193
8	Structure from Motion	201
8.1	Structure from General Motion	202
8.1.1	Pinhole Camera Model and Homogeneous coordinates	203
8.1.2	Reliable trajectory tracking	206
8.1.3	Two View Geometry	207
8.1.4	Bundle Adjustment	210
8.1.5	Incremental Projective Structure Recovery	210
8.1.6	Autocalibration for Metric Reconstruction	212
8.1.7	Depth-map post-processing	216
8.2	Results	219
8.2.1	State of the Art Comparison	219
8.2.2	Groundtruth Comparison on Stereo	221
8.2.3	Limitations	221
III Conclusions		223
9	Conclusions	225
9.1	Limitations	226
10	Open Problems and Future Research	227

CONTENTS

11 List of Publications	228
IVReferences	231
References	233

Part I
Introduction

1 Introduction

1.1 Motivation

When humans face different kind of scenes, they can easily estimate the scene structure even if only one point of view is available. In monocular situations, knowing which objects are in front of the others is an obvious task, even if the scene is new to the observer. Reconstructing the underlying visual process has been a field of study since many years, formally starting with (Von Helmholtz 1866). In the subsequent years, authors focused on particular aspects of depth perception, such as which cues are for humans the most significant for depth detection (Braunstein et al. 1989; Ono et al. 1986).

Monocular depth ordering systems are currently an active field of research in the image processing and computer vision field. While most of the works assume some image structures, there are very few that perform the depth ordering task using only low level cues. The proposed system is motivated by the work of (Dimiccoli 2009), tackling the problem without prior knowledge of the type of scene. For this reason, only still images occlusion cues are used, such as T-junctions and convexity. Both cues offer good signs of depth discontinuities (McDermott 2004), although only the relative depth between regions can be retrieved. For image sequences other kind of cues can also be used, such as motion occlusions or motion parallax. As with still cues, motion occlusion only offer signs on the relative depth order, while motion parallax can be used to retrieve a full depth map, with absolute values (up to some factor).

The knowledge of relative depth can be useful in many applications, such as occlusion boundary detection or depth plane segmentation. Moreover, the system can even be used as a first step to retrieve a full depth map, as done by the film industry. Therefore, the main objectives of this project is to provide a system able to retrieve the relative depth either on monocular single images or sequences relying only on local depth cues. Additionally, in image sequences, the absolute depth could also be retrieved by using structure from motion.

Many vision theories such as (Von Helmholtz 1866) state that the human perception is a Bayesian inference process. To adequate the proposed algorithm to existing vision theories, our work states a probabilistic framework to detect low level and order image/video regions according to depth. Moreover, human scene interpretation is known to be a cooperative process of smaller problems such edge detection, texture recognition, cue inference, etc. For this reason, a hierarchical image/video represen-

tation is built, so the algorithm is able to deal with details appearing at multiple resolutions. To estimate relative depth, local depth cues are aggregated and a global inference is computed so as to provide with consistent relative depth order maps.

1.2 Research Contributions

In this thesis the depth ordering/estimation problem is addressed in single images, single frames and video sequences. The main contributions can be found below, organized in fields and mentioning the associated publications, either published or in peer review process in the date of the writing of the document (November 2013).

Contribution in Evaluation Methods To the date, evaluation methods on depth ordering were rather inadequate and lacked rigorousness. As the depth ordering problem encompasses a first image/video segmentation step, most state of the art algorithms decoupled both steps, computing two independent performance measures to each one. In this thesis we propose a unified framework for problems that include both detection (segmentation) and classification (depth ordering), showing that both steps are strongly correlated. Related publications:

- G. Palou and G. Salembier. "Precision-Recall-Classification Evaluation Framework: Application to Depth Estimation on Single Images". In: *Submitted to CVPR*. 2014

Contributions in Single Images The proposed approach for single images aims to push the limit of low level static cues, without supposing any a priori structure of the scene. To this end, in this thesis a new method to estimate junctions based on segmentation information is proposed. Moreover, low level depth cue estimation is integrated with the region merging process, introducing depth information to the segmentation hierarchy. Additionally, local information provided by depth cues is propagated to infer a global consistent depth map with a new probabilistic framework using concepts of reliability networks. Related publications:

- G. Palou and P. Salembier. "Occlusion-based depth ordering on monocular images with Binary Partition Tree". In: *IEEE ICASSP*. Prague, Czech Republic, 2011
- G. Palou and P. Salembier. "From local occlusion cues to global depth estimation". In: *IEEE ICASSP*. Kyoto, Japan, 2012

- G. Palou and P. Salembier. "Monocular Depth Ordering Using T-junctions and Convexity Occlusion Cues." In: *IEEE Trans. on Image Proc.* 2013

Contributions in Single Frames Depth ordering on frames is performed using dynamic low level depth cues such as motion occlusions. In this thesis, a new way of estimating motion occlusions is designed by integrating segmentation in the detection process. Moreover, motion information is introduced to the segmentation process, improving the region quality shown by quantitative measures. Related publications:

- G. Palou and P. Salembier. "2.1 Depth Estimation of Frames in Image Sequences Using Motion Occlusions." In: *ECCV Workshops*. Firenze, Italy, 2012
- G. Palou and P. Salembier. "Depth ordering on image sequences using motion occlusions". In: *IEEE ICIP*. Orlando, FL, USA, 2012
- G. Palou and P. Salembier. "Depth order estimation for video frames using motion occlusions". In: *IET Computer Vision* 2013

Contributions in Video Sequences One of the major contributions for video sequences is the design of a hierarchical region representation based on color and long term motion information. As with single frames, motion occlusions are used to order objects by their relative depth. Related publications:

- G. Palou and P. Salembier. "Hierarchical Video Representation with Trajectory Binary Partition Tree". In: *IEEE CVPR*. Portland, OR, USA, 2013
- G. Palou and P. Salembier. "Hierarchical Video Representation with Trajectory Binary Partition Tree and its Applications". In: *IEEE TPAMI*, in peer review 2013

Another contribution, is the design of an algorithm that uses motion information to recover absolute depth maps for video sequences of static sequences with arbitrary camera motion. The goal of introducing structure from motion is to introduce a possible approach that allows us to merge motion occlusions (relative depth) and structure computation (absolute depth). This will permit to construct dense maps for sequence with arbitrary moving objects, and not only static scenes. Structure from motion is an active field in the literature and can, by itself alone, provide results on depth map generation in certain types of sequences. Therefore, a system using only optical flow to recover depth is designed.

1.3 Thesis Organization

This thesis can be divided in eight major chapters, a part from the current introduction in Chapter 1 and the conclusions in Chapter III.

Depth Perception In Chapter 2 the fundamentals of human vision are reviewed and how they can be related with the computer vision and image processing field. The depth perception process and visual interpretation of scenes is reviewed in this part, giving an insight of possible computational approaches to mimic human vision. Additionally, the general system architecture is described.

Monocular Occlusion Cues The whole chapter 3 is devoted to explain the types of low level depth cues that the proposed depth ordering approaches use. Both static and dynamic cue estimations are shown.

Evaluation Methodology Prior to deal with the depth ordering problem, a framework to evaluate the results is proposed in Chapter 4. Evaluation is a key aspect to assess if obtained results are competitive with the state of the art.

Depth Ordering in Still Images In Chapter 5 the problem of depth ordering in single images is tackled. In this chapter, a deep review of image region representation is shown, and well as their uses to the depth ordering problem.

Depth Ordering in Single Frames of Video Sequences Extending the algorithm for single images to frames inside video sequences is done in Chapter 6. Both systems are similar, but in frames dynamic low level cues can be used instead of only the static ones. To this end, the different steps of the algorithm are particularized to use motion information.

Depth Ordering of Video Sequences Similar to the previous two chapters, in Chapter 7 an automated system is designed to retrieve relative depth for video sequences.

Structure from Motion in Video Sequences Adopting a different perspective than the other chapters, in Chapter 8 an algorithm that recovers full depth maps using motion information is designed. This algorithm will be used as a preliminary step to merge motion occlusions and structure from motion. In this sense, this chapter can be viewed as as an ongoing work extending the technological core of this thesis.

2 Depth Perception

2.1 Vision: the Early Process

Humans are known for their ability to recognize objects and determine the scene structure in many distinct situations. Our capacity to retrieve a coherent depth interpretation of the environment seems to be robust and reliable in the majority of cases (Epstein and S. Rogers 1995), with the exception of some optical illusions. The ability to perceive a 3D world in humans is mainly due to binocular vision, where each eye provides a different image of the scene and disparity is subconsciously inferred. However, in monocular situations, perception is affected but still, depth information can be perceived, see Fig. 2.1 for which cues are used for different types of viewing distances. The scientific community has tried to mimic the human behavior to determine the depth structure of scenes. To this day, human performance is still much better than computer based approaches in both time and accuracy, but the evolution of 3D visualization hardware encourages researchers devote efforts to estimate depth from visual content.

In this section a review of human vision is given, with special focus on depth perception. The vision community has been studying how light photons are combined to produce images in the eye, how these images are transferred to the brain and how this information is transformed to sensorial or semantic data to condition human behavior. Although the complete mechanisms are not fully known, computer algorithms rely on vision studies to base their reasonings on vision theories. The first part in Sec. 2.1.1 discusses the biological mechanisms behind image formation, comparing it to modern photograph cameras. The second part Sec. 2.1.2 gives an insight in current perception theories and how they influence the development of algorithms. Depth perception is discussed more deeply in Sec. 2.2.

2.1.1 The visual system

Given a three dimensional scene, the key problem (prior to scene understanding) is to know how points and regions in space are transformed into point and regions on the image plane. The role of the (human) eye is to collect environment light, regulate its intensity, focus it through a series of lenses and produce an image. Images formed in both eyes will then be sent using electrical signals to the brain for their interpretation. This process is well known, and its working principle is very similar in species with

2. DEPTH PERCEPTION

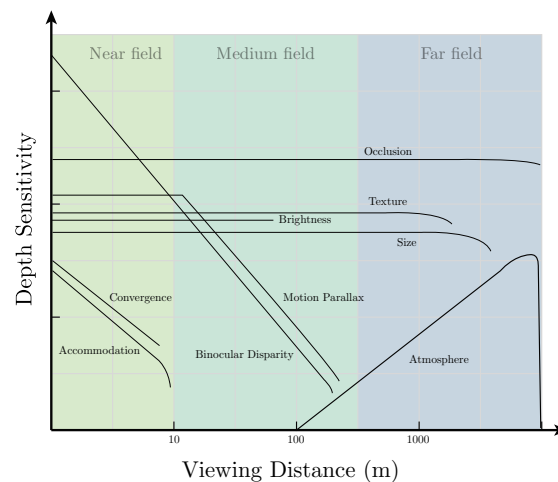


Figure 2.1: Graphic adapted from (Nagata 1991) showing the sensitivity in depth for the different types of cues. Note that monocular depth cues are valid in all near, medium and far fields. Disparity, although much more reliable for near viewing distances, has a limited range of action.

complex eyes (96% of living organisms (Fernald 1997)). There exist many types of complex eyes, and each one of them is adapted to the biological need of each species. For instance, the number of lenses, the position of eyes in the head or the operating band of photoreceptors can be substantially different between types of eyes.

Concerning the human case, shown in Fig. 2.2, when rays of light enter through the cornea and the iris, they are distorted by the lens such that they become focused on the retina. Focusing is needed so as to produce sharp images regardless on the distance of the external objects. When light hits the retina, two types of photoreceptor cells (rods and cones) get excited depending on light intensity and frequency. Cones respond to bright light and high resolution color vision, while rods respond to monochromatic vision in very deam light environments. The sensitivity of the cones in several wave-lengths is what provides humans the capacity to see colors. Cones are divided into subgroups selective to short, medium and large wavelengths. Due to this three subgroups, humans are able to distinguish three colors (commonly names red, green and blue) but many other animals have normally two or sometimes four subgroups.

The level of excitement of photohoreceptor cells is sent to the brain through the optical

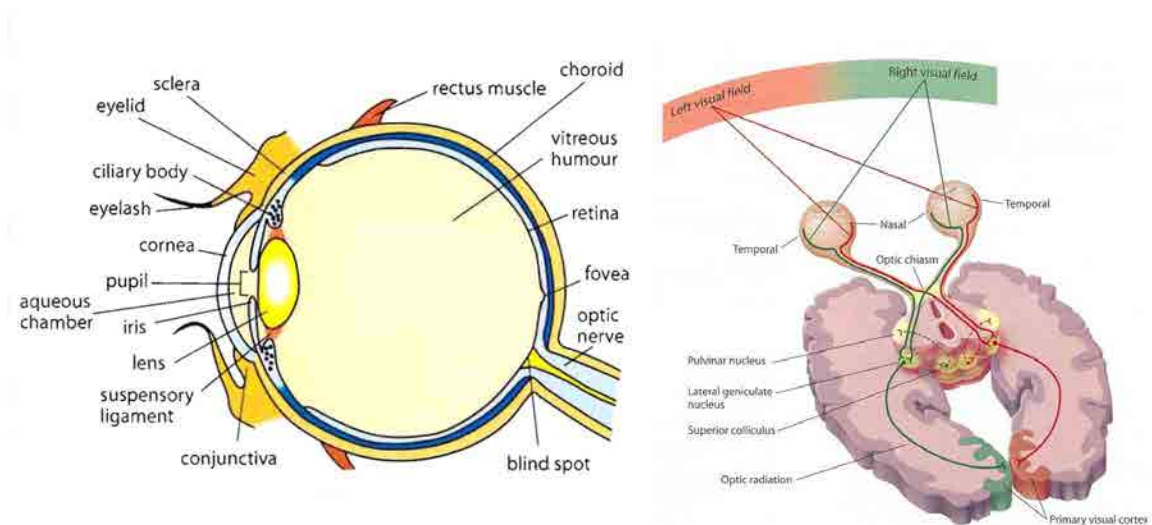


Figure 2.2: Vertical section of the human eye in the left, and an overview of the human visual system on the right

nerve, which is then responsible to form one image for each eye. Information paths are divided between the left and right visual tracks, and each track carries the representation from half the visual field (Wurtz et al. 2000) and they meet on the visual cortex, the back part of the brain responsible for visual perception. While images travel through optic tracks, the image is processed in simple ways by neurons so extra information to the visual cortex is provided. This low level processing is then used to provide higher level information such as disparity (depth) estimation or pattern recognition. This process is also known as visual perception and contrarily to the visual system, the way in which the brain perceives remains still unknown.

2.1.2 Visual perception

Research on how and what humans see may go back as far as the ancient Greece, but it was not until Hermann von Helmholtz where the first theories of the visual perception of space were stated (Von Helmholtz 1866). His work proposed that vision was the result of unconscious processes based on previously learned situations. Examining the eye, this German physicist stated that it had poor optical features and that perception was a phenomenon hardly linked with a learning process that lasted many years: the unconscious inference. In that way, humans have a priori expectations of the scene structures such as the position of light and the object orientations. Due to the introduction of the learning process, human vision was seen with a broader perspec-

2. DEPTH PERCEPTION

tive, introducing new disciplines for its comprehension. Helmholtz work provided empirical theories about his studies on spatial, color and motion perception. The relevance of this study made it the reference on the theory of vision throughout the second half of the nineteenth century. Helmholtz theories were redefined using probabilistic Bayesian theory (Bayes and Price 1763), and experimental validation (Mamassian and Landy 2001; Stone and Pascalis 2010; Stone 2011; Mamassian 2006) suggests that priors are indeed very influential in human perception.

In the latter years, namely between 1930-1940 (Köhler 1929; Koffka 1935), Gestalt psychologists presented their theory in perceptual organization. Their basic theory stated that objects are first perceived as whole rather than their individual parts. Their theory of vision was based on the capacity of the brain on figure-forming and visual completion instead of perception of individual, simpler visual elements. The Gestalt school was based under two suppositions. First, the conscious experience is the sum of individual aspects of the individual and it must be considered as a whole. Second, the order in which stimulus were perceived is the same order in which the brain processes the information. That is, if two situations are perceived similarly, the brain will process them in identical ways. One of their founders, Max Wertheimer defined the purpose of Gestalt in (Wertheimer 1938) as:

“There are wholes, the behaviour of which is not determined by that of their individual elements, but where the part-processes are themselves determined by the intrinsic nature of the whole. It is the hope of Gestalt to determine the nature of such wholes.”

Gestalt theory attempted from the first moment to define universal principles which allow humans to make the perception arise as a global process, rather than the sum of local stimulus. From this supposition, gestaltists published in a series of works the famous laws of organization, in which they determine how individual parts are associated to form a global perception. These laws are used as a basis in practical state of the art algorithms in object detection (Carreira and Sminchisescu 2012), optical character recognition (King et al. 2011) or depth estimation (Saxena et al. 2005). Since they are specially relevant to the purpose of this thesis, a short review is given with examples in Fig. 2.3.

Proximity Objects close to another are associated as belonging to the same group

Similarity Objects with similar properties such as shape or color are integrated

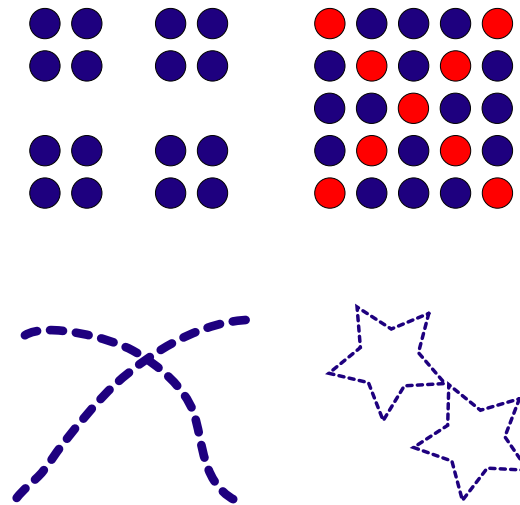


Figure 2.3: Four examples of Gestalt laws of organization. Beginning from the top left, in scan order: Proximity, similarity, continuity and closure

Continuity Oriented units or groups tend to be integrated into perceptual wholes

Closure Individual parts tend to be associated if they are part of a closed form

Common fate Elements moving together are perceived together

Simplicity Perception as a whole tends to be simple, orderly, balanced, unified, coherent and regular.

Although there may be other factors influencing the perception, such as a priori information, these seven laws are the ones used in most applications.

Although these principles seem to play an important role in many situations, the Gestalt theory was strongly criticized due to its descriptive nature instead of explaining these processes. Gestaltist thought that these abilities were innate, successfully explaining *what* brain sees, but they not explaining *how*. At this point, several theories arose to explain how the perceived information was processed. The two most important new theories of vision arose during the second part of the twentieth century. The first one, ideated by Gibson ([J. J. Gibson 1986](#)), claimed that vision should be understood as a survival tool that enables animal to move, eat and avoid predators. Another more pragmatic approach, proposed by David Marr ([Marr 1982](#)) compared

2. DEPTH PERCEPTION

the brain to a computer and stated three levels of information processing for vision. Marr decomposed the problem into three independent levels of understanding:

Computational theory Sets out the goal of the task and why it should be performed. This includes to define which are the inputs and the outputs of the problem. Example: using the right and left eye images, one can perform binocular stereo estimation to be able to move in an environment.

Algorithm Sets out a descriptive process on how to complete the task defined in the previous step. Example: on binocular stereo estimation point correspondences should be used to estimate disparity and therefore, depth.

Implementation Specifies the real (physical) implementation of the algorithm either in biological systems or in computer devices. Example: the visual cortex process or a C/C++ algorithm.

These three levels of description helped modern vision research to decompose the vision problem into the essence of the problem (what needs to be perceived) and how it is performed. With the introduction of Marr theories, the Bayesian theories coming from Helmholtz regained popularity. The modification of the Helmholtz principles to introduce a more objective concept, such as probability, was a first step to explain the vision from a more technical point of view. Although a universal model is still unknown, the proponents considered that the human brain, through processes of (un)conscious learning, stated the problem as a form of Bayesian inference from sensory data ([Moreno-Bote et al. 2011](#); [Knill and Richards 1996](#)). This inference was not treated in the Helmholtz original theory, although the rules that governed this probabilistic approach were not known.

One of Marr's most important contributions was made in the first two levels when he proposed a representational framework for vision. He concentrated on the vision task of deriving shape information and three dimensional structure from images. This is a specially relevant work for the purposes of this thesis and many state of the art algorithms make use of Marr's problem decomposition.

2D or Primal Sketch: the first stage of vision consists in gathering the principal features of the scene, namely lines, common figures, forming edges and regions. This sketch can be compared with the first step an artist would take to represent a drawing.

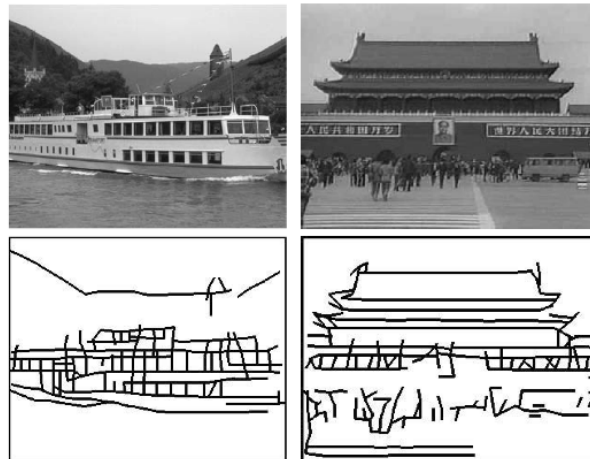


Figure 2.4: Images and their primal sketch, top and bottom respectively.

2.5D Sketch: the second stage gathers texture. In this step, the lighter and darker regions are detected, identifying shades, as well as surface orientations thanks to textures. Object and depth discontinuities are also detected.

3D Sketch: the final step of the algorithm combines all the previous perceived structures and constructs a 3D scene, hierarchically organized in terms of volumetric and surface primitives.

Figure 2.4 represents what the first sketch is. Marr theory does not include the use of multiple points of view (the two retinal images, for example). To adopt several perspectives, the general approach is to consider that after the three sketches the different images are combined for a better scene understanding.

The model of David Marr was very helpful in computer vision and image processing areas which were seeking algorithms for scene understanding. Nowadays, in the computer vision field, the primal sketch is performed mainly by the edge detectors, the 2.5D Sketch is more focused on region/texture segmentation while the last step, the most difficult one even now for computer systems, is performed by (among others) pattern recognition and scene understanding/interpretation algorithms. Following the work of David Marr, (Guo et al. 2003; Guo et al. 2007) proposed practical approaches to arrive at a representation of the primal sketch. The work presented in this thesis is focused on the second and third steps of Marr's theory: Provide a 2.5D representation and, when possible, estimate a full 3D structure of the scene being observed.

2.2 Depth Perception in Humans

Depth perception, as a field of scene understanding, is the one of the most difficult part of image processing, either for the brain or for a computer. While Marr's theory is a good starting point, there still remain unsolved problems both in the way humans perceive depth, and how a computers can estimate it from 2D observations. Nevertheless, the common agreement is that humans perceive depth from a set of simple cues and that these cues appear depending on the situation. For example, it is different to perceive depth in real environments than in single photos or video sequences in flat screens. When multiple views are available, depth cues are much stronger than when only one single view is present. Nevertheless, the depth perception is a process where the set of cues compete with each other, hopefully creating a coherent interpretation of the scene. Many psychologists and artists ([Ponzo 1910](#); [Escher and Brigham 1967](#)) observed that this competition may not always give a unique outcome, and created a set of optical illusions to show that depth perception is one of the most difficult problems in vision.

Noting that depth perception is the inverse process of the visual system: while vision projects a 3D world into (multiple) images, the 3D information is still needed for many purposes. Therefore, the task of depth perception is an ill-posed problem ([Bertero et al. 1988](#)). Due to the loss of information of the 3D-to-2D projection and given an image, there are an infinite number of possible 3D reconstructions. Even with more than one point, there exists some ambiguity on the position of the objects in the space. Therefore, it is only by some intrinsic suppositions that humans are able to estimate depth such as local cues or a priori known situations. The nature of the cues differs very much whether the observer has one/multiple or static/dynamic points of view. A first description is given to differentiate these scenarios and an overview of the cues used in each of the situation is exposed in subsequent sections.

Multiple Views: This case occurs either in natural human vision or in multiple camera systems. It provides multiple points of view of the same scene. In the case of humans, these two viewpoints correspond to the eyes; for computers, there can be more than two viewpoints.

Monocular View: A single point of view is present. This situation can be found also in nature (animals with one eye on each side of the head) or in computer systems (for instance, viewing a photo in a LED display).

Motion Information: This type of information can be associated to either multiple or monocular views. The particularity resides in the temporal information that can be gathered. As in the first two cases, motion information is present in nature or in computer systems equipped with video devices.

2.2.1 Multiple View Depth Cues

Multiple view appears naturally in humans with only two points of view. This kind of structure is known as stereo or binocular vision: only two very close points of view are available. Although there are popular systems, such as the 3D cinema, which uses stereo vision, systems relying on multiple viewpoints (more than 2) usually perform more robustly. Obviously, a system with more than two viewpoints is impossible to have for human vision. Architectures with more points of view are only available in computer-aided systems. Two types of cues can be found to infer the depth from a pair of (or more) images.

2.2.1.1 Binocular Disparity

When two cameras located at different positions observe the same scene, the created images are closely related. Normally, large regions of both images can be matched as shown in Fig. 2.5. From the displacement of the matched region and the knowledge of the camera positions, it is possible to infer the absolute depth of the objects present in the scene. The use of only two points of view may introduce some uncertainty areas. these areas are regions that can be observed only from one point of view. Therefore, in these areas disparity is not available. If the uncertainty regions needs to be reduced, this can be achieved gradually by introducing more cameras. Although disparity has proven to be one of the most reliable cues (Jones and D. N. Lee 1981; Burr and J. Ross 1979), it is not always the case, as conjectured by (Antonides and Kubota 2013).

2.2.1.2 Vergence and Accomodation:

In humans, the visual axes of the eyes (cameras) must converge on the observed object to allow to focus and to infer the depth information. The synchronized eye movement is known as vergence, and this movement depends on the object spatial position. If objects are far away, eyes diverge, while if they are close enough, the eye movement converges (the pupils become closer to each other). Depending on their position, humans know if they are focusing a near or far object (Wismeijer et al. 2008). Note

2. DEPTH PERCEPTION



Figure 2.5: Depth perception from two views. From left to right: left image, right image and true depth map. The depth map can be constructed from binocular disparity.

that situations in which the eyes do not converge correspond to peripheral vision. Although humans can see outside the main focused object, their ability to detect shapes and depth decreases abruptly outside a field of view of approximately 30 degrees. Peripheral vision is used to detect rapid movements and the background structure, but humans must focus objects to examine them carefully (Day and Schoemaker 2004). Accommodation is closely related to vergence, in the sense that the lens should focus near and far objects depending on the position of the eyes. From the eyes muscles controlling the lens, humans may infer the depth at which they are focusing.

2.2.2 Monocular Depth Cues

Monocular vision does not occur only in computer vision. Animals that have one eye on each side of the head, can only rely on monocular cues to detect depth. There is an evolutionary theory stating that animals that need high precision in their fast movements (such as predators) have their eyes coupled to permit stereo vision at the expenses of a reduced field of view, while animals which do not (as herbivores) can rely only on monocular cues with a much broader field of view (Henson 1998). Stereo vision allows to compute distances precisely for hunting, while a broader field of view allows to detect incoming dangers much easily. It is important to note that the cues acting for monocular vision may work together with multiple viewpoints. Humans, for example, a part from disparity, they can take advantage of monocular depth cues. To infer depth using only one image, there are several cues which can be used, and some of them are presented here.

2.2.2.1 Oclusions / Interposition

Junctions Occlusion, also known as interposition, is known to be a strong depth cue and it is found locally at some special points, known as junctions (Anderson 2003). These kind of points are created thanks to the projection of the real world scene to a visualization plane. Junctions have been suggested to be involved in many depth perception tasks, specially surface occlusion geometry retrieval. There are many kinds of junctions (Malik 1987): T-junctions (Guzmán 1968) and L-junctions (Rubin 2001) and X-junctions (Anderson et al. 1997) are strong indicators for occlusion (T and L types) and transparency (X types). L-junctions are also commonly known as corners and can be closely related to convexity cues, so they are discussed more deeply in the next section. Transparency is a phenomenon that rarely occurs in natural images and it will not be further discussed here, although in (Dimiccoli 2009) a special treatment was given to X-junctions. Oclusions occur mainly in T-junctions and, according to (McDermott 2004), they can be easily detected by humans and can be combined to provide an initial estimation of the depth of an image. Nevertheless, occlusion only permits to determine the relative depth order of the regions involved.

T-junction are not always good indicators of depth. To do so, specific sizes, angles and boundary coincidence are needed (McDermott 2004). As an example, in Fig. 2.6, when projecting spheres in the real world into circles in the image, some intersections (junctions) are created where generally three objects meet. Locally, at these intersections, the boundaries of the objects define the junction angle characteristics. With a global view of the image it is easy to see which objects are in the front of in the back, but locally in junctions, the depth configuration is difficult to tell. Normally (but not always), the region belonging to the object lying closer to the viewpoint will form almost a flat angle in the junction and hence the name of T-junction. The other two regions will form two arbitrary but similar angles. In Fig. 2.6, it can be seen that, in the marked T-junction, the red region, R_2 , occupies most of the local window, shaping a nearly perfect 180 degree boundary with the other regions.

Stronger and more confident T-junctions appear when the two rear regions, although not having any restriction about the angle, form a smallest angle bigger than 40 degrees; below from that, the perception of depth decreases rapidly, (McDermott 2004). Even for humans, T-junction detection may be difficult locally and some global reasoning from the image structure is needed. Moreover, the scale to correctly classify T-junctions greatly depends on the image nature (synthetic or natural) and many other factors such as colors, angles, etc. (Lindeberg 1994). If no global reasoning is done, T-

2. DEPTH PERCEPTION



Figure 2.6: T-junction examples. In the left image: locally, region R_2 is the one forming the largest angle, appearing to be over R_1 and R_3 . At the center image, the depth ordering is inverted, since the sky region is forming the largest angle but belongs to the background. At the right image, a T-junction counterexample formed by texture variation is shown. Stripes in the tiger form junctions with the background, but the foreground regions are the smallest ones.

junction depth order cannot be reliably set. For example, textures may generate color differences which, at a small image extent, can replicate the region angle configuration, see center and left pictures of 2.6 for a couple of examples.

A simple classification on T-junction can be done, depending on their local depth ordering: the *normal* and the *inverted* order. The former class is when the region forming the largest angle is the foreground. The latter class is when the opposite depth order is found. Locally, both types of junctions have the same feature configurations. As an example, see how the configuration in the center and left images in Fig. 2.6 is very similar to the right T-junction in the same figure. However, humans interpret correctly the types of junctions easily.

Convexity: Convexity is also a good sign for perceptual organization in an image. Psychological studies such as (Burge et al. 2010; C. C. Fowlkes et al. 2007; Slugocki et al. 2013) aim to prove that natural objects which present convex shapes appear to be in the foreground, while the concave ones seem to lie in the background. Highly convex boundaries are also known as corners or L-junctions (Rubin 2001) and provide a lot of information on surface organization and on object segmentation. (Hoffman and Singh 1997) states that natural objects such as persons, animals, trees... are mainly composed of convex parts. Recalling Marr's theory in Sec. 2.1, the first step humans do when facing an image, is to detect object boundaries in the primal sketch. In a second step, the integration of these boundaries may form the observed object shape, normally composed of a (non-smooth) closed contour. The curvature of this contour is what will determine the first insights on depth organization. For example, in convex object parts such as people extremities like legs, hands or head, the curvature of the shape will be positive, while in concave part the curvature will be negative. From the



Figure 2.7: Example on how the minima rule is used to extract points of high curvature and a global depth decision from local convexity relations. In the left image the boundary of the animal is shown outlined in white. In the right, points with convexity are marked with green arrows, while points with red arrows indicate high concavity. The final depth decision results from averaging these points.

curvature characteristics, the overall shape is divided into smaller parts at the points of negative minima curvature. This division appears to be inherent in vision, and it is by the averaging of the local depth sign of these parts that relative depth is perceived. That is, the minima rule (Braunstein et al. 1989; Siddiqi et al. 1996; Walker and Malik 2003) is a procedure for dividing a shape/mesh into simpler subparts, at the points of high curvature (Y. Lee et al. 2005). As a result, objects in the scene may present points of high positive curvature (and thus locally convex) and points of high negative curvature (perceived locally as concave). Thus, to decide if an object is the foreground or background region, humans integrate along the overall shape the local convexity decisions. The overall decision will depend on the averaged sign of the curvature/convexity as shown in Fig. 2.7

Although convexity may present a correct approach for depth perception, it only permits to determine the relative depth order like T-junction. Usually, convexity cues are weaker than occlusion cues. The human vision system will then first use occlusion to structure the image. After this first step, if no occlusion relation can be inferred from a set of image regions, the foreground/ background relations are inferred mainly from convexity shapes.

2.2.2.2 Light Shading and Shadows

Retrieving depth in some specific situations can also be done by means of analyzing the reflectivity of objects and the casted shadows. The work in (R. Zhang et al. 1999) summarizes the computational techniques to do it but, as always, humans are much more effective in this area (Kleffner and Ramachandran 1992). Nevertheless, there

2. DEPTH PERCEPTION

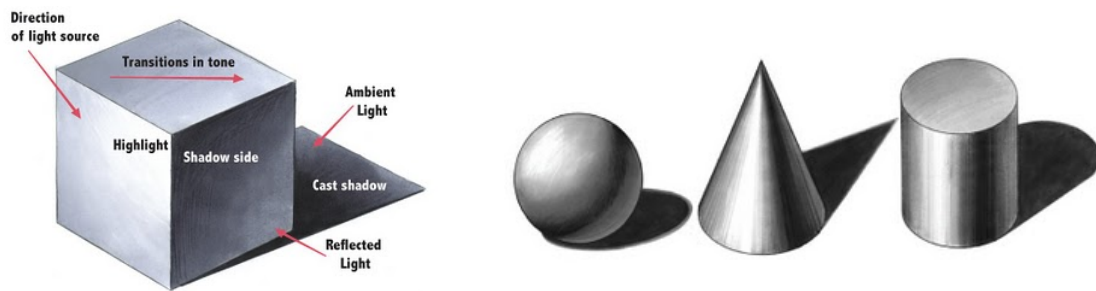


Figure 2.8: Shape from shading examples. Left: example with light effects carefully detailed. Right: three different objects with their shading and their shadow casted to an imaginary ground.

are strong assumptions made about the scene when relying on this cue. First, only one prominent light source may be present and second, this light source is shining from above. The second assumption comes from the subconscious inference theory (Von Helmholtz 1866), commented in Sec. 2.1. These assumptions come from the fact that in most natural images, the light structure conforms to this model, i.e. the sun shines from the top. There is however, a huge literature on this topic and in (Atick et al. 1997; Prados and O. Faugeras 2006) some computational approaches are proposed. Assuming a lambertian surface, where light is scattered uniformly in all the directions, light shading allows humans to infer the object shape. Shadows, in the other hand, allow to relate the position of the different objects relatively to the light source. Some simple examples are given in Fig. 2.8, where one can see clearly that object shape greatly contributes to perceive the surface curvature of the objects and shadows allow to know the position of the ground and the direction of the light source.

2.2.2.3 Relative Dimensions and their Cognition

If two objects are known to be of the same class (e.g. two persons) but their absolute size is unknown within a degree of variability, the two objects observed size in the image can provide information about their relative depth (Hochberg and McAlister 1955). This cue is known as relative size and, from the viewpoint (a camera, for example) the observed size of an object is measured by the visual angle occupied on the field of view. The bigger object will appear to be nearer. Additionally, if some insights are known about the absolute size of an object, some suppositions about their absolute depth can also be inferred. As the visual angle projected to the camera (or retina) decreases with distance, the lesser the area is occupied in the image, the further appears



Figure 2.9: Examples of size cues. Left: from the relative size of the two boats, one can derive that the smaller one is further from the viewpoint. Right: Regardless of the illusion, the Eiffel tower is known to be much bigger than the person, so it is placed far away from the man.

the object. This cue is only applicable with prior learning of the objects, such as persons, cars and several other familiar things from which its size can be approximated. Fig. 2.9 shows two examples of size cues that help to determine the relative position of objects in the scene. Since humans easily recognize objects, this cue appears whenever a known class of objects is seen (which is almost always).

2.2.2.4 Texture Gradient of a Surface

The texture gradient (Clerc and Mallat 2002; Bajcsy and L. Lieberman 1976) is defined to be the distortion in size experimented by regions close to the point of view with respect to regions far way, Fig. 2.10. Texture by itself, is also known to help in image segmentation and object differentiation. To represent texture two approaches are used in the literature: frequency/space oriented filter banks (in which Gabor filters, (Clausi and Jernigan 2000), is a particular case) and Markov Random Fields, (MRF)'s. However, to compute the texture gradient, the former approach is much more used. Generally, it will be defined as the increase of frequency of close image regions (Malik and Rosenholtz 1997). Normally, under known light and camera conditions, texture gradient offers the possibility to infer the three dimensional shape of the objects, providing the absolute depth of their surfaces. If no information about the camera and lighting properties are available, still some insights about the surface orientation may be estimated (Malik and Rosenholtz 1997). Texture gradient appears normally in very

2. DEPTH PERCEPTION



Figure 2.10: Texture gradient examples. Notice the incrementing high frequencies when the surfaces move away from the camera.

specific environments ([Okoshi 1976](#)), so its importance is relative.

2.2.2.5 Atmospheric Effects : Visibility Variation

This cue is observed in very specific situations, where the scene extend may reach several kilometers. Typically, when one observes a landscape, points very far away appear blurred and with low contrast. This blurring is due to the effects of the atmosphere, making further away points to fade into the same color than the sky. Although it can be used to distinguish relative depth, this cue is very approximate since the effects of the sky will differ from one place to another, or even from day to day at the same place. An example is shown in the left part of Fig. [2.11](#)

2.2.2.6 Linear Perspective

This cue is closely related to what is called vanishing point. Due to the projection of the 3D-world into the image plane, parallel lines in the real scene appearing the 2D image as lines crossing at a common point. This effect appears mostly in lines perpendicular to the image plane. Lines parallel to the image plane do not experiment this effect. Needless to say, the closer are the lines observed in the image, the further they appear to be. As an example, the right image in Fig. [2.11](#) shows a clear case of converging lines at the vanishing point.



Figure 2.11: Atmospheric and perspective cue examples. Left: atmospheric effects provide with a good cue to determine which regions in the image are far apart. Right: perspective projection creates a clear vanishing point, where straight and parallel lines appear to converge in the image.

2.2.2.7 Other secondary cues and competition

A part from the cues exposed above, there are other minor cues which may help to infer depth in specific situations. These cues can be peripheral vision, accommodation, retinal image size, height in the visual field (Schwartz 2004). Although they can also be used, they are not considered to be as important as the other ones because of the low-rate appearance in natural scenes or their lack of reliability. One cue, which is very important but very specific and thus, of unpractical use, is the familiar configuration cue. If a situation has already been observed and learned, realizations of the same or similar situations may induce to the same depth perception, regardless of other cues. Another important factor to consider is that in an image, many of these cues may be present at the same time, thus giving a much richer depth information than one cue alone. However, this information should be combined somehow, depending on the degree of cue importance. Although all vision theories suggest that this is indeed the operation done by the brain, the way in which they are combined is still not completely known (Van den Berg and Brenner 1994; Landy et al. 1995; Qiu and Von Der Heydt 2005). Some consensus exists on a Bayesian approach (Jacobs 2002; Kersten and Yuille 2003), and it is indeed what many computational approaches try to mimic in scene understanding algorithms (Ren et al. 2006; D. Hoiem et al. 2011). However, to show that even humans have difficulties interpreting some scenes (Gregory 1994), examples on wrong cue competition are given in Fig. 2.12, and a brief explanation of each illusion is given below:

- Perspective and position of the object makes the three cars appear to be of differ-

2. DEPTH PERCEPTION

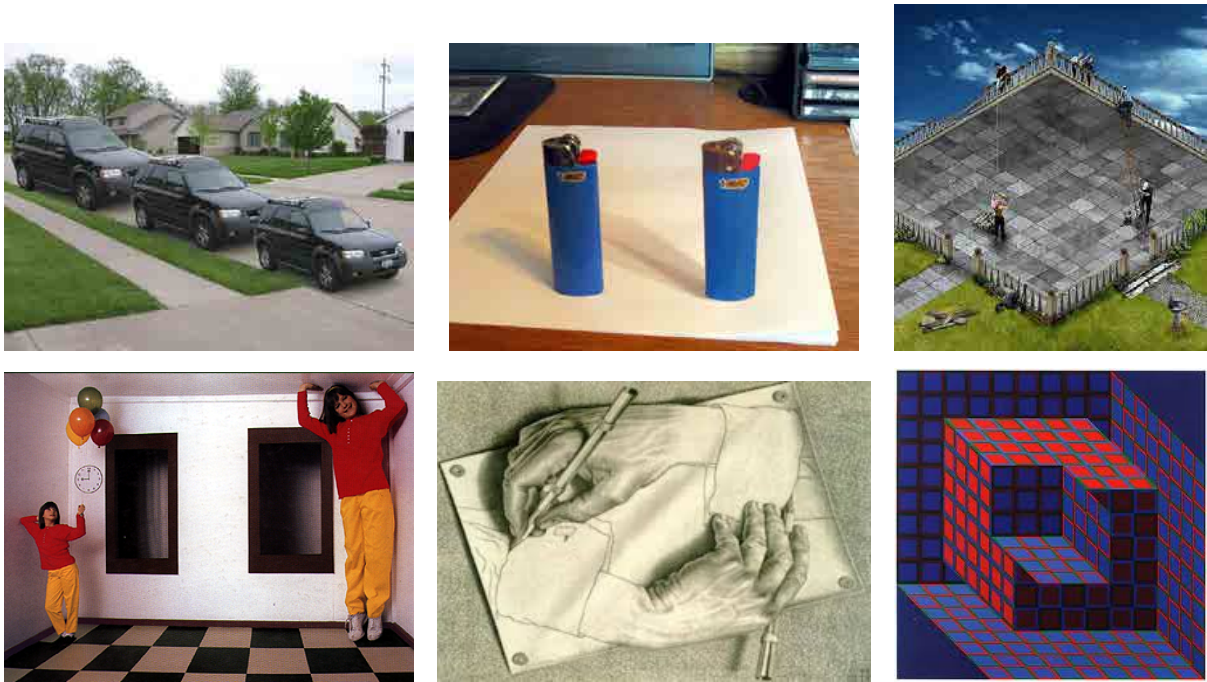


Figure 2.12: Examples of optical illusions due to depth cue competition. Generally, the trick is to generate contradictory cues to induce a wrong interpretation of the scene. A brief commentary is given for each one, from left to right, top to bottom. See description of each image on the text.

ent sizes, although they are not.

- Due to shadow casting and perspective correction, the right (drawn) lighters appear to be standing up rather than in the paper plane.
- A global depth interpretation of this image is impossible due to familiar configurations.
- Ames room ([Gregory 2005](#)). A precisely distorted room also confuses the sizes of the objects within it.
- Light shading and shadows are created artificially to make the hands appear to be outside the paper.
- The viewpoint angle and the lack of junctions of this image does not permit to know the surface orientation.

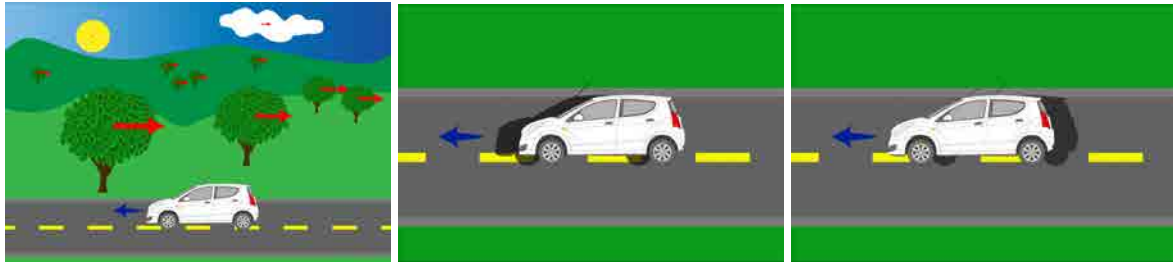


Figure 2.13: Example of motion cues. In the left an illustrative example of motion parallax can be seen. The center and the right images show two consecutive frames with motion occlusions marked in darkened areas.

2.2.3 Dynamic Depth Cues

2.2.3.1 Motion Parallax

When the observer moves, the relative movement of objects in the scene with respect to the point of view may give hints about their relative or absolute distance if some parameters are known, (Ono et al. 1986; Nawrot and Stroyan 2009). In the projected image, normally, if the background is still (not inherently moving) objects far away will be displaced much less than objects being near the observer. A typical example of this situation is the view from a moving vehicle window such as a car or a train. If one would look through a lateral window, far objects will appear to be much more fixed than objects near the vehicle. An illustration can be seen in Fig. 2.13, where the trees near the moving car appear to move faster than far away trees, the cloud and the sun (which almost does not move).

Motion parallax appears always when objects in the scene are static between each other and there is relative movement between the camera and these objects. In this situation, motion parallax and stereo vision can be easily related because essentially they are equivalent (B. Rogers and Graham 1982).

2.2.3.2 Motion Occlusions

When objects move relatively to the camera, background areas may appear and disappear, providing a reliable cue to determine the depth order. Note that motion occlusion appears when the apparent motion of two overlapping objects/regions is different. This situation occurs either when:

- The real motion of the two objects is different (e.g. two cars in a road)

2. DEPTH PERCEPTION

- The scene is static and the object depths are different (e.g. a building occluding the sky)

Given two consecutive frames from a video, occlusions are points in one frame which have no corresponding point in the other frame, see Fig. 2.13. Therefore, it is possible to distinguish several classes of points depending on their contribution to motion occlusion cues. Points that appear on the first frame but not in the second (that is, they disappear) are commonly known as occluded points, while points appearing in the second frame but not in the first (they appear) are known as disoccluded. As with occlusion cues in the static case, motion occlusion only allow to determine the relative depth order between objects in the scene.

2.2.3.3 Depth from Motion

Depth from motion is also a cue present only when several images are available. It is strongly related to the relative/familiar size monocular depth cues as it also relates the sizes of regions and objects in the scene (Sperling and Doshier 1994). Depth from motion is created when the objects move in a direction which is parallel to the view axis. During such movement, objects grow and shrink, depending on whether the observer moves away from or get closer to the object. This situation offers two types of cues. First, if the objects change their size, they also change their depth. Second, depending on the degree of change it is also possible to know which objects are nearer than the others. This cues is rather specific, as it appears only when a special case of motion is present.

2.2.4 General Cue Combination

Normally, several types of depth cues are available when observing a scene. If a real scene is observed with ones eyes, stereopsis is available. If , instead, a photo in a frame is observed, only static cues are available. Many studies investigate how these cues are combined (Landy et al. 1995; Ernst and Banks 2002; Hillis et al. 2002) but the agreement is that humans perform a statistical analysis on the most likely depth configuration based on the observed cues. That is, there exist a Bayesian prior which influences the confidence of each depth cue based on prior experiences or innate conditions (Bulthoff 1996). There is a general agreement in the community that the most reliable cues are the ones obtained when both eyes operate (stereopsis), followed by dynamic cues and static cues. This fact can be correlated by the performance of systems proposed by the

computer vision community. In this field, stereo vision systems (Baker, Szeliski, et al. 1998; Hirschmuller 2008) are known to be quite reliable, followed by the depth from motion algorithms (Davison, Reid, et al. 2007; G. Zhang, Jia, Hua, et al. 2011). Finally, systems attempting to estimate depth from single images (Saxena et al. 2005; D. Hoiem et al. 2011) are the ones having the most difficulties on providing a reliable system.

2.3 Depth Cue Perception in Computer Vision: Proposed Approach

During this chapter, the way in which humans process visual information has been reviewed, and the set of visual cues that are used to estimate depth are exposed. Although these tasks are performed subconsciously in humans, mimicking this behavior in computers is still nowadays a challenge. We do not know how brain fully works and even if we did, the processing power of the whole brain is still unreachable by today's computers in some tasks, specially in pattern recognition (feature detection, face recognition, high level reasoning, etc.). There are many challenges which are still unsolved in the computer vision community which are needed for the particular problem that is addressed here: depth estimation. Below we review the problems in the main fields which are of concern for the development of this thesis:

Image/Video Segmentation Detecting regions of interest when facing a scene (image or video) has long been known as an ill-posed problem. Two different observers will consider distinct partitions since, semantically, the same object or scene structure can be interpreted differently in the two subjects. Automated systems are getting closer to the performance in segmentation/contour detection, but semantics cannot easily be retrieved (Bertero et al. 1988).

Feature/Cue Detection As well as for segmentation, feature detection performance in computer vision is still far from the capacity of humans to perceive points (or lines, regions, etc.) that are relevant for scene structure inference. Again, this is probably due to the semantic and prior knowledge that humans use to interpret even local cues (Holender 1986).

Global inference Observing cues and objects, humans hardly misinterpret a scene. That is, humans always know how the scene is structured, which are the observed object classes or which people are present in front of him. Computers, on the other hand, find this kind of high level reasoning much more difficult.

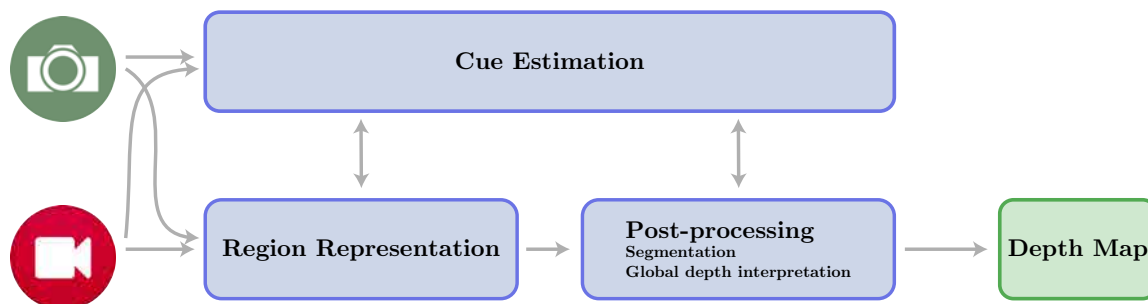


Figure 2.14: Block diagram of the general system architecture.

Thus, to obtain depth information from single points of view, either in images or in video sequences, these three problems should be addressed at some point. The general architecture of the system studied in this thesis can be seen in Fig. 2.14 and it mainly consists of 3 parts which can interact with each other. The system, accepting either images or videos as input signals, first transforms the original discretization of these signals into a region-based representation. In this first process, the ill-posed condition of the segmentation problem is handled using hierarchical representations. That is, in an image, objects and regions can be organized in hierarchies such that the granularity of the segmentation can be chosen depending on the application. After the construction of the hierarchical representation, the hierarchy is processed to obtain a segmentation and to help the global depth interpretation. Since the two previous steps can benefit from cue estimation and vice versa, key features to estimate depth are used throughout the process with multiple interaction which vary from the image to the video case.

Since cue estimation is key to the performance of the system, the following chapter will address the way static and dynamic cues are estimated in this proposed work. In posterior chapters, the particularization of the representation and the postprocessing is discussed for single images in Chapter 5, for single frames in video sequence in Chapter 6 and for full video sequences in Chapter 7.

Part II

Monocular Depth Estimation from Occlusion Cues

3 Monocular Occlusion Cues

3.1 Depth Cues in Static Images

In Sec. 2.2 we reviewed the types of depth cues that humans use to estimate depth from the scene. Many factors come into play when understanding a scene, although many stimulus come in very specific situations (Nagata 1991). For example, when looking at a landscape, binocular disparity does not help for very far away objects, but atmospheric effects could play an important role. In interior scenes such as offices, perspective cues may help to determine surface orientations. From all the depth cues exposed in previous chapters, there is one type that can appear in every situation: occlusion cues. Occlusion is manifested through T-junctions and convex/concave contours and their are present in almost all types of scene. Nevertheless, there are several limitations of occlusion cues:

- They only establish a relative depth order, not an absolute one.
- Individual cues are detected locally, albeit the depth interpretation is a global process.

Obtaining a relative depth order instead of absolute depth values may not be of crucial importance, as from relative depth a quite decent 3D interpretation can be done (Hubona et al. 1999). In the post-processing film industry, depth illusion is created by creating several layers of constant depth, (Van Sijll 2005). Normally, these layers are limited to three: foreground, background and middle-ground. However, in the last years, producers pushed the industry to use more complex systems, such as (Phan et al. 2011) which allowed the creation of many layers instead of only three. As an example of a real post-production case, Fig. 3.1 show artist depth annotations on one frame of the well known film, “The Lion King”. As can be seen, relative depth plays a very important role since it is the first step into inducing smooth depth gradients to the scene. The mentioned figure shows how tedious this process can be, so a system which is able to automatically retrieve these layers can very useful to the film industry, among others. To the date, the most used tool is Stereo-D software¹, which is also a semi automatic driven system which allows the user to input depth values to the image which are interpolated automatically to produce depth.

¹www.stereodllc.com



Figure 3.1: Depth estimation as a post-production step. Left: Reference frame of the “Lion King” movie. Center: annotations done by hand indicating a relative depth with (rough) depth values. Right: Final depth results. See how using only relative depth, the depth map can have high quality.

Relative depth is estimated in local cues, although the depth interpretation of the scene is a global process. So, this implies that automatic systems exploiting occlusion cues should consist of two steps: a low level detection followed by a high level (global) reasoning on scene interpretation. In this section we concentrate on exposing the cue detection process.

Detecting both T-junction and convexity cues has been already tackled in several works. Junctions can be seen as the confluence of three different contours (Y-junctions), a contour ending on a straight contour (T-junction) or two crossing contours (X-junctions). Although X-junction provide cues related to transparency (Beck et al. 1984), they are rarely found. Y-junctions are normally found when three surfaces with different orientations, but no depth discontinuity are found. Among all three, T-junctions are the only kind of junction which provide a clear depth sign for image structure interpretation.

Since junctions can be seen as a special kind of corners, contour and corner detectors are used to detect them. A modified version of detectors (Harris and Stephens 1988; S. M. Smith and J. M. Brady 1997; D. R. Martin et al. 2004) are used to detect junctions in the works (Bergen and Meyer 2000; Dimiccoli 2009; M. Maire et al. 2008) respectively. None of them uses segmentation information to help the contour detection step, although the inverse situation can be found in (Ishikawa and Geiger 1998), where junctions help the segmentation process. Whether segmentation should help the junction detection process or vice versa is a kind of ‘chicken and egg’ problem that we are going to address in Sec. 5.2 but, for the moment, it is assumed that during the detection process a segmentation of the image is available.

In the work of (Calderero and Caselles 2013) a generalization of occlusion cues is proposed by computing the *ownership likelihood* to a component. The authors integrate the

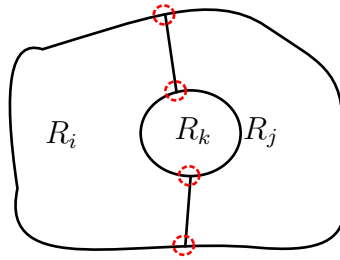


Figure 3.2: Example of multiple T-junctions between a pair of regions.

principles of depth perception using occlusions by using level sets theory. The authors propose to use the dead leaves model of an image of (A. B. Lee et al. 2001), where the image is formed by a set of overlapping components. The ownership likelihood is derived from (Kogo et al. 2010), by considering T-junctions, convexity and corners in a multiscale approach. A global integration is then performed by first combining the estimated cues at different scales with a posterior diffusion process as in (Dimiccoli 2009) to propagate depth values through the image.

3.1.1 T-junction estimation

Several approaches can be found in the literature about T-junction estimation and many of them rely on a hard threshold to detect these points (Lindeberg 1994; Dimiccoli 2009; Ruzon and Tomasi 2001; Bergevin and Babel 2004). The proposed system attempts to assign a confidence value $0 \leq p \leq 1$ to each point of the image, determining the probability of that point to be a T-junction. Since junctions are formed when boundaries meet, their coordinates lie in-between pixels.

In this section, we assume that we are analyzing a T-junction candidate local configuration on a point where 3 regions R_1, R_2, R_3 meet. If R_1 and R_2 share a common neighbor R_3 , at least a T-junction candidate n is present at the contact point(s) of the three regions. Depending on the region shape, there may be more than one junction, see Fig. 3.2. For each candidate n a probability p_n of occlusion is computed in which one of the three regions R_1, R_2 or R_3 may be on the top the other two. To simplify the notation, we call p this value of p_n . To estimate the confidence value p of a T-junction, color difference, angle structure and boundary curvature confidence are evaluated at each candidate point within a centered circular window ($R = 10$), except for the angle. Color contributes to differentiate between contrasted regions, angle helps to infer the depth relationship and curvature detects if the junction has clearly defined boundaries.

Since they are independent features, the final confidence is computed as the product of the three confidences: $p = p_{color} \times p_{angle} \times p_{curve}$.

3.1.1.1 Color

When a T-junction is formed in an image at a location \mathbf{p}_t , it may have some color characteristics that indicate a depth discontinuity. Rather than the common way to represent color in the three primary channels Red-Green-Blue, the color space chosen is the CIE Lab (Robertson 1990). Due to its perceptual nature, numeric differences in the Lab space correspond directly to perceived color differences. The analysis of the color characteristics is limited to a local neighborhood $\Omega(\mathbf{p}_t)$, see Fig. 3.3. In this local window, the three regions can be modeled with a three dimensional histogram. As shown in Fig. 3.3, the pixels used for color confidence(s) evaluation are the ones which are not neighbors of the other two regions. Due to the blurring of contours, all region boundary pixels are discarded to avoid a bias in the signature calculation.

Local Region Model The analysis of the color characteristics are limited to a local neighborhood $\Omega(\mathbf{p}_t)$. In this local window, the three regions can be modeled with a three dimensional histogram. Due to its sparse nature, the 3D histogram is modeled by its n most dominant colors. This adaptive form of modeling is also called signature, and can be expressed as:

$$R_i(\Omega(\mathbf{p}_t)) = s_i = \{(p_1, \mathbf{c}_1), (p_2, \mathbf{c}_2) \dots (p_n, \mathbf{c}_n)\} \quad (3.1)$$

With $i = 1, 2, 3$, R_i refers to each one of the meeting regions at the junction points. $\mathbf{c}_1 \dots \mathbf{c}_n$ are the n dominant colors for the histogram and $p_1 \dots p_n$ are their respective probability of occurrence. n is fixed to 3 and the representative colors are found by using a K-means clustering approach. Since the analysis is done in a local neighborhood, $n = 3$ representative colors proved to be sufficient. The window used for all the calculations is circular with a radius R of 10 pixels. The choice of this value comes from (McDermott 2004) where the author states that a large window is required to have a robust junction detection. In natural images, junctions may appear in different resolutions/scales. In (Lindeberg 1994; Lindeberg 1999) an automatic scale selection algorithm is proposed but, for this project, a fixed window radius proved to be adequate to detect almost all the possible junction candidates.

Considered pixels The pixels which are included for color confidence(s) evaluation are the ones which are not neighbors of the other two regions. All the region boundary

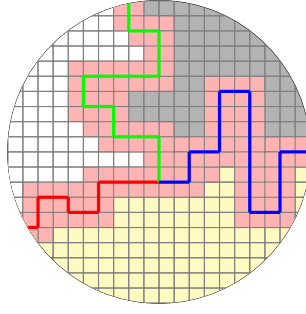


Figure 3.3: Color analysis of a T-junction candidate: Three regions (white, gray and yellow) meet and potentially create a T-junction. Pixels close to the region boundaries (pink) may introduce a bias in the color estimation and are discarded.

pixels are discarded to avoid a bias in mean calculation. During the image formation process, optical devices act as low pass filters (Baker and Nayar 1999). Parts of the image such as edges, that contain high frequencies, appear to be somehow blurred. This blurring introduces false statistics during the color characterization of the regions. For this reason, as seen in Fig. 3.3, these pixels are discarded for the color characterization.

Define s_i $i = 1, 2, 3$ to be the histogram of region R_i near the T-junction candidate. Since a distance measure can only be applied to a histogram pair at a time, a total of three color distances are computed. λ_{ij} , $i < j$, $i, j = 1, 2, 3$, represents the distance between region R_i and region R_j . Distances are computed using the Earth Mover's Distance (EMD) (Levina and Bickel 2001; Ruzon and Tomasi 2001):

$$\lambda_{ij} = EMD(s_i, s_j) \quad (3.2)$$

The EMD distance is defined to be the minimum cost to transport a certain probability masses f_{ij} to transform one signature s_1 to another s_2 , according to some costs between signature colors. Formally, the EMD is defined as:

$$EMD(s_1, s_2) = \min \sum_i \sum_j f_{ij} c_{ij} \quad (3.3)$$

$$\text{subject to: } f_{ij} \geq 0, \quad \sum_i f_{ij} = p_{2j}, \quad \sum_j f_{ij} = p_{1i} \quad (3.4)$$

The costs c_{ij} define the cost of transforming a unit of mass of color c_i in signature s_1 to a color c_j in s_2 . These costs can be arbitrary positive numbers, and in this work, they are defined as: ,

$$c_{ij} = \left(1 - e^{-\frac{\Delta_{ij}}{\gamma}} \right) \quad (3.5)$$

With Δ_{ij} being the euclidean distance between *Lab*-colors c_i and c_j . The decay parameter γ indicates a soft threshold of distinguishable colors and is set to 14.0 as in (Ruzon and Tomasi 2001). Both EMD costs c_{ij} and signature weights p_{1i}, p_{2j} are positive and less than one. Therefore the color distance of Eq. (3.2) gives a value $0 \leq \lambda_{ij} \leq 1$

If $\lambda_{ij} \approx 0$, in Eq. (3.2) the regions R_i and R_j do not seem different in a local neighborhood. Conversely, if $\lambda_{ij} \approx 1$, a strong contrast is present between R_i and R_j . Junctions are supposed to have three high λ_{ij} values. To characterize each points with a confidence value, λ_{min} and λ_{max} are defined to be the minimum and maximum respectively of $\lambda_{12}, \lambda_{13}$ and λ_{23} . Following the ideas proposed in (Harris and Stephens 1988) but adapting them to the notion of color distances, we can distinguish three situations:

- If $\lambda_{min} \approx 0, \lambda_{max} \approx 0$, the pixel p_t does not have any feature of interest.
- If $\lambda_{min} \approx 0, \lambda_{max} \approx 1$, the pixel p_t is likely to belong to an edge.
- If $\lambda_{min} \approx 1, \lambda_{max} \approx 1$, the pixel p_t belongs to a junction.

Therefore, the function determine the color confidence should take values near 1 when both $\lambda_{max}, \lambda_{min} \approx 1$ and values near 0 when either λ_{max} or λ_{min} are 0. A function that fulfills these requirements can be:

$$p_{color} = \frac{2\lambda_{min}\lambda_{max}}{\lambda_{min} + \lambda_{max}} \quad (3.6)$$

The measure in Eq. (3.6) is motivated by the Harris corner detector (Harris and Stephens 1988) and $p_{color} \approx 1$ only when all $\lambda_{ij} \approx 1$.

3.1.1.2 Angle

The angle is a fundamental local cue to determine the depth order of the three regions meeting at a T-junction, see Sec. 2.2. If segmentation is available, the angles of a T-junction point are determined by the region boundaries. Information at the junction center is considered to be unclear, so all the boundaries falling within a small circle of radius 3 are neglected. Region boundaries around T-junctions are locally considered to be straight lines corrupted by noise. The boundary coordinates can be modeled by:

$$\mathbf{b}_{ij}(n) = \mathbf{t} + n\boldsymbol{\varphi}_{ij} + \mathbf{z}(n) \quad (3.7)$$

Where $\mathbf{t} = (t_x, t_y)$ is a vector containing the T-junction coordinates. $\boldsymbol{\varphi}_{ij} = (\varphi_x, \varphi_y)$ is a vector indicating the main direction of the boundary and $\mathbf{z}(n)$ represents the noise.

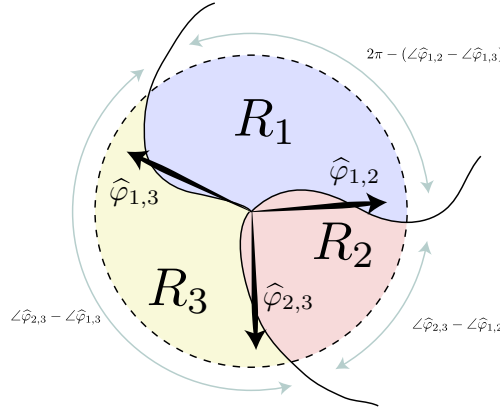


Figure 3.4: Angle computation of a T-junction. See text for the details on how to compute the angles of the branches and the angle formed by each region.

Without the presence of the noise, the region boundary would contain the points $(\varphi_x, \varphi_y), (2\varphi_x, 2\varphi_y), (3\varphi_x, 3\varphi_y) \dots (N\varphi_x, N\varphi_y)$ and would form three perfect straight branches. The tangent vector at each boundary point is approximated with finite differences as $\tau_{ij}(n) = \mathbf{b}_{ij}(n) - \mathbf{b}_{ij}(n-1)$. To mitigate the presence of noise in the estimation of each branch b_{ij} orientation φ_{ij} , the average tangent vector $\hat{\varphi}_{ij}$ is found by means of an exponential weighted mean.

$$\hat{\varphi}_{ij} = \frac{\sum_{n=0}^{N_{ij}-1} \lambda(n) \tau_{ij}(n)}{\sum_{n=0}^{N_{ij}-1} \lambda(n)} = \frac{\sum_{n=0}^{N_{ij}-1} \lambda_0^n \tau_{ij}(n)}{\sum_{n=0}^{N_{ij}-1} \lambda_0^n} \quad (3.8)$$

The total number of considered points for a branch is N_{ij} and depends directly on the damping factor λ_0 and it set to be $N_{ij} = \frac{1}{1-\lambda_0}$. The points near the junction have more importance (and thus are weighted by a larger factor) than the points being further away. Since contour points lie between pixels of integer coordinates, there is a finite number of values for the tangent vectors $\tau_{ij}(n) = (\pm 1, \pm 1)$. This finite set of values introduces high frequency changes in the mean estimation. Therefore, the estimator in Eq. (3.8) should attenuate these high variations while keeping the angle estimation as local as possible. The parameter λ_0 controls both the locality of the estimator and frequency selectivity. Typical values are in the range $\lambda_0 = 0.9 - 0.99$.

At a T-junction there will be three orientation estimates, one for each branch, $\hat{\varphi}_{1,2}$, $\hat{\varphi}_{1,3}$ and $\hat{\varphi}_{2,3}$. The angle of the region is the angle difference of the two vectors concerning

the region boundaries. For example, angle of region 1, $\hat{\theta}_1$, will be $\angle\hat{\varphi}_{1,2} - \angle\hat{\varphi}_{1,3}$ or $2\pi - (\angle\hat{\varphi}_{1,2} - \angle\hat{\varphi}_{1,3})$ depending on the angle of the remaining vector $\hat{\varphi}_{2,3}$, see Fig. 3.4 for details . The angle of a vector $\hat{\varphi}$ is defined as

$$\angle\hat{\varphi} = \arctan\left(\frac{\hat{\varphi}_y}{\hat{\varphi}_x}\right) \quad (3.9)$$

Once the three average tangent vectors are available, each region angle θ_i is used to evaluate junction angle characteristics. Considering the angles, ideal shaped T-junctions have a maximum angle of π and a minimum angle of $\frac{\pi}{2}$. Two measures are then proposed:

$$\Delta\theta_{max} = \|\theta_{max} - \pi\| \quad \Delta\theta_{min} = \|\theta_{min} - \frac{\pi}{2}\| \quad (3.10)$$

Where θ_{max} and θ_{min} refer to the maximum and minimum of the three angles respectively. To obtain the confidence value, $\Delta\theta_{min}$ and $\Delta\theta_{max}$ are considered to be Rayleigh distributed. With this assumption, two confidences can be obtained using:

$$\Theta_{max} = \exp\left(-\frac{\Delta\theta_{max}}{\sigma^2}\right) \quad (3.11)$$

$$\Theta_{min} = \exp\left(-\frac{\Delta\theta_{min}}{\sigma^2}\right) \quad (3.12)$$

$$(3.13)$$

with $\sigma = \frac{\pi}{6}$. This value is obtained from (McDermott 2004), as the perception of occlusion on T-junctions drops rapidly when angle variations are greater than 30-40 degrees from the ideal angle configuration. By combining these two values, p_{angle} is obtained similarly to Eq. (3.6):

$$p_{angle} = \frac{2\Theta_{min}\Theta_{max}}{\Theta_{min} + \Theta_{max}} \quad (3.14)$$

The value p_{angle} , jointly with the color confidence, proved to be the most discriminating factors to compute the T-junction confidence. However, although the computation of the angle estimation may result in good angle distributions, the branches of the junction may not be very regular, having erratic and noisy shapes. To discriminate highly curved boundaries, a curvature measure is introduced.

3.1.1.3 Curvature

Although curvature is not as important as color and angle, it serves to measure the branch straightness. If boundaries are highly curved, the point may not be perceived

as a junction and, instead, only erratic and noisy boundaries are seen. Although the definition of curvature was originally thought in the physics domain, it has its own applications in image processing. Stated in (Guichard and Morel 2001), it may help to describe the shape of the objects presents in a particular scene. It was originally used in curvature scale space representation (Asada and M. Brady 1986) and anisotropic diffusion (Perona and Malik 1990). Curvature was also used jointly with the level sets theory (Guichard and Morel 2001) to determine the curvature of regions in grayscale images. Following this idea, the level sets theory is used to compute the boundary curvature near the T-junction.

Curves on Level Sets Consider a gray level image I and a pixel p_0 . Let $u(p)$ be the gray level of the image at a certain pixel p (which can be the gray level or the luminance for example). Considering that the image is a continuous function, it can be proven that the set of pixels on a level λ , $u^{-1}(\lambda)$, forms a set of disjoint curves. It is possible to calculate the curvature of each curve c . Without loss of generality, from now on a particular pixel p_0 on an arbitrary level set λ is considered. The resulting equations, deduced in (Osher and Paragios 2003; Guichard and Morel 2001), are briefly summarized here.

If the image first order partial derivatives u_x, u_y along the x and y directions are available at a point p_0 and $u_x^2 + u_y^2 \neq 0$. The curvature of the level λ is defined as:

$$\kappa(p) = \frac{u_{xx}u_y^2 - 2u_{xy}u_xu_y + u_{yy}u_x^2}{(u_x^2 + u_y^2)^{3/2}} \quad (3.15)$$

Where u_{xx}, u_{yy} and u_{xy} are the second order partial derivatives. Since in practice the image is discrete, the value of the derivatives should be estimated using any of the available techniques such as convolution by a high pass filter.

The process of curvature confidence calculation is shown in Fig. 3.5 and a summary is given here. Each region R_i is isolated creating a binary image of the local window. Note that since the regions may have arbitrary shapes, other regions than R_1, R_2, R_3 may be present in the local window. To eliminate possible interferences from these outliers, a reconstruction process is performed where, from the boundaries, binary markers are extended eliminating the holes that may be present. The second and third steps in Fig. 3.5 illustrate this hole filling process. Finally, the mean absolute value $|\overline{\kappa}|_i$ of the curvature $\kappa(x_l, y_l)$ of the two branches forming a region R_i is computed at the boundary points (x_l, y_l) in the binary image using Eq. (3.15) (Guichard and Morel

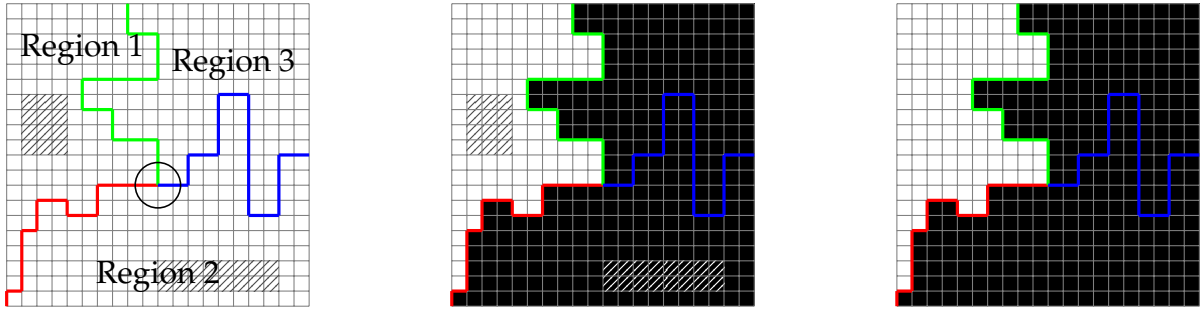


Figure 3.5: Process to calculate the curvature. Left, local window with the three regions and some outliers (diagonal striped pixels, belonging to other regions). Center, binary image where Region 1 has been isolated. Right, reconstructed image without outliers.

2001). Each of the $\overline{|\kappa|}_i$ measures (one for each region) are also assumed to be Rayleigh distributed to obtain:

$$\Upsilon_i = \exp\left(-\frac{\overline{|\kappa|}_i}{\sigma_c^2}\right) \quad (3.16)$$

Similar to color and angle, curvature confidence should have high values when all κ_i are low and the boundaries are straight. Therefore, p_{curve} is obtained by finding Υ_{max} and Υ_{min} :

$$p_{curve} = \frac{2\Upsilon_{min}\Upsilon_{max}}{\Upsilon_{min} + \Upsilon_{max}} \quad (3.17)$$

3.1.1.4 Local depth gradient determined by T-junctions

Previous work on T-junctions (Dimiccoli 2009) imposed unique depth configuration for these kind of cues: the region forming the largest angle was always assumed to lie closer to the viewer. However, experience shows that T-junction may also indicate the opposite depth relation. Since, locally, all kinds of junctions are similar, deciding whether T-junctions are *normal* or *inverted* should be done by looking at other characteristics than intrinsic color, angle and curvature local features. Instead, a global reasoning on all the other depth cues should take place, determining the sign of the depth gradient based on other T-junction observations. We expose this process in Sec. 5.4 as it involves many factors and a complex reasoning.

T-junctions actually indicate depth discontinuities but the sign of the discontinuity proved to be rather uncertain. Normally, if an object is really occluding other objects in the background, more than one T-junction is likely to be formed in the image, and all these T-junctions may have the same region/object as the occluding region. This is

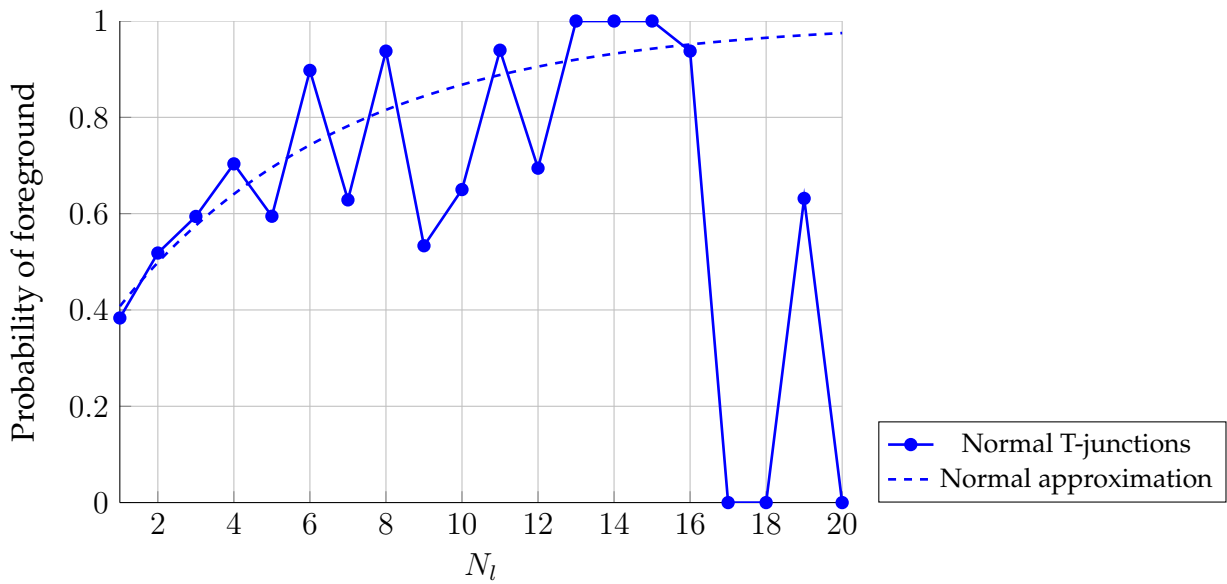


Figure 3.6: Probability of foreground versus the number of times N_l a region is seen as the largest region in T-junctions. When many T-junctions agree on the largest region, it is likely that this region is in front of their neighbors. However, when little T-junction information is available, the decision is much more difficult (and even the inverse conclusion is more likely).

why a global reasoning is helpful. Moreover, it is possible to detect a T-junction even though no real occlusion relation exists. False detections often occur due to color or texture variations. In Fig. 3.6 all T-junctions of the groundtruth contours in the dataset of (D. Martin et al. 2001) are examined and the number of times N_l a region appears as the one forming the largest angle are counted. The figure plots the probability to be the region in the foreground versus N_l , showing that when many T-junctions agree on the angle configuration the decision is more easy to take. However, when $N_l \approx 1$, the depth order determined by a T-junction is somewhat arbitrary. Therefore, a single T-junction cannot determine the order of the regions involved, but additional junction information may help to the estimation process.

In our case, as a starting point we consider that all T-junctions are *normal*. This initial guess has a low confidence and will be allowed to change when estimating the global depth ordering of the scene. That is, in some circumstances, the depth gradient of a T-junction will be changed if there are many other occlusion relations indicating the opposite depth relationship, see 5.4.

3. MONOCULAR OCCLUSION CUES

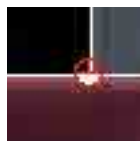
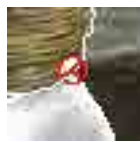
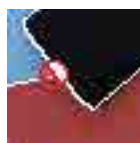
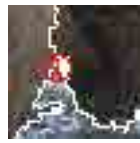
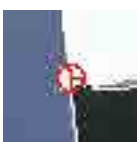
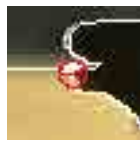
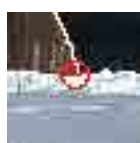

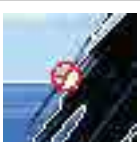
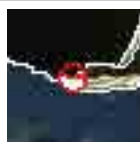
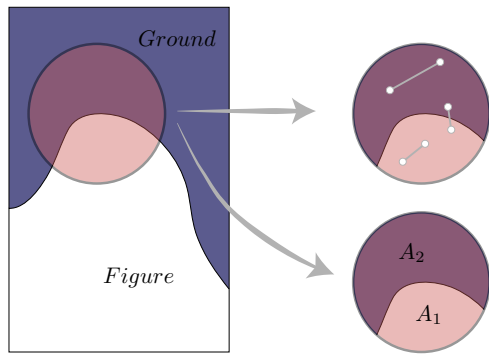
T-Junctions Example					
Local Window	Confidence	Value	Local Window	Confidence	Value
	color	0.84		color	0.73
	angle	1.00		angle	0.56
	curvature	1.00		curvature	0.39
	overall	0.84		overall	0.16
	color	0.97		color	0.55
	angle	0.66		angle	0.39
	curvature	0.80		curvature	0.57
	overall	0.52		overall	0.12
	color	0.88		color	0.51
	angle	0.90		angle	0.25
	curvature	0.56		curvature	0.77
	overall	0.45		overall	0.10
	color	0.77		color	0.19
	angle	0.66		angle	0.40
	curvature	0.59		curvature	0.57
	overall	0.30		overall	0.04
	color	0.42		color	0.12
	angle	0.54		angle	0.60
	curvature	0.79		curvature	0.92
	overall	0.18		overall	0.03

Table 1: Examples of T-junctions, ordered in decreasing value of confidence, from top to bottom and left to right. Junctions are marked with a red circle. The region lying on top of the other two is locally filled in white.

3.1.1.5 T-junction examples

A few examples of T-junction confidence estimation are shown in table 1. Note that the color confidence is the measure that has the highest relevance of the junction as it is the first feature detected by humans when examining the scene (McDermott 2004). Angle plays also an important role in perception. When the biggest region does not give a clear occlusion cue, confidence drops rapidly. Curvature is the less sensitive parameter.



Top illustration: $Convexity \propto \frac{r_1}{r_2}$
 r_1 and r_2 are the fraction of straight lines lying completely in the same region

Bottom illustration: $Convexity \approx \frac{A_2}{A_1}$
 A_1 and A_2 are the region areas
 r_1 and r_2 are the ratio of pair of points which their segment lies completely in the same region

Figure 3.7: Two ways of computing contour convexity. Left: convexity is determined locally at region boundaries. Top right: exact way to measure local convexity, determining the number of pair of points which belong to convex sets. Bottom right: approximate way to compute convexity. Normally, convex shapes present less area in small neighborhoods centered on contour points.

3.1.2 Convexity estimation

Convexity depth cues are defined locally at region boundaries. A region R_1 is convex with respect to R_2 if, on average, the curvature vector on the common boundary is pointing towards R_1 . If R_1 appears to be convex, it is perceptually seen as the foreground region (and thus, closer to the viewer). In the previous section a technique to measure the curvature along a contour was presented, but it presented several limitations, which were not significant for that case. First, using level sets only a very local estimate of the curvature is obtained, while for convexity cues a larger scale is needed. Second, the process of estimation is rather slow and, therefore, for long contours the performance may suffer. For these two reasons, an alternative approach is presented here. Generally, when examining boundary pixels, if R_1 presents less area than R_2 in a local neighborhood, R_1 may be seen as convex, see Fig. 3.7.

In a local window around a contour point p , the degree of convexity of p is defined as the ratio $\frac{r_1}{r_2}$ of straight segments lying completely on the same region. That is, using all the possible pairs of points (a, b) within the window a segment \bar{ab} is formed. If all points of the segment \bar{ab} belong to the same region, r_1 increases. If \bar{ab} falls in-between regions, r_2 increases, see the center-top illustration in Fig. 3.7 for a few examples of segments. Similarly, the area of each region in the local window can be a good indicator of convexity. Following the bottom-center illustration of Fig. 3.7, pixels belong to each region within the window are counted. The ratio $\frac{A_1}{A_2}$ is small when $A_1 \ll A_2$, indicating

3. MONOCULAR OCCLUSION CUES

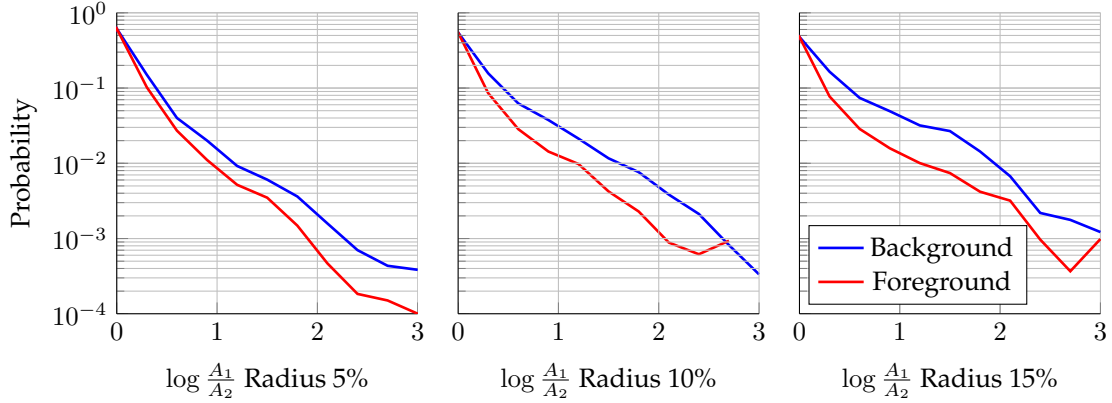


Figure 3.8: Probability density function of $\log \frac{A_1}{A_2}$ with different window radius (expressed in relative terms of the contour length). As the ration A_1/A_2 , the difference between figure/ground becomes more evident. for equal sized regions, humans are not able to distinguish which region is either figure or ground. Note also that the window size does not seem to be an influential factor on the decision.

that A_1 may be seen locally convex.

Reference (C. C. Fowlkes et al. 2007) claims that the analyzed local depth cues (convexity, position in the image and size) are valid for figure ground reasoning in natural images. To prove this claim, statistics on groundtruth contours were gathered from the BSDS dataset (D. Martin et al. 2001). They show that the area, convexity, lower region as well as some non-linear combination of cues are quite reliable to distinguish between figure and ground in a local neighborhood. Here, for the concerning case, the size cue experiment is reproduced, as it gave slightly better results for classification than the true convexity, see (C. C. Fowlkes et al. 2007) for details. For each contour point of a set of groundtruth annotated contours in the BSDS dataset, the ratio $\log \frac{A_1}{A_2}$ is calculated. Fig. 3.8 shows the probability density function of a region being either foreground or background depending on the $\log \frac{A_1}{A_2}$ value, showing that when $A_1 > A_2$, the first region is effectively seen as background.

Formally, the overall boundary convexity is obtained from the combinations of two measures:

$$\zeta_c(R_1, R_2) = \sum_{(x,y) \in \Gamma} \frac{\alpha(x,y)}{L} \sum_{(x,y) \in \Gamma} \frac{w(x,y)}{L} \quad (3.18)$$

With $\alpha(x, y) = 1$ if the area of R_1 is greater than the area of R_2 in $\Omega(x, y)$, and $\alpha(x, y) = -1$ otherwise. The function $0 \leq w(x, y) \leq 1$ is a weighting function of the points and


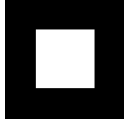

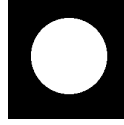
















Convexity Examples					
Local Window	Partition	Confidence	Local Window	Partitions	Confidence
		0.63			0.64
		0.94			0.77
		0.00			0.80
		0.15			0.14
		0.0			0.45

Table 2: Examples of convexity estimation. For each image, the corresponding partition and the confidence of the convexity estimation is shown.

it is chosen to be the normalized Sobel gradient of the image, although other gradient operators work too. L is the number of points where the measure $\alpha(x, y)$ is calculated and it depends of the window size. The overall convexity confidence of a boundary is:

$$\zeta(R_1, R_2) = 1 - \exp\left(-\frac{1}{\gamma_c} \|\zeta_c(R_1, R_2)\|\right) \quad (3.19)$$

γ_c has been determined experimentally and set to $\frac{1}{12}$. If the result $\zeta_c(R_1, R_2)$ is positive, R_1 is considered to be convex and, therefore, on top of R_2 with confidence $\zeta(R_1, R_2)$. The converse indicates that R_2 is on top of R_1 . To make the measure as scale invariant as possible, the neighborhood $\Omega(x, y)$ of a pixel is chosen to be a circular window with a radius of about the 5% of the contour length. Points lying near junctions, image borders and other regions are discarded for the measure. Contours having small lengths ($L < 100$ pixels) are considered to be non-significant for convexity cues.

3.1.2.1 Convexity examples

Some examples of convexity estimation are shown in Table 2. The confidence measure can be seen as the probability of the brighter region to be foreground with respect to the darker one. Some synthetic examples are shown on the upper part of the table, while natural cases are shown in the lower part. When the partition contains a white region surrounded by a black region, the convexity confidence is high, as the overall white shape is seen as convex with respect to the black region. In cases where the region shape is erratic, the convexity measure gives a correct very low confidence, as local contour convexities compensate and no overall cue is observed.

3.1.3 Probability of ownership - Combining T-junction and Convexity cues

Throughout this section two ways of explicitly detecting T-junction and convexity cues have been exposed. In a recent work from (Calderero and Caselles 2013) a different approach on how low level cues are treated is presented. The main idea is to integrate both T-junction and convexity cues into a probability of ownership. Locally, convexity relations can be seen as the interaction of two connected components (region). Similarly, T-junctions can be thought as the interaction of three different components. In this way, (Calderero and Caselles 2013) generalizes low level depth cues between two and three components to an arbitrary number of components. Other than estimating local depth relations created by T-junctions and convexity, a low level indicator of the relative depth can be retrieved by formalizing the *probability of ownership*.

3.1.3.1 The dead leaves model

The approach of (Calderero and Caselles 2013) relies on a complete different model of the image. The image is considered to be generated by a set of opaque components “dropped” into the image plane, possibly occluding each other, called the dead leaves model (DLM). However, in standard image processing applications, such as color quantization (Orchard and Bouman 1991) or denoising (Buades et al. 2005), the image is usually considered to be a random variable with Gaussian random noise. The Fig. 3.9, taken from (A. B. Lee et al. 2001), shows the difference between both models, showing that the DLM is perceptually more close to normal images, as well as offering a mathematical model for occlusions. Additionally, the dead leaves model proved to generate more accurately the statistics of natural images (gradient, scale invariance

...) and in the work from (Calderero and Caselles 2013) is the theoretical basis for the occlusion reasoning.

The DLM assumes that the world can be broken down into opaque pieces and that these pieces are then stacked (projected) into the image plane to form the image of the three dimensional world (A. B. Lee et al. 2001). This model is similar to the local representation used for T-junction estimation, although the DLM is used in the whole image. The approach taken by (Calderero and Caselles 2013) is to assume that these objects are infinitesimal, and that each point in the image belongs to at least one of these components. The authors reason, with Gestalt cues about the shape of occluded shapes an encode inside the probability of boundary low level cues such as boundary convexity and T-junctions. Here a brief overview is given, although the reader is encouraged to see the details in the original article.

The algorithm does not explicitly detect any T-junction or convexity cues, but it estimates the likelihood for a pixel \mathbf{p} in an image to belong to different components. The DLM of an image image is considered to be a union of N overlapping sets $X_1 \dots X_N$ with increasing relative depth, and their visible part is:

$$A_i = X_i \setminus \bigcup_{1 \leq j < i} \text{interior}(X_j) \quad (3.20)$$

The union of all the A_i determine an image partition. The border ownership density function of a pixel, $Z(\mathbf{p})$, is computed only the visible part of each component:

$$Z(\mathbf{p}) = \sum_{i=1}^N D(\mathbf{p}, A_i) \quad (3.21)$$

The density term $D(\mathbf{p}, A_i)$ is defined using two principles so that the pixel is more likely to belong to a set X_i if 1) the pixel is close to A_i and 2) the boundary is highly curved. The concrete expression of $D(\mathbf{p}, A_i)$ can be quite complex and we refer the reader to (Calderero and Caselles 2013) for more details. The function $Z(\mathbf{p})$ is an indicator function on the number of sets the pixel \mathbf{p} belongs to. Thus, if $Z(\mathbf{p}) = 0$ the pixel only belongs to a single component, while if $Z(\mathbf{p}) > 0$ the pixel belongs to more than one component. The only reason for a pixel to belong to more than one component is occlusion. Therefore, $Z(\mathbf{p})$ is a direct indicator of local depth without explicitly detecting occlusion cues. The higher $Z(\mathbf{p})$ is, the closer will be the pixel to the viewer. In Fig. 3.10 example of $Z(\mathbf{p})$ can be seen on the second column showing that, the function $Z(\mathbf{p})$ reacts in points near T-junctions and convexity regions.

3. MONOCULAR OCCLUSION CUES

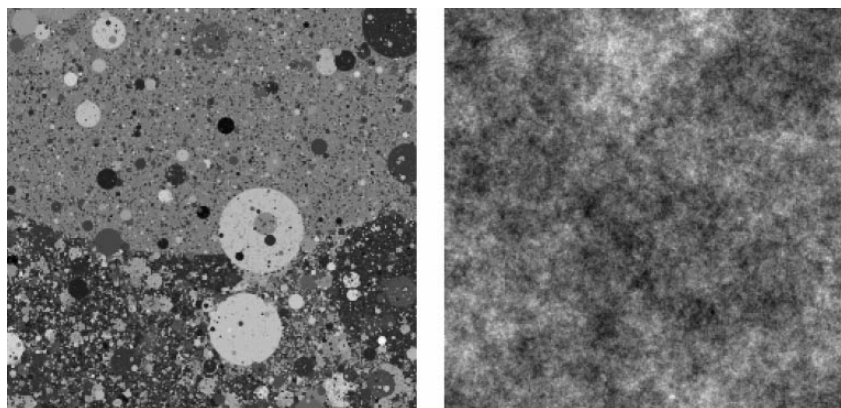


Figure 3.9: Two synthetic generated images showing the difference between the dead leaves model (left) and the additive noise model (right).

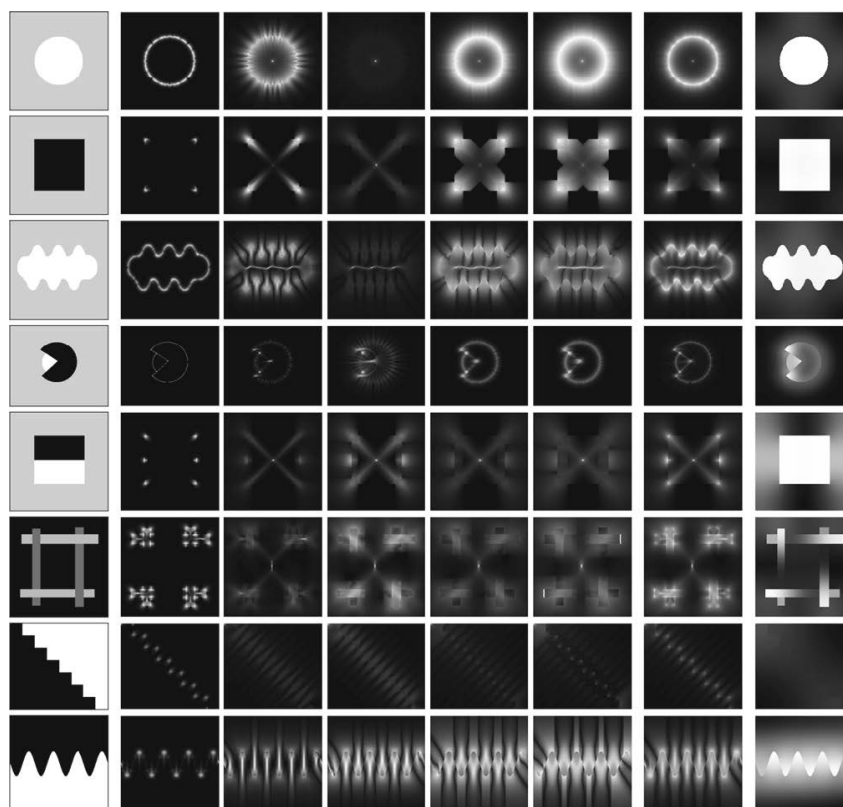


Figure 3.10: Example from (Calderero and Caselles 2013). From left to right, example image, local ownership likelihood (columns 2-6), global feature integration and depth diffusion. See that the algorithm naturally integrates the majority of occlusion cues such as convexity (rows 1,2,3) and T-junctions (rows 5 and 6) and discriminates ambiguous cases such as rows 7 and 8. Note that the case in row 4 is difficult even for human perception, as the depth order is not clear.

If the measure in Eq. (3.21) is computed at different scales, it is possible to estimate the local depth configuration at different scales. The combination of different scales lead to a global depth likelihood for each pixel. Examples of ownership likelihood computation in different situations are shown in Fig. 3.10. Since $Z(\mathbf{p})$ is locally defined near boundaries, a diffusion process is performed to extend its values to the whole image.

It is important to note that while T-junction and convexity give cues between three or two components of the image respectively, the probability of ownership is an intrinsic property of each pixel. It is a direct indicator of the relative depth of a pixel and the obtained probability map is already globally consistent. The main drawback of this technique is that all pairs connected components of the image should be examined, making the process computationally heavy. In (Calderero and Caselles 2013) the CPU burden is alleviated by considering only close components, discarding long range information.

3.2 Depth Cues in Dynamic Scenes

Humans easily sense object movement and the cues generated from it. Movement is a very important cue to determine the depth of a scene, as it can generate either motion occlusions and motion parallax among other kind of cues. What's more important, is that motion cues are very reliable (E. J. Gibson et al. 1959; B. Rogers and Graham 1979). With the only presence of motion parallax cues, humans are able to infer absolute depth. That is, it is possible to generate dense depth maps (up to a scale factor) when the scene moves in some conditions. Motion occlusions, on the other hand, only allow to determine the relative order of independently moving objects. Both cues appear always when objects move, but each of them has its limitations:

- Motion parallax (or motion depth cue) can be very informative about the absolute depth and the 3D structure of the scene only when the objects are rigid. That is, structure from motion can only be perceived when the faced objects have a rigid structure and their shape do not change (only rotations and translations).
- Motion occlusions are always valid, as long as there are two moving objects, or their apparent motion is different with respect to the camera. This situation occurs either when:
 - The real motion of the two objects is different (e.g. two cars in a road)

3. MONOCULAR OCCLUSION CUES

- The scene is static and the object depths are different (e.g. a building occluding the sky)

Note that these two cues can be coexistent and complimentary, helping each other to achieve a complete depth interpretation of a scene. In this section we have been talking about dynamic cues, bypassing the fact that first, we should detect movement. Humans have an inherent capacity to detect movement, as the eyes continuously gather information of the environment (Johansson 1973) and motion is subconsciously inferred (Grossman et al. 2000). However, computers have more difficulties on estimating motion in video sequences, as they can rely only on the projection of objects to camera frames.

The usual approach to detect object motion is to detect optical flow between video frames. That is, it is not the real motion of objects, but their apparent motion from frame to frame. Apparent motion is defined as the motion of a point in an image at time t to time $t + \delta t$. Optical flow is one of the computer vision areas in which researchers have put more efforts, likely because it is of great importance and gives much information for many tasks: coding (Krishnamurthy et al. 1995), segmentation (L. Xu, J. Chen, et al. 2008) or tracking (Sundaram, Brox, et al. 2010) to cite some. Optical flow literature is huge. Here, we will only give insight of a few very relevant works.

- (Lucas and Kanade 1981) Not precisely used for optical flow estimation, but it was first proposed for stereo vision. It is the first system to propose a dense matching using correlation, and it is used until the date.
- (B. K. P. Horn and Schunk 1981) The first dense optical flow estimation algorithm relying on the solution of a variational problem.
- (Black and Anandan 1996) Introduction of robust statistics to allow sharp transitions to appear in motion edges.
- (Brox, Bruhn, et al. 2004) New practical way to estimate the flow, based on a theory of image warping.
- (C. Liu et al. 2008) Introduction of SIFT point based descriptors for flow estimation.
- (Brox and Malik 2011) State of the art technique, used in this thesis for optical flow estimation.

To understand dynamic cues, and how they can be detected in computer systems, it is of crucial importance to learn how optical flow is estimated. The following section explains the most common approaches for optical flow estimation and gives insights about the strengths and weaknesses of the algorithms which can be exploited to estimate either occlusions and motion parallax.

3.2.1 Optical Flow Estimation

Optical flow plays a central role when dynamic depth cues should be estimated. A deeper understanding of how motion is computed in most of the art algorithms estimate helps to exploit flow characteristics for a posterior depth inference. In this section the key parts of the algorithm (Brox, Bruhn, et al. 2004) are shown, as long with an extension to incorporate sparse descriptors (Brox and Malik 2011). Conventional optical flow algorithms attempt to find an horizontal $u(x, y)$ and a vertical $v(x, y)$ flow for each point on the image domain $(x, y) \in \Omega$ by minimizing a functional.

$$E(u, v) = E_D(u, v) + \lambda E_S(u, v) \quad (3.22)$$

The first term $E_D(u, v)$ imposes assumptions on the model, while the second terms $E_S(u, v)$ imposes smoothness on the functions $u(x, y)$ and $v(x, y)$ to overcome the aperture problem (Nakayama and Silverman 1988; Hildreth 1984). λ is often known as the regularization term, as its value can be related on the smoothness of the result. That is, higher λ is, the more smooth/regular the function is. The data model in (Brox, Bruhn, et al. 2004), consists of two premises:

Brightness constancy It is assumed that the color value of a pixel does not change over an interval δ_t , despite a displacement (δ_x, δ_y) :

$$I(x + \delta_x, y + \delta_y, t + \delta_t) = I(x, y, t) \quad (3.23)$$

If the spatial and temporal displacements are sufficiently small, Eq. (3.23) can be linearized. Defining $u = \frac{\delta_x}{\delta_t}$ and $v = \frac{\delta_y}{\delta_t}$ the so called *optical flow constraint equation* is obtained:

$$I_x u + I_y v + I_t = 0 \quad (3.24)$$

Where I_x and I_y are the image horizontal and vertical derivatives respectively. Linearization of Eq. (3.23) by means of Eq.(3.24) is a key step to the numerical resolution of optical flow problems.



Figure 3.11: Optical flow estimation examples. The first column shows the frame at time t of the sequence and the second column shows frame at time $t + 1$. The two last columns show the forward flow field (center right) and the backward flow field (right). Optical flow direction is color coded, whereas the flow magnitude is encoded in the color saturation.

Gradient constancy Brightness consistency has one decisive drawback: small scene illumination changes may lead to wrong pixel correspondences. To overcome this limitation, the brightness gradient, which is illumination invariant, is also assumed to be constant.

$$\nabla I(x + \delta_x, y + \delta_y, t + \delta_t) = \nabla I(x, y, t) \quad (3.25)$$

The gradient of an image at a point (x, y) is defined as $\nabla I(x, y, t) = (I_x, I_y)^T$. As with brightness consistency, Eq. (3.25) is also linearized in the numerical scheme resolution. Conditions on subsequent image moments can also be imposed, but a first order assumption is sufficient for practical scenarios.

When $\delta_x = 1$ the obtained flows are known as forward flow fields, and when $\delta_x = -1$ the backward flow fields are obtained instead. Examples of forward and backward flow estimation are shown in Fig. 3.11 with the technique (Brox and Malik 2011). Although it may be difficult to visually appreciate, but forward and backward flows are not symmetric as every pixel in one image does not have a correspondence in the other image. This property will be the basis for occlusion detection, commented in Sec. 3.2.2.1.

The brightness and gradient consistency constraints are widely used in the state of the art algorithms (Brox, Bruhn, et al. 2004; L. Xu, Jia, et al. 2010). Assuming that $\delta_t = 1$ and forming the vectors $\mathbf{x} = (x, y, t)$ and $\mathbf{w} = (u, v, 1)$, the data term in the energy Eq. (3.22) can be expressed as:

$$E_D(u, v) = \int_{\Omega} \Psi (\|I(\mathbf{x} + \mathbf{w}) - I(\mathbf{x})\|^2 + \gamma \|\nabla I(\mathbf{x} + \mathbf{w}) - \nabla I(\mathbf{x})\|^2) d\mathbf{x} \quad (3.26)$$

Where $\Psi(s^2) = \sqrt{s^2 + \epsilon^2}$ is the Charbonnier robust penalty function (Charbonnier et al. 1994), which results on a modified, differentiable L^1 norm. Other choices for $\Psi(s^2)$ are also possible (Black and Anandan 1996), but the one used in the proposed approach is convex with respect to s , allowing easier numerical minimization.

The use of robust methods, in contrast to (B. K. P. Horn and Schunk 1981), attempt to reduce the influence of outliers in the estimation process and allow sharp transition motion edges. Outliers normally appear at object boundaries, producing an over smoothing if the classical quadratic penalization is used.

Due to the aperture problem, it is only possible to estimate the motion perpendicular to boundaries, so complementary conditions on the flow (u, v) are needed. This limitations comes from Eq. (3.24), where there exists only one equation but two variables must be found (u, v) . Usually, local smoothness is assumed, penalizing high flow discontinuities. Real world objects do not 'break apart' and their motion is coherent along their points, so smoothness is a reasonable assumption. As with the data term, a robust function is needed to ensure that object boundaries are not over smoothed, leading to a total variation algorithm (Cohen 1993; Werlberger et al. 2009). This is also applied in other image processing problems, such as denoising (Rudin et al. 1992) or image segmentation (Mumford and J. Shah 1989). The term $E_S(u, v)$ for the energy Eq. (3.22) is:

$$E_S(u, v) = \int_{\Omega} \Psi(\|\nabla u\|^2 + \|\nabla v\|^2) d\mathbf{x} \quad (3.27)$$

If more than one image is available, the flows can be forced also to be 'temporally' smooth (Volz et al. 2011). Normally, temporal smoothness increases the system complexity (both in CPU and memory usage), so the usual approach is to process two frames at a time.

3.2.1.1 Energy Minimization

Minimization of the functional in Eq. (3.22) can be done by solving the corresponding Euler-Lagrange equations (Fox 1950) with homogeneous Neumann boundary condi-

tions:

$$\frac{\partial E_D}{\partial u} - \lambda \left(\frac{d}{dx} \frac{\partial E_S}{\partial u_x} + \frac{d}{dy} \frac{\partial E_S}{\partial u_y} \right) = 0 \quad (3.28)$$

$$\frac{\partial E_D}{\partial v} - \lambda \left(\frac{d}{dx} \frac{\partial E_S}{\partial v_x} + \frac{d}{dy} \frac{\partial E_S}{\partial v_y} \right) = 0 \quad (3.29)$$

The details of the numerical scheme for the minimization are not the main concern for the purpose of this thesis. However, an overview of the warping theory follows. Two main problems arise when minimizing Eq. (3.22). First, the energy equation is not convex with respect to (u, v) , so the minimization can be stuck in a local minima. Second, to resolve this variational problem, the set of partial differential equations (PDE) should be linearized with respect to (u, v) . The linearization relation of the data Eqs. (3.23), (3.25) only holds for small values of (u, v) . To cope with large displacements, the flows (u, v) are computed incrementally, by constructing a coarse-to-fine image pyramid. The construction of this pyramid deals with the non-convex nature of the problem by splitting the flows (u, v) into a known part (u_0, v_0) and a small, unknown part (du, dv) . Initially, at the coarsest level $(u_0, v_0) = (0, 0)$, and the flows (du, dv) are computed at each level by warping an image to the other (Brox, Bruhn, et al. 2004; Black and Anandan 1993). Since (du, dv) are relatively small at each scale, the linearization of the data term equation holds, and the Euler-Lagrange equations can be solved using a sparse linear system.

3.2.1.2 Complimentary information for flow estimation

Although many optical flow estimation approaches are variants of Eq. (3.22), there are some worth mentioning due to their relevancy. With the popularity rise of SIFT (Lowe 2004), SURF (Bay et al. 2008) and HOG (Dalal and Triggs 2005) descriptors, many researchers used dense versions of these descriptors to find a motion field. (Brox and Malik 2011) combines both local (color) information and large displacement matchings coming from HOG descriptor matchings. This allows the algorithm to account for large displacements regardless of the regularization term, although sometimes failures in descriptor matching lead to incorrect motion estimations. The energy to minimize becomes:

$$E(u, v) = E_D(u, v) + \lambda E_S(u, v) + \lambda_d E_M(u, v) \quad (3.30)$$

Where λ_d controls the influence of E_M , E_M is the penalty term coming from descriptor matchings. The numerical minimization is similar to (Brox, Bruhn, et al. 2004). In

this work, we use (Brox and Malik 2011) to estimate optical flow, as many sequences presented large displacements which could be captured by descriptor matching.

3.2.2 Motion Occlusions

Once the basis of the optical flow algorithms are known, it is much easier to understand how motion occlusions can be estimated. Using three frames I_{t-1}, I_t, I_{t+1} , it is possible to detect pixels becoming invisible from I_t to I_{t+1} , called *occluded pixels* and pixels becoming invisible from I_t to I_{t-1} called *disoccluded pixels*. Here, we describe the detection of occluded pixels as the detection of disoccluded pixels can be done similarly by working on the past frame I_{t-1} instead of the next frame I_{t+1} . There is a lot of literature on occlusion estimation. Many approaches choose to integrate the occlusion detection to the motion estimation process (Sun et al. 2010; Leordeanu et al. 2013; Ayvaci et al. 2010), but this increases a lot the computational burden. Instead, by using simple error measures, see Eqs. (3.31)-(3.34), it is possible to get rather accurate occlusion estimates. Below a review the most common ways of the state of the art to detect if a pixel \mathbf{p} becomes occluded is given. Consider the following definitions:

$$\text{Endpoint Error} = EE(\mathbf{p}) = |\mathbf{w}^{t,t+1}(\mathbf{p}) - \mathbf{w}^{t+1,t}(\mathbf{p} + \mathbf{w}^{t,t+1}(\mathbf{p}))| \quad (3.31)$$

$$\text{Angle error} = AE(\mathbf{p}) = |\pi - \cos^{-1}(\mathbf{w}^{t,t+1}(\mathbf{p}) \cdot \mathbf{w}^{t+1,t}(\mathbf{p} + \mathbf{w}^{t,t+1}(\mathbf{p})))| \quad (3.32)$$

$$\text{Photoconsistency} = PC(\mathbf{p}) = |I(\mathbf{p}) - I(\mathbf{p} + \mathbf{w}^{t,t+1}(\mathbf{p}))| \quad (3.33)$$

$$\text{Flow variation} = FV(\mathbf{p}) = |\nabla \mathbf{w}^{t,t+1}(\mathbf{p})| \quad (3.34)$$

where $\mathbf{w}^{t,q}$ is the flow mapping frame t to frame q . The four previous measures give real-valued error measures, and detection is often performed by setting a threshold. The first two Eqs. (3.31) and Eq. (3.32) rely on the bijective property of the optical flow: if a point is visible in both frames t and $t + 1$, then there should be a one-to-one mapping of flow fields. In other words, the forward and backward should be equal in magnitude with complementary angles, compensating each other. The third Eq. (3.33) relies on the suppositions of the optical flow estimation, where motion tends to relate pixels having the same color in both images and penalizes high color differences. The last method in Eq. (3.34) assumes that occlusions occur in motion edges.

More simple occlusion classifiers/detectors can be designed using multiresolution, textures or descriptors for example; but these four methods represent the main ideas behind occlusion detection:

- In non-occluded areas flow must be bijective.

- Flow must relate the same point in both images
- Occlusions occur near motion edges

Alternatively, some approaches choose to combine multiple simple occlusion estimators to provide more reliable occlusion detectors (Humayun et al. 2011). Other approaches use filtering to refine optical flow and detect occlusions (Xiao, Cheng, et al. 2006). These approaches, along with the ones integrating occlusion and motion estimation perform costly operations. In the case concerning this thesis, we choose to exploit segmentation with the simple classifier of Eq. (3.31)

3.2.2.1 Occluded and disoccluded pixel estimation

There are two principal problems of using raw optical flow to estimate occlusions. First, motion edges do not correspond with color edges due to the regularization term in Eq. (3.22). Second, estimated flow in occluded areas is not reliable (Mac Aodha et al. 2013). Although the optical flow estimation allows sharp transitions between different motion, the effect of the regularization appears in zones where small detail is present. That is, in image corners or small regions, flow estimation tend to oversmooth the flow regardless of color estimation. This regularization serves, among other things, to mitigate noise effects but comes with the price of missing details at different resolutions.

A possible workaround to make color and motion edges coincide is to rely on segmentation. Generally, frame/video partitions are obtained using color information so, it is expected that partition regions coincide with color edges. Segmentation can also use motion, as we will see in Secs. 6.2 and 7.2 but color proved to be the most confident information to distinguish between objects. Additionally, get a reliable flow in occluded areas a parametric flow model is robustly fit to the different region forming the partition of the frame. Although the way in which segmentation of frames and sequences are obtained will be explained in Secs. 6.2 and 7.2, suppose from now on that for occlusion detection a partition P of the frame at time t is available.

The key idea is to detect occlusions by allowing a small error on the endpoint error compensation in Eq. (3.31). A pixel becomes (dis)occluded if:

$$\Lambda(\mathbf{p}) \neq \Lambda(\mathbf{p} + \tilde{\mathbf{w}}^{t,t+1}(\mathbf{p}) + \mathbf{w}^{t+1,t}(\mathbf{p} + \tilde{\mathbf{w}}^{t,t+1}(\mathbf{p}))) \quad (3.35)$$

Where Λ is an operator which maps each pixel to a region label of the partition. $\tilde{\mathbf{w}}^{t,q}(\mathbf{p})$ is a parametric flow model fitted to $\mathbf{w}^{t,q}(\mathbf{p})$ in the region where \mathbf{p} belongs. In other



Figure 3.12: Three examples of occlusion detection. The left image shows the original frame, the center left image shows the modeled flows for a given partition. The two right images show the occlusion relations superposed to the image of the methods in Eq. (3.31) and Eq. (3.35). Occluded and disoccluded areas are shown in red, while their occluding pixel is shown in green. Note how the second method obtains clearer boundaries and much less false alarms.

words, the forward and backward consistency is relaxed, marking a pixel occluded only if their compensating flows end up in a different region than the original pixel. In that way, we make motion occlusions and region boundaries coincide, as can be seen in examples of Fig. 3.12. The parametric flow model for each region is obtained using the following approach.

Parametric flow fitting To get a region-based modeling of the optical flow, a parametric projective model (Kanatani 1988) is used. The flows $\tilde{w}_{R_i}^{t,q} = (\tilde{u}, \tilde{v})$ with $q = t \pm 1$, associated to region R_i can be expressed as a quadratic model on the x and y coordinates:

$$\tilde{u}(x, y) = a_1 + a_2x + a_3y + a_7x^2 + a_8xy \quad (3.36)$$

$$\tilde{v}(x, y) = a_4 + a_5x + a_6y + a_7xy + a_8y^2 \quad (3.37)$$

3. MONOCULAR OCCLUSION CUES

where $(x, y) \in R$. The $a_1 \dots a_8$ parameters are estimated with robust regression using iterative least squares (Andersen 2008) due to the presence of outliers:

$$\tilde{\mathbf{w}}_{R_i}^{t,q}(\mathbf{p}) = \arg \min_{\tilde{\mathbf{w}}^{t,q}} \sum_{\mathbf{p}=(x,y) \in R_i} \Psi \left(\|\mathbf{w}^{t,q}(\mathbf{p}) - \tilde{\mathbf{w}}^{t,q}(\mathbf{p})\|^2 \right) \quad (3.38)$$

with the robust penalizer $\Psi(z) = \sqrt{z^2 + \epsilon^2}$ with $\epsilon \ll 1$. The approximation of optical flow with a quadratic parametric model assumes that objects are planar and rigid, moving and rotating in 3D space. To prove this, assume that a rigid object is moving with velocity defined as:

$$\begin{bmatrix} V_x \\ V_y \\ V_z \end{bmatrix} = \begin{bmatrix} 0 & -\omega_3 & \omega_2 \\ \omega_3 & 0 & -\omega_1 \\ -\omega_2 & \omega_1 & 0 \end{bmatrix} \begin{bmatrix} X_t \\ Y_t \\ Z_t \end{bmatrix} + \begin{bmatrix} T_x \\ T_y \\ T_z \end{bmatrix} \quad (3.39)$$

Where $\boldsymbol{\omega} = [\omega_1, \omega_2, \omega_3]$ is the angular velocity vector and $\mathbf{T} = [T_x, T_y, T_z]$ is the linear velocity. Assuming that the scene is generated under perspective projection, the image coordinates $(x, y) = (\frac{X}{Z}, \frac{Y}{Z})$ and the velocities on the image plane (i.e the optical flow) can be written as:

$$\begin{aligned} u &= f \frac{V_x}{Z} - x \frac{V_z}{Z} = f \left(\frac{T_x}{Z} + \omega_2 \right) - \frac{T_z}{Z} x - \omega_3 y - \frac{\omega_1}{f} xy + \frac{\omega_2}{f} x^2 \\ v &= f \frac{V_y}{Z} - y \frac{V_z}{Z} = f \left(\frac{T_y}{Z} + \omega_1 \right) - \omega_3 x - \frac{T_z}{Z} y - \frac{\omega_2}{f} xy + \frac{\omega_1}{f} y^2 \end{aligned} \quad (3.40)$$

where f is the focal length of the camera, and has no relevance on the model complexity. The previous equation shows that the optical flow velocities have a direct dependence on the depth of the object Z , being the apparent motion smaller as Z is larger. This result agrees with the perception of the motion parallax cue, where objects far away ($\frac{1}{Z} \approx 0$) appear to move much slower than objects near the viewer ($\frac{1}{Z} \gg 0$). Nevertheless, this kind of relation is not useful for the depth ordering estimation, as Z is basically the value that should be estimated (and thus unknown). However, if the object is assumed to be planar, that is:

$$AX + BY + Z + C = 0 \quad (3.41)$$

it is possible to remove the dependence of (3.40) in depth by setting $Z = (D - AX - BY)^{-1}$ and express equations (3.37) only as a function of $\boldsymbol{\omega}$, \mathbf{T} and A, B, C . The exact dependence of a_1, \dots, a_8 with respect these parameters is not relevant for the purposes of this discussion, but the reader is referred to (Kanatani 1988) for more details. The

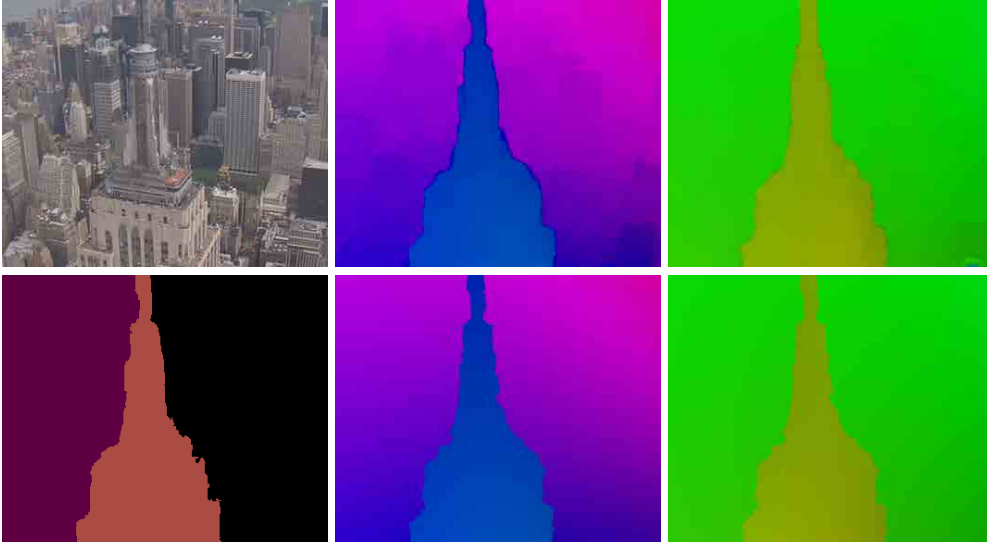


Figure 3.13: Example of flow fitting example for three regions. From left to right, the top row shows the reference frame and the forward and backward flows. The bottom row shows a partition of the frame and the two modeled flow fields. Each modeled flow field contains three parametric flow models (one for each region).

key idea is that, normally, non-planar objects which are far from the camera position can be approximated as planar. In these cases, the error of the flow model and the estimated raw optical flow become negligible.

In optical flow estimation techniques it is common to deal with a high amount of outliers, so the usual least squares estimation of the quadratic model may fail in some cases. Therefore, the $a_1 \dots a_8$ parameters are estimated with robust regression using iterative least squares (Andersen 2008):

$$\tilde{\mathbf{w}}_{R_i}^{t,q}(\mathbf{p}) = \arg \min_{\tilde{\mathbf{w}}^{t,q}} \sum_{\mathbf{p}=(x,y) \in R_i} \Psi \left(\|\mathbf{w}^{t,q}(\mathbf{p}) - \tilde{\mathbf{w}}^{t,q}(\mathbf{p})\|^2 \right) \quad (3.42)$$

with the robust penalizer $\Psi(z) = \sqrt{z^2 + \epsilon^2}$ with $\epsilon \ll 1$. Examples of flow fitting can be seen in the right part of Figure 3.13.

Once the occluded pixels have been defined, we need to find the *occluding pixels*, which correspond to the pixels that will cover the occluded pixels in the next frame. Indeed, it is the relation between occluded and occluding pixels that provides a depth cue. Of course, a similar detection has to be done for disoccluded and disoccluding pixels.

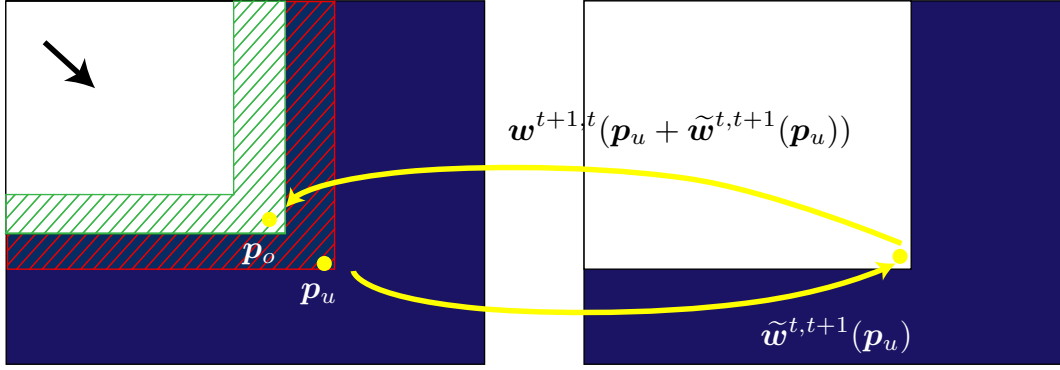


Figure 3.14: Detection of occluding pixels (green area). The image on the left (right) is I_t (I_{t+1}). See text for more details on occlusion relation estimation.

3.2.2.2 Occlusion relation estimation

With the partition P_f defined and a parametric optical flow model is available for each region, occluding pixels can be defined by projecting the occluded pixels in I_{t+1} with $\tilde{w}^{t,t+1}$ and by getting back to the current frame following the backward flow $w^{t+1,t}$. This is illustrated in the right part of Fig.3.14 where occluding pixels appear in the green area. So, for each occluded pixel p_u , the corresponding occluding pixel p_o is given by:

$$p_o = p_u + \tilde{w}_{R_i}^{t,t+1}(p_u) + w^{t+1,t}(p_u + \tilde{w}_{R_i}^{t,t+1}(p_u)) \quad (3.43)$$

From Eq. (3.35), it follows that $\Lambda p_o \neq \Lambda p_u$. Therefore, although Eq. 3.43 establishes an occlusion relation between pixels, the relation can be propagated to regions Λp_o and Λp_u . Λp_o is the occluding region, and thus closer to the viewer. The central image of Fig.3.13 also shows these occluding pixels in green. At this point, we know that occluding pixels are in front of occluded pixels and similarly, disoccluding pixels are in front of disoccluded pixels.

3.2.2.3 Occlusion detection performance

To compare the various methods of occlusion detection available in the literature, experiments performed in (Humayun et al. 2011) are reproduced. The datasets consists of 11 synthetic sequences with annotated groundtruth occlusions. Pixels going out of the screen due to movement are also considered occlusion in the annotations, but since they are not really occluded, they are discarded here for the evaluation. Hence, only occlusion between objects which remain in the camera field of view are consid-

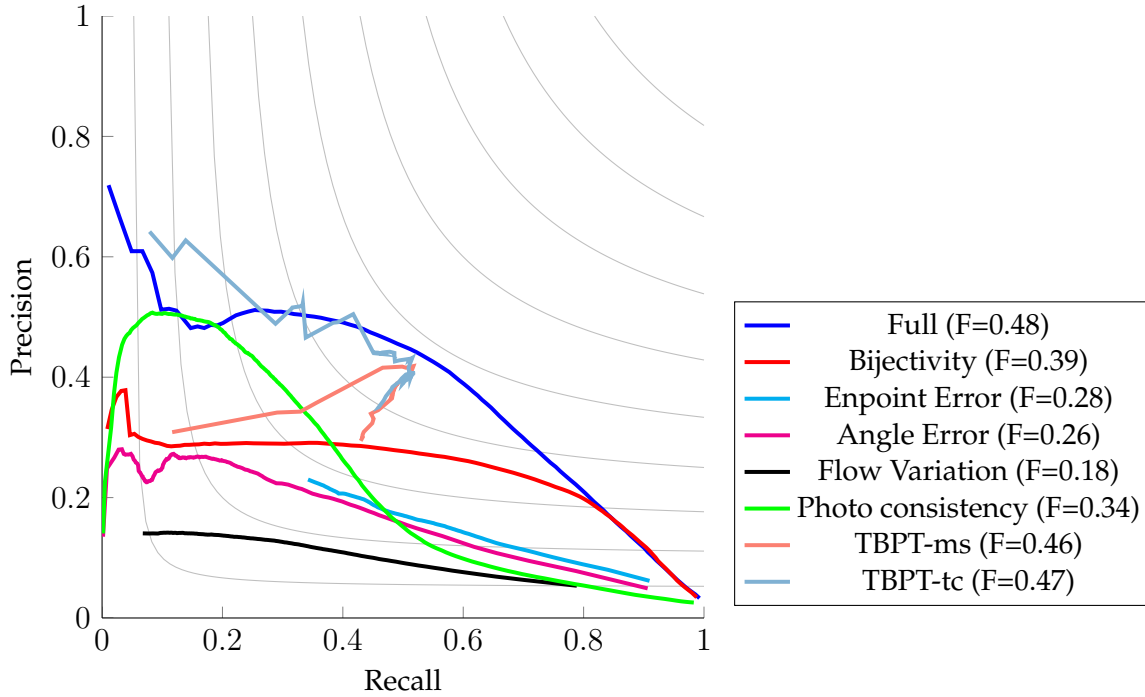


Figure 3.15: Precision recall of occlusion detection. The dataset is taken from (Humayun et al. 2011). See text for details on the compared methods.

ered. The performance measure is the well known precision-recall framework used for detection. In this case, the units to be detected are occluded pixels.

the chosen methods for comparisons are the ones from Eqs. (3.31)-(3.34) as well as the full (Full) and the simple classifier (Lean) proposed in (Humayun et al. 2011). For the performance of the region-based approach in Eq. (3.35), regions are obtained by constructing a Trajectory Binary Partition Tree (TBPT) for each sequence and extracting two sets of segmentations for each sequence. For details on how to construct the TBPT and how to obtain segmentations from it, see Sec. 7.2. For the moment, assume that the two sets of segmentations are given and they create two precision recall curves named TBPT-ms and TBPT-tc.

Quantitative results are shown in Fig. 3.15 showing that detecting occlusions using regions performs as well as the Full method from (Humayun et al. 2011). The main drawback about using regions is that the obtained result is extremely dependent on the partition obtained. Nevertheless, if the proper segmentation is found, results are competitive with state of the art best techniques. Qualitative results are shown Fig.

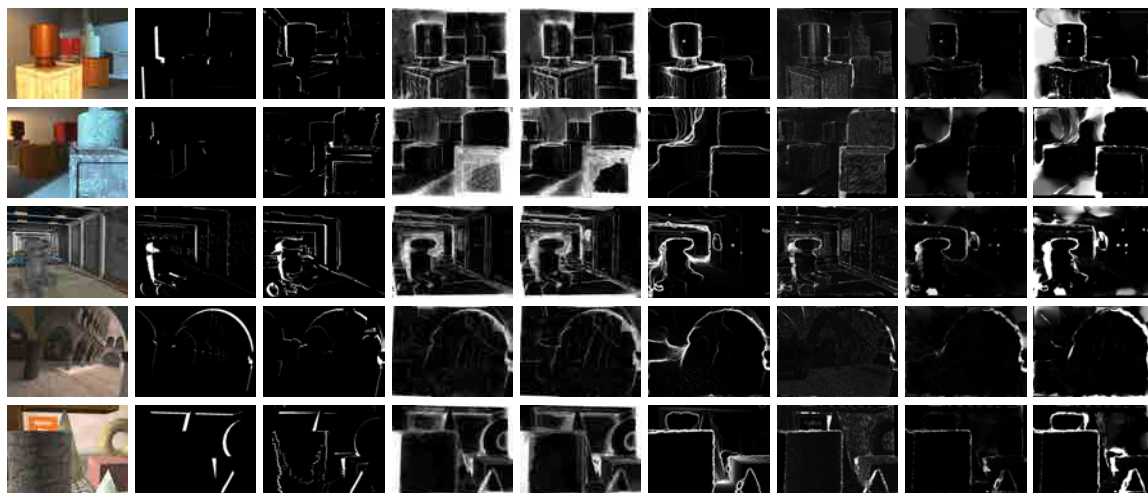


Figure 3.16: Five cases of the occlusion dataset from (Humayun et al. 2011). From left to right, for each column where likely occluded pixels are brighter. Reference frame, groundtruth occlusions and results using the proposed region approach in Eq. (3.35) with TBPT-tc partitions, the Full and Lean methods from (Humayun et al. 2011), and simple methods: flow variation, photo consistency, angle and endpoint error.

3.16. Although the Full classifiers performs slightly better than the proposed occlusion detection mechanism, its computation is extremely heavy. For example, the classifier involves computing four different types of optical flow algorithms at two different resolutions. The occlusion classifier is based on a feature space of approximately 250 dimensions. Clearly, as shown Fig. 3.15, with much less overhead comparable results can be found, so the proposed region approach is used in this thesis.

The monocular depth cues exposed in this chapter will be integrated into a whole system to estimate depth order maps. Either T-junction, convexity or motion occlusion provide a local information on the depth gradient between regions but, sometimes, local depth relations disagree with the global depth structure. To assess the quality of the generated depth order maps, an evaluation framework should be designed. The next section is devoted to present a new depth ordering dataset and contributions on the evaluation of depth ordering algorithms.

4 Evaluation Methodology

4.1 Integrating detection and classification problems

Depth ordering or figure/ground estimation problems are problems which assign an order to a set of detected regions or contours. This problem is normally divided in two steps: 1) a segmentation or contour detection and 2) figure/ground or depth order assignment. Many state of the art algorithms, see Sec. 5.1 decouple the problem and decide to evaluate only the second part. Nevertheless, it is likely that both steps are strongly related, as suggest results in (Ren et al. 2006) or (B. Liu et al. 2010) where much better figure/ground scores are obtained with perfect segmentations. It is then logical to try to evaluate both steps at the same time. For instance, Fig. 4.1 shows an image with its groundtruth depth order along with four possible outcomes of four different depth ordering systems. Which one is better? The answer is not trivial, as the user may sacrifice some segmentation quality so as to obtain correct depth relations or vice versa. A quantitative evaluation is needed in these cases.

These kinds of problem also arise in other fields, such as structured prediction with latent variables (Kumar et al. 2010). In these problems, latent variables are not explicitly modeled but they are key to the performance of the system. In the case of (Kumar et al. 2010) the latent variable is the object localization and the output of the system is its classification into some specific class (human, animal...). Therefore, the problem is conceptually the same: 1) detection of object location and 2) object class classification. In this case, the better the object localization is, the better the classifier will perform. As detection and classification performance can be very correlated, it is interesting to have an evaluation framework capable of capturing all the information. To this end a Precision-Recall-Classification (PRC) framework is proposed in the following sections.

Contributions on Performance Measures

- G. Palou and G. Salembier. “Precision-Recall-Classification Evaluation Framework: Application to Depth Estimation on Single Images”. In: *Submitted to CVPR*. 2014



Figure 4.1: Depth ordering evaluation problem. From left to right: original image, ground-truth depth order and four results of depth ordering systems. A part from the second result, deciding which is the best result is not an easy task.

4.1.1 Detection Problems

In detection problems, systems are designed to decide whether a given event or feature (object, contour, activity... etc) is present or absent in a given space. Given a ground truth annotation, a desirable system behavior is to detect all possible entities without giving any false alarms. Quantifying a system performance is normally done in a framework where true/false positives/negatives are combined to provide precision and recall:

- True positives (TP): events detected by the system and marked as positive on the groundtruth
- True negatives (TN): events not detected neither by the system nor annotated on the groundtruth
- False Positives (FP): events detected by the system but not annotated on the groundtruth
- False Negatives (FN): events not detected by the system but marked as positive on the groundtruth.

From these four quantities, precision arises as the fraction of correct detections with respect all the detections. Recall is the fraction of detected events among the groundtruth. Formally, they can be expressed as:

$$Precision = \frac{TP}{TP + FP} \quad Recall = \frac{TP}{TP + FN} \quad (4.1)$$

The perfect score for a system is when both precision and recall are 1, although normally there is a compromise between these two quantities. That is, a system that has

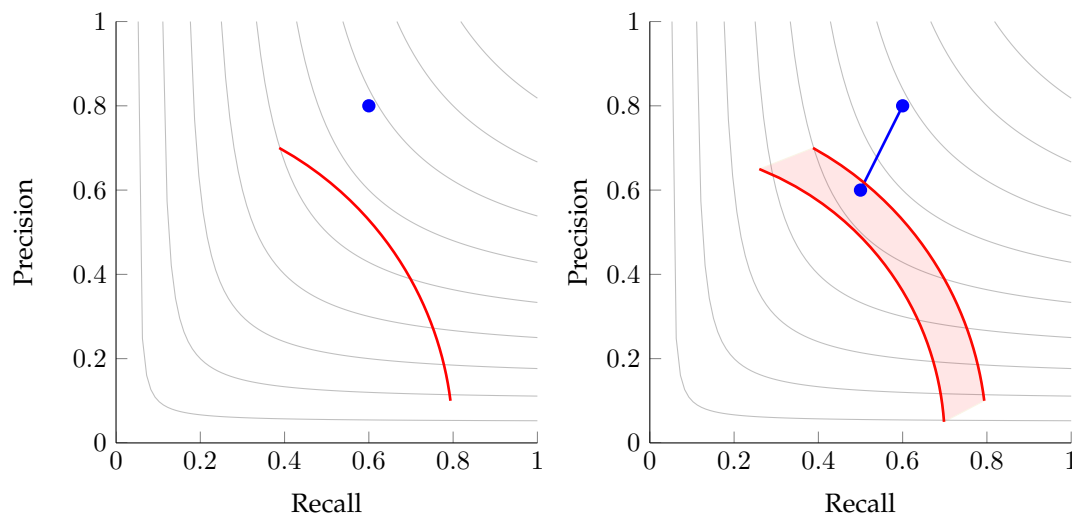


Figure 4.2: Operating regions of two different algorithms (red and blue) for classic Precision-Recall (left) and Precision-Recall-Classification (right) frameworks. Gray lines indicate points with the same F measure.

a high recall is also likely to have false positives, and a system which is very precise is likely to miss some true annotations. Often, the two quantities are summarized into a single number F , defined as the harmonic mean of precision and recall:

$$F = \frac{2PR}{P + R} \quad (4.2)$$

Normally system performance is plotted on a precision-recall plane. A particular output of a system is a point in this plane, although it is normal to see operating curves if a system depends on a given set of parameters θ . Therefore precision and recall of its output also depends on θ , i.e. $P(\theta)$ and $R(\theta)$. This gives a set of points on the precision recall plane which normally are represented as a curve, see Fig. 4.2.

Although the number F may give a hint to compare two systems and see which of the two has a better behavior, different points on the plane may give the same F measure, regardless of their precision and recall values. If a curve is present, the system performance is normally given by its maximum F measure along the curve. This point is considered to be the operating point of the system.

In this thesis we propose an extension of the detection problem which includes a binary classification among the detected objects. The objective to do so is to integrate both detection and classification into the precision-recall framework. We apply this

		\emptyset	1	
			A	not A
1	A	FD	CD	ID
	not A	FD	ID	CD

Table 3: Confusion matrix of the proposed classification framework. \emptyset indicates no detection, while 1 indicates a detection. A and not A are the possible outcomes of the classifier. TN , MD and FD stand for true negatives and missed detections. The other concepts are defined in the text.

extension to depth ordering evaluation, but results can be applied to a variety of problems.

4.1.2 Combining Detection with Binary Classification

In classification problems, results are often represented with confusion matrices, where the miss-classification rate is observed among different classes. If ground-truth results are available, the classifier performance can easily be computed. However, if classified objects should be first detected by an algorithm, it is likely that the classification score will depend on the operating point of the detection system. For instance, if only confident detections are considered (low recall, high precision), a high classification score is likely to be obtained. On the contrary, if many detections are retrieved, (low precision, high recall), the classification performance is likely to be worse. To integrate the detection and classification problems, we introduce two concepts:

- Inconsistent Detection ID : a correct detection that has been erroneously classified.
- Consistent Detection CD : a correct detection that has been properly classified.

All possible combinations of system output and ground-truth annotations are shown in Table 3. Similarly to pure detection scores, these measures are combined to provide precision-recall measures. CD and ID should be interpreted with care. Note that IDs , although not desirable, are in some way “better” than miss-detections MD or false detections FD since a correct detection is present and a post-processing step may correct the classification. Let us consider two extreme cases of the evaluation scenario:

Pure detection system. In this scenario, we ignore the classification and consider an outcome to be correct if the detection is correct. In this approach CD and ID are equivalent and $TP = CD + ID$, $FP = FD$ and $FN = MD$.

Pure classification system This scenario considers that an outcome is correct if and only if detection and classification are correct. Hence, one should consider that TP are only correctly detected events when the same classification than ground-truth is produces. In this context, $TP = CD$ while ID should be interpreted in two ways:

- Detecting an incorrect class is equivalent to detect an event/object that does not exist. Therefore $FP = FD + ID$.
- Detecting an incorrect class leaves a ground-truth result without correct detection. Therefore $FN = MD + ID$.

To consider a scenario in-between these two extremes, a parameter $0 \leq \beta \leq 1$ is introduced to regulate the compromise between segmentation-classification quality. In this way, it is possible to redefine:

$$TP(\beta) = CD + \beta ID \quad (4.3)$$

$$FP(\beta) = FD + (1 - \beta)ID \quad (4.4)$$

$$FN(\beta) = MD + (1 - \beta)ID \quad (4.5)$$

$$(4.6)$$

Therefore precision (P) an recall (R) are redefined as:

$$P(\beta) = \frac{CD + \beta ID}{CD + \beta ID + FD + (1 - \beta)ID} = \frac{CD + \beta ID}{CD + ID + FD} = C_p + \beta I_p \quad (4.7)$$

$$R(\beta) = \frac{CD + \beta ID}{CD + \beta ID + MD + (1 - \beta)ID} = \frac{CD + \beta ID}{CD + ID + MD} = C_r + \beta I_r \quad (4.8)$$

With $C_p = \frac{CD}{CD+ID+FD}$ and $I_p = \frac{ID}{CD+ID+FP}$ are the consistent and inconsistent precision respectively. C_r and I_r are defined similarly as consistent and inconsistent recalls. As shown in Fig. 4.2, each depth ordered partition establishes a line segment on the precision-recall plane by changing the β value between 0 and 1. If the algorithm to be evaluated depend on a set of parameters θ , evaluation results in a region in the same plane. To differentiate these measures with pure detection system, we will refer to them as Precision-Recall-Classification (PRC) framework.

The PRC plot of Fig. 4.2 gives insight about the system performance. Ideally, a system should reach $P(\beta) = R(\beta) = 1$ for all β values. Real systems however present

a compromise between precision and recall. In the PRC framework, there is an additional compromise corresponding to the width of the operating region. A wide region indicates poor system performance in classification ($I_p, I_r \approx 1$), while a thin region ($I_p, I_r \approx 0$) indicates that the system is a good classifier. Moreover, as the operating point of the detection system detects only confident event/objects (low recall), the region width is expected to decrease, as classification is easier. Based on this framework, examples PRC measures are proposed in the next section.

4.2 Two PRC frameworks on Depth Ordering

A special case of detection plus classification problem is depth ordering. In this scenario, the scene should be segmented, and each region should be ordered according to its relative depth. The segmentation step can be considered as the detection stage of the system (contours need to be detected) and the classification step corresponds to the correct assignment of relative depth. In this section, the PRC framework will be applied to the depth ordering problem, by exploring two different perspectives for evaluation. A first PRC framework is designed for local depth relations in detected contours, where the local depth gradient is evaluated. This measure is called Local Depth Consistency (LDC). A second evaluation, called Global Depth Consistency (GDC), is proposed by looking at the global structure of the image, by examining all pairs of detected regions. The following two sections explain more thoroughly both measures.

4.2.1 Local Depth Consistency

We extend the original bipartite matching for contour evaluation ([D. R. Martin et al. 2004](#)) to include the classification step as follows. Bipartite matching is used for contour detection evaluation, and finds a one-to-one mapping between detected and groundtruth contours. See [Fig. 4.3](#) for a graphical explanation of the process. From the resulting matched elements, true/false positives/negatives are found and a precision/recall measures are computed to evaluate the detection score of a system. Here we extend this bipartite matching to include the depth gradient on detected contours.

In depth ordering, even if a contour is detected correctly, it can still be consistent with the groundtruth depth order (assigning figure/ground correctly to both sides) or inconsistent. Originally proposed in ([Ren et al. 2006](#)), the performance of a figure/ground (f/g) classification algorithm is measured with two steps:

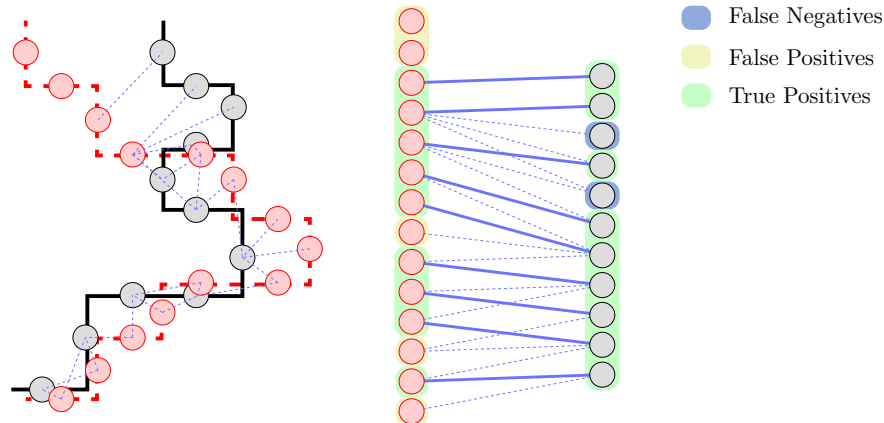


Figure 4.3: Bipartite matching on contours. The left figure shows two contours, the dashed red corresponds to the detected one, and the black corresponds to a groundtruth. Detected contour locations (red circles) are associated with groundtruth contour locations (black circles) if they are close enough. On the right, a bipartite graph is constructed and a maximum one-to-one matching is performed, giving rise to true positives (matched contours, solid lines) and false positives (non-matched detected contours), as well as false negatives (groundtruth contours not matched).

1. Bipartite matching on the contours of P_S with contours of P_G (Arbeláez et al. 2011).
2. Measure the f/g classification accuracy only on detected contours.

That is, the final f/g score is the classification accuracy of the boundary recall. The main problem with this measure is that it completely ignores the quality of the segmentation, leading to biased results if only confident contours are detected. As stated in (M. R. Maire 2009), the f/g assignment on confident contours is easier than the assignment on ambiguous ones. Therefore, if the system only outputs the most confident contours, the f/g score could be biased towards higher performance. In other words, there exists a compromise between the quality of the segmentation and the f/g labeling problem which, to this day, has not been fully addressed. In (M. R. Maire 2009) a first step is proposed by evaluating the f/g score versus the boundary recall, showing that, effectively, there exists a compromise between these two values. However, this approach is only sufficient to show the performance for a single system. Using the PRC framework it is possible to assess both contour detection and depth gradient classification at the same time. The original matching scheme (D. R. Martin et al. 2004) is

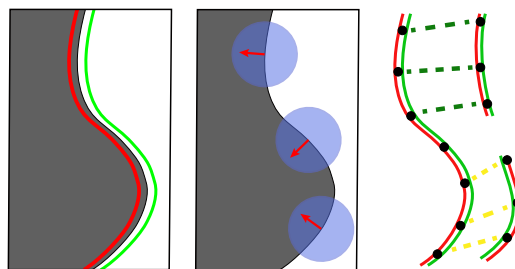


Figure 4.4: Process to evaluate the local depth consistency. Left: Depth partition with one contour. The green overlay indicates the figure side, and the red overlay indicates the ground side. Center: Contour normals are estimated by averaging local orientations. Right: Bipartite matching of the groundtruth contour (right) and detected contours (left). The figure shows consistent and inconsistent matchings, in green and yellow respectively, due to the incorrect estimation of the depth gradient.

modified to measure inconsistent matchings, A brief explanation illustrated in Fig. 4.4 follows:

1. From the depth partition, figure and ground sides are identified by examining the depth of each region.
2. The orientation of the depth gradient is estimated by averaging contour normals within a local window.
3. Bipartite matching of ground-truth and detected contours: CD and ID are marked with green and yellow grayed lines respectively. A matching is inconsistent if the orientation of the depth gradient exceeds a specified threshold (15°).

It is important to note that ID s, although not desirable, are in some way 'better' than misdetections MD or false detections FD since a correct contour location is detected and, eventually, a postprocessing step could correct the depth gradient. As in (D. R. Martin et al. 2004), false detections FD are the detected contours with no correspondence in the groundtruth, and missed detections MD are contours in the groundtruth with no correspondence in the detected contours. This measure will be referred as Local Depth Consistency(LDC) as it measures local depth relations on contours.

4.2.1.1 F/G Over Random Index

As in detection systems, it is always desirable to summarize the performance of a system in a single number for comparison. In pure detection systems, this 'sum-

mary measure' is done by means of the F-measure. Nevertheless, summarizing a two-dimensional (precision-recall, PR) space into a single dimension may lead to poor interpretation of results. For instance, systems having different operating points of precision and recall may give the same F-measure while outputting very different results.

In the proposed framework, as shown in Fig. 4.2, the system may output a different PR-curve C_β for each value of the parameter β . For each β , it is possible to compute the curve maximum F-measure, F_β . The detection and classification performance of the system may be characterized by the pair:

$$F_{min}, F_{max} = \min_{\beta} F_{\beta}, \max_{\beta} F_{\beta} \quad (4.9)$$

The higher both numbers are, the better the systems detects. Additionally, the smaller the difference $\Delta F = F_{max} - F_{min}$ is, the better the system classifies. Nevertheless, the main drawback about ΔF is that the classification performance depends directly on the precision-recall point giving F_{max} . For example, a system having $\Delta F = 0.2$ and $F_{max} = 0.8$ has a much better performance on classification than a system with $F_{max} = 0.5$ and $\Delta F = 0.2$. Given this situation, it can be convenient to present a classification measure that does not depend on the operating point.

Classification Measure According to equations (4.7) and (4.8) precision and recall are divided into their consistent and inconsistent subparts. Consider a contour detection system S and two classification systems working on the detections of S : an intelligent system S_i and one random system S_r . In S_i depth gradients are assigned using some sort of reasoning, while in S_r the depth gradient is assigned randomly.

Assume the operating point of S_i has a given set of CD^i and ID^i . If S_r has the same operating point on detection, the chance of assigning a correct depth gradient is 50% and, therefore $CD^r = ID^r = \frac{CD^i + ID^i}{2}$. It is possible to show with (4.7) and (4.8) that the precision, recall and F measure (P_r, R_r, F_r) of a random system are related to their counterparts (P_i, R_i, F_i) of a system with the same detection score by:

$$P_r = (1 + \beta) \frac{P_i}{2} \quad R_r = (1 + \beta) \frac{P_i}{2} \quad F_r = (1 + \beta) \frac{F_i}{2} \quad (4.10)$$

Showing that when $\beta = 1$ all measures are the same (since no classification is considered). On the contrary, when $\beta = 0$, the three scores of a random system are divided by two. Therefore, the system behavior can be assessed by comparing it to its theoretical

random point:

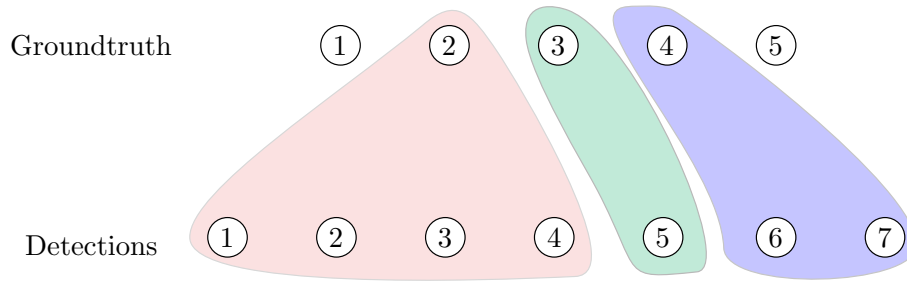
$$ORI = \max\left(0, \frac{F_{min}^i - \frac{F_{max}^i}{2}}{\frac{F_{max}^i}{2}}\right) \quad (4.11)$$

Since contours have variable lengths and F_{min}^i can be lower than $\frac{F_{max}^i}{2}$, the maximum operation is done to ensure positive scores. Therefore, a perfect classification system should have $ORI = 1$ while a random-like system will have $ORI = 0$.

4.3 Global Depth Consistency

When estimating depth maps or figure/ground, it is important that the whole depth map is consistent given a ground truth. That is, the global depth structure of the image should coincide both in the estimated and the groundtruth, even if the contours do not match perfectly. Even if two parts of the image are not spatially adjacent, the relative orders between these regions should be coherent with the groundtruth and thus, pleasant to an external observer. Therefore, a non local measure that quantifies the global depth consistency is desirable. Since contour localization is not always reliable and regions offer a more coarser view of the scene; regions can be used to localize zones in the image with different depths. To this end, a region based precision-recall framework similar to the LDC is designed. The proposed name is Global Depth Consistency (GDC) as the measure takes non-local relations into account and relates all the regions in the image, whether they are adjacent or not (unlike the LDC).

Assume the output of the system is a partition P_S formed by a set of N_S regions $S = \{S_i\}$ and the groundtruth data is also a partition P_G of the image with N_G regions $G = \{G_i\}$. Unlike contours, regions by themselves do not incorporate the notion of relative order. However, if we consider pairs of regions, the notion of depth transition arises naturally. Since these pairs of regions do not necessarily need to be adjacent (unlike contours, which delimitate two spatial adjacent regions), evaluating all pairs of regions leads to a global depth interpretation of the estimated P_S with respect to P_G . Denoting the relative depth order of regions S_i, G_i as Δ_i^S, Δ_i^G the following measures are designed to provide a PRC framework for global depth ordering. The process is detailed in the following lines and a short example is given afterwards following Fig. 4.5. As with contours, detected regions should be matched with groundtruth regions. A common way to perform this matching is to compute the Jaccard index for each G_i



	1	2	3	4	5
1	Grey	Red	Red	Red	Red
2	Grey	Grey	Green	Green	Red
3	Grey	Grey	Grey	Green	Red
4	Grey	Grey	Grey	Grey	Red
5	Grey	Grey	Grey	Grey	Grey

	G_1	G_2	G_3	G_4	G_5
Depth	1	2	3	4	3

	1	2	3	4	5	6	7
1	Grey	Blue	Blue	Blue	Green	Yellow	Green
2	Grey	Grey	Blue	Blue	Green	Green	Yellow
3	Grey	Grey	Grey	Blue	Yellow	Yellow	Yellow
4	Grey	Grey	Grey	Grey	Yellow	Yellow	Yellow
5	Grey	Grey	Grey	Grey	Grey	Green	Green
6	Grey	Grey	Grey	Grey	Grey	Grey	Blue
7	Grey	Grey	Grey	Grey	Grey	Grey	Grey

	S_1	S_2	S_3	S_4	S_5	S_6	S_7
Depth	1	2	3	4	3	2	1

Figure 4.5: Example of region matching. Top figure: Each Detected region is matched to a groundtruth region. In case of subsegmentation, there may be groundtruth regions which are not matched. In case of oversegmentation, the same region can be matched multiple times. Bottom figure: Two tables showing all possible groundtruth pairs (left) and detected region pairs (right). Red squares count as MD , blue count as FD , green as CD and yellow as ID . In See the text for a more extended explanation.

against all S_j and taking the maximum:

$$m(S_i) = \tilde{G}_i = \arg \max_{G_i} \frac{S_i \cap G_j}{S_i \cup G_j} \quad \forall S_i, G_j \quad (4.12)$$

When matchings have been done, FD are detected as the number of detected pairs of region, with different depth values, that are matched with the same ground-truth region:

$$FD = \sum_{S_i, S_j \in S} (1 - \delta(\Delta_i^S, \Delta_j^S)) \delta(\tilde{G}_i, \tilde{G}_j) \quad (4.13)$$

where $\delta(a, b) = 1$ if $a = b$ and 0 otherwise. False detections contain, intuitively, the number of false depth transitions within groundtruth regions. It is very similar to the concept of FD in the LDC measure, where a false alarm is detected if a non-existent true contour is detected by the algorithm. Missed detections (MD) are the total number of missed transitions due to the region matching process in equation (4.12). Let the set of matched groundtruth regions be \tilde{G} , the formal expression for MD is:

$$MD = \frac{|G|(|G| - 1)}{2} - \frac{|\tilde{G}|(|\tilde{G}| - 1)}{2} \quad (4.14)$$

where $|\cdot|$ denotes the set cardinality. This measure can be explained by following the example in Fig. 4.5. If G_1 is missed ($\tilde{G}_i \neq G_1 \forall i$), a total of $(|G| - 1)$ transitions are missed $(G_1, G_2), (G_1, G_3), (G_1, G_4), (G_1, G_4)$. If an additional region G_5 is missed, the number of missed transitions are $(|G| - 2)$, so no missed relation is counted twice. This mechanism gives rise to Eq. (4.14) and in the extreme case when $|\tilde{G}| = 1$, $MD = \frac{|G|(|G|-1)}{2}$ (note that at least one region will always be detected).

The last two quantities to define are consistent and inconsistent detections. CD and ID are found by examining each pair G_i, G_j and averaging the pairs of detected regions with the same and different depth order respectively. Intuitively, CD and ID for a pair G_i, G_j measures how, in average, the detections are consistent with the groundtruth depth. This is done because for very oversegmented partitions, $|S| \ll |G|$ and while FD and MD grow linearly with $|G|$, the number of pairs of detected regions grow quadratically with $|S|$. Define α_{ij} and β_{ij} the number of consistent and inconsistent matches for a pair S_i, S_j respectively. $\gamma_{ij}^{G,S} = \text{sgn}(\Delta_i^{G,S} - \Delta_j^{G,S})$ is an indicator of the order of the regions i, j in the sets G, S . Then, α_{ij}, β_{ij} is expressed as:

$$\alpha_{ij} = \sum \delta(\gamma_{kl}^S, \gamma_{ij}^G) \quad (4.15)$$

$$\beta_{ij} = \sum 1 - \delta(\gamma_{kl}^S, \gamma_{ij}^G) \quad (4.16)$$

Both summations are performed over the regions S_k, S_l fulfilling $m(S_k) = \tilde{G}_i$ and $m(S_l) = \tilde{G}_j$. The final consistent and inconsistent measures are given by:

$$CD = \sum_{G_i, G_j} \frac{\alpha_{ij}}{\alpha_{ij} + \beta_{ij}} \quad (4.17)$$

$$ID = \sum_{G_i, G_j} \frac{\beta_{ij}}{\alpha_{ij} + \beta_{ij}} \quad (4.18)$$

To understand the GDC evaluation works, the example of the Fig. 4.5 is explained more thoroughly. There are:

- *MD*: Groundtruth regions G_1 and G_5 have been missed. As a result, 7 transitions are missed: $MD = 7$ as shown by red squares on the left table.
- *FD*: S_1, S_2, S_3 and S_4 have been matched with the same groundtruth region G_2 . As a result 6 false transitions $(S_1, S_2), (S_1, S_3), (S_1, S_4), (S_2, S_3), (S_2, S_4), (S_3, S_4)$ are detected: $FD = 6$ as shown by blue square on the right table. Additionally, S_6 and S_7 are both matched to G_4 , creating 1 additional *FD*, thus $FD = 7$.
- *CD, ID*: Consistent and inconsistent relations can be resumed into the following:
 - The groundtruth transition (G_2, G_3) on the left table generates the following inconsistent detections: $(S_3, S_5), (S_4, S_5)$ and consistent detections: $(S_1, S_5), (S_2, S_5)$. For this particular groundtruth transitions: $CD = \frac{2}{4}$ and $ID = \frac{2}{4}$
 - The (G_2, G_4) transition on the left table generates is more complicated. It generates 8 transitions, of which 2 are consistent detections: $(S_2, S_6), (S_1, S_7)$ and 6 are inconsistent: $(S_1, S_6), (S_3, S_6), (S_4, S_6), (S_2, S_7), (S_3, S_7), (S_4, S_7)$. This gives : $CD = \frac{2}{8}$ and $ID = \frac{6}{8}$.
 - The groundtruth transition (G_3, G_4) on the left table generates 2 detected transitions. Both transitions (S_5, S_6) and (S_6, S_7) are consistent with the ground truth so, $CD = \frac{2}{2}$ and $ID = \frac{0}{2}$

Summarizing, the total number of consistent detections is $CD = \frac{2}{4} + \frac{2}{8} + \frac{2}{2} = 1.75$ and the number of inconsistent detection is $ID = \frac{2}{4} + \frac{6}{8} + \frac{0}{2} = 1.25$. Using $MD = 7$ and $FD = 7$, the numeric expression for the PRC in this case can be computed using Eqs. (4.7) and (4.8):

$$P(\beta) = \frac{CD + \beta ID}{CD + ID + FD} = \frac{1.75 + 1.25\beta}{1.75 + 1.25 + 7} = 0.175 + 0.125\beta \quad (4.19)$$

$$R(\beta) = \frac{CD + \beta ID}{CD + ID + MD} = \frac{1.75 + 1.25\beta}{1.75 + 1.25 + 7} = 0.175 + 0.125\beta \quad (4.20)$$

4. EVALUATION METHODOLOGY

In this particular example, both expressions $P(\beta)$ and $R(\beta)$ are the same because $MD = FD = 7$, but this is generally not the case. In practical situations, the GDC is much more restrictive than LDC because it considers not only local relations but other non-adjacent depth transitions. Therefore it is expected that precision-recall values to be lower than in the LDC measure. Once presented the evaluation framework suitable for segmentation and depth ordering, we are now going to discuss state of the art depth ordering systems as well as the details of the scheme developed in the context of this thesis.

5 Depth Ordering in Still Images

5.1 State of the Art

Converting monocular content to 3D to some extent has been an important objective for many industrial actors such as Microsoft ([Ward et al. 2011](#)), Disney ([Wang et al. 2011](#)) or Prime Focus (a post-production company for Hollywood Studios) with View-D software ([Bond 2011](#)). Even with current technology, the film/photograph industry keeps shooting in normal monocular cameras due:

Financial costs Both the camera acquisition and the post production process are more expensive in the stereo/multiview case than the traditional monocular case.

Technical difficulties Disparity cues are only valid within a limited range of viewing distances. So large field of view shots are not suitable for stereo vision and only close views could benefit from two points of view.

Artistic motifs Like shooting in black and white, this only depends on the objective of the author of the visual content. Some directors/photographers do not find the added value of 3D to worth the increase on complexity, and prefer to shoot in monocular.

These three reasons are the key to consider monocular 2D to 3D conversion a field of interest, regardless of its proven difficulty and ill-posed condition. To the date, monocular depth systems are not able to estimate a perfect depth map, but, in practice, a rough representation may suffice for humans to perceive a three dimensional effect ([Hubona et al. 1999](#); [Phan et al. 2011](#)).

Monocular depth estimation in still images started in the 70s with the so called shape-from-shading algorithms. The starting point was ([B. K. Horn 1970](#)), where simple objects were reconstructed in illumination controlled conditions. In these algorithms, objects were supposed to be formed by a Lambertian surface, where light is scattered depending on the incidence angle, and surface does not have many texture. After this initial work, many other appeared, but the principle was the same: estimate depth from a single cue. A survey on classical shape from shading algorithms can be found in ([R. Zhang et al. 1999](#)). A worth to mention reference is the work ([Barron and Malik 2013](#)), where objects are presented in uncontrolled environments with unknown light

position, shading surface model and object texture. From an image, the reflectance, shading and texture maps are recovered from the object.

Other kind of depth-from-X include depth from defocus/focus, perspective and texture. They can be compared to shape from shading methods in the sense that they only exploit a single depth cue in controlled environments. Defocus assumes that some objects on the scene are outside the depth of field of the camera so they appear blurred. The original idea to exploit focus and defocus to retrieve depth was proposed in (Pentland 1987) and (Darrell and Wohn 1988). Absolute depth is achieved when two or more images with different focus points are used due to unknown camera calibration parameters. Nevertheless, approximate results can be obtained using one single image (Zhuo and Sim 2009; Ghita et al. 2005). The principle of shape/depth from texture algorithm is to observe that textured surface exhibit different patterns of their texture depending on their relative orientation and position with respect to the camera. Early algorithms tested the validity of the texture gradient cue on simple texture patches (Aloimonos 1988). More recent approaches also tried to recover surface orientation on natural images (Super and Bovik 1995), but since the interpretation of natural environments is much more difficult, the problem is still open.

Due to their restrictions, these systems work only in environments where the background and the illumination are controlled and with isolated objects. Research only recently tackled the problem in natural common images taken with cameras. Depth from perspective is probably the bridge between the approaches estimating depth using controlled environments and approaches attempting to recover the structure of general and natural scenes. Since perspective cues arise from the projection of parallel lines to the image plane and the creation of vanishing points, they were initially used

Contributions on Depth Ordering on Single Images

- G. Palou and P. Salembier. "Occlusion-based depth ordering on monocular images with Binary Partition Tree". In: *IEEE ICASSP*. Prague, Czech Republic, 2011
- G. Palou and P. Salembier. "From local occlusion cues to global depth estimation". In: *IEEE ICASSP*. Kyoto, Japan, 2012
- G. Palou and P. Salembier. "Monocular Depth Ordering Using T-junctions and Convexity Occlusion Cues." In: *IEEE Trans. on Image Proc.* 2013



Figure 5.1: Examples of the figure/ground groundtruth annotations for the BSDS300. Original images are shown in first, third and fifth column. Groundtruth contours in second, fourth and sixth column mark with white strokes the figural side, and with black strokes the background.

to estimate surface orientations on scenes with marked structures such as buildings, interior offices or any kind of human construction (Criminisi et al. 1999). With minimal information of the scene, the authors are able to recover the scene layout and produce accurate 3D environments from single images. Posterior works (Delage et al. 2006) are able to recover depth without any prior information, just by looking at perspective cues.

Although depth estimation in human built environments appeared in the late part of the 90s, it wasn't after some years that computer vision addressed depth estimation in natural scenes. Nevertheless, the firsts works only estimated a general structure (Torrvalba and Oliva 2002) such as predominant (or mean) depth of the scene. That is, it was only possible to estimate whether the observed scene involved close or distant objects, but not its specific structure. Subsequent works (M. G. Ross and Oliva 2010) also assessed the estimation of prominent scene features from a perceptual point of view.

Specific object structure in natural images was not tackled until the Berkeley group released its segmentation dataset BSDS300 in the work (D. Martin et al. 2001) where the depth gradient at object contours was annotated, see Fig. 5.1. Although closely related, estimating the image depth is not exactly the same as assigning depth gradient at contours, which is also known as figure/ground (f/g) labeling. There are two main differences between depth ordering and f/g labeling:

- Depth ordering should be globally consistent with the observed cues, while the contour depth gradient can be unrelated with other contours.
- Figure/ground problems do not need partitions (closed contours) to obtain their result. Depth ordering systems need a partition of the image to assign depth to regions.

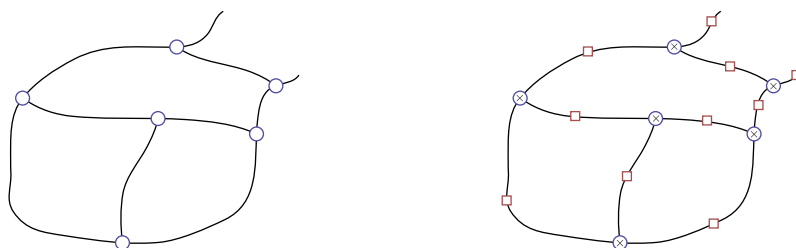


Figure 5.2: Graphical model of the CRF globalization process for the f/g assignment in (Ren et al. 2006). The left figures shows possible contours found in an image with junctions marked as points where three or more contours meet. The right figures shows the associated graph of the CRF. The depth gradient are the variables to find, which are the red squares on edges.

Moreover, depth ordering system are somewhat more flexible in the sense that they can estimate surface orientation and semantics on the scene. Figure/ground systems are, nevertheless important as they can be a first step to a global depth structure estimation. In this direction, the scheme (Zhao and Davis 2004) proposed an iterative approach to segregate an object from its background. A more recent work (Ren et al. 2006) assigns f/g labels to contours based on local decisions followed by a global optimization step with conditional random field (CRF) (Lafferty et al. 2001). The used local depth cues are clustered contour shapes, called *shapemes* (Berg and Malik 2001). They intrinsically encode low level vision cues such as parallelism and convexity. Technique (Ren et al. 2006) is one of the first that exploits the perceptual statement that the depth gradient at junctions should be determined by a global reasoning rather than with junction intrinsic characteristics. The CRF framework proposed in the cited paper allows to arrive at a consistent f/g labeling of the depth gradient by enforcing junction consistency and local depth cues.

It is worth detailing the globalization process of the mentioned algorithm, as many of the following works use a similar approach. The algorithm tries to find a labeling on the edges based on three classes, denoted by a random variable $X_e = -1, 0, 1$ which encodes the direction of the depth gradient (0 acts as null gradient). From a set of contours, a graph is constructed, see Fig. 5.2 and probabilistic inference on X_e is run on the graph so as to maximize a likelihood based on local depth estimations and junction compatibility. Although this approach offers still state of the art perfor-

mance, junctions are estimated by propagating contour direction and it may assign depth gradients to spurious and unrelated contours.

Other works in f/g labeling followed, such as (Leichter and Lindenbaum 2009) and (M. Maire 2010). The first work uses explicit depth cues such as lower region (Vecera et al. 2002), parallelism and T-junctions to build a CRF to estimate discrete depth labels on both side of the edges. Although the algorithm ends up estimating f/g labels, it makes use of depth order information to force global consistency. Its performance outperforms (Ren et al. 2006), but it uses more cues and makes some assumptions on the scene structure, such as the lower regions are closer to the viewer. Moreover, its performance decreases rapidly when no perfect segmentation is available. The work from (M. Maire 2010) performs segmentation and figure/ground assignment jointly using angular embedding (Yu 2009) and a local convexity classifier to estimate the depth gradient at each contour. To obtain a global consistency, the algorithm makes use of the normalized cut machinery (Shi and Malik 2000) to obtain a set of complex images where the contour strength and f/g direction are encoded in the real and imaginary parts respectively. The main drawback of this approach is that there exists a compromise in cases when segmentation and f/g cues disagree. In such cases, the algorithm favors one or the other depending on the value of a parameter which is not easily tuned.

However, one of the advantages of (M. Maire 2010) is that it contains a 'figural likelihood' for each pixel in the image and not only in contours. That is, it obtains a functions which directly inform about the depth order of each pixel. From the raw output, the authors obtain a depth order partition by applying the Ultrametric Contour Maps (UCM) from (Arbelaez et al. 2009) and assigning the depth order of each region the mean 'figural likelihood' of its pixels, see results in Fig. 5.3.

Prior works attempted even to recover the absolute distance (up to a factor) from a single point of view. This is the case from (Saxena et al. 2005) and (B. Liu et al. 2010). The first work oversegments the image and then gathers features for each generated superpixel. It then uses a learning algorithm to assign an absolute depth value to each superpixel, also forcing a global consistency with a random field. A part from the examples given by authors, the algorithms generalizes poorly to different types of scenes, see Fig. 5.3.

Avoiding over-training on a given type of scenes was tackled by (D. Hoiem et al. 2011). In this work, authors propose to use the surface layout classifier (Hoiem et al. 2007) to classify the different types of regions in the scene as either horizontal, vertical, porous



Figure 5.3: Results of the main state of the art methods. In the left column, the original image is showed. From right to left: groundtruth depth annotation, method of (Saxena et al. 2005), (M. Maire 2010), (D. Hoiem et al. 2011) and (Calderero and Caselles 2013). In the depth maps, closer region are encoded in brighter colors

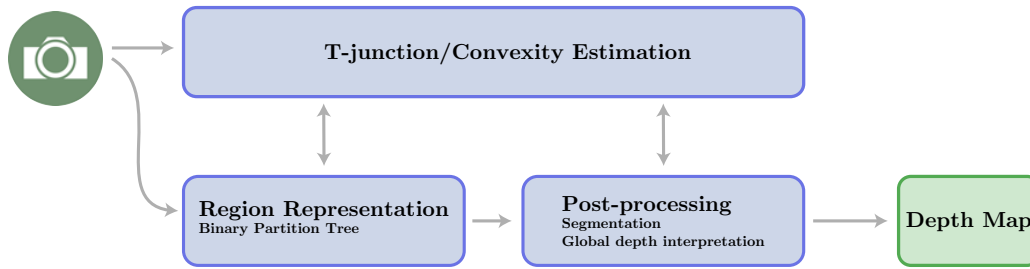


Figure 5.4: System architecture for single images. This figure is a particularization of the general scheme in Fig. 2.14.

surfaces or sky. From this classification and other low level features, a CRF is used to construct a global consistent depth map. The advantage of this method is that it can produce smooth depth gradients when surfaces are not parallel to the camera view plane, although surface misclassification leads to poor depth estimations. Nevertheless, the algorithm which is originally intended to detect occlusion boundaries, presents good performance on contour detection and is one of the best methods of the state of the art on occlusion boundary detection.

High level information can be very informative about the scene structure but, to the date, scene recognition (Xiao, Hays, et al. 2010) and surface description (Gould et al. 2009) are unsolved problems. Therefore, one way to assess the scene structure is to base the algorithm reasoning on low level vision processes. Such processes, stated by Gestalt as in 2.2, allow to infer some depth relationships from local descriptions of the image. Either by explicitly detecting depth cues as in (Dimiccoli 2009) or by implicitly reasoning about occlusion (Calderero and Caselles 2013), these approaches estimate local depth relations which are propagated afterwards to the whole image to provide a global depth ordering. Propagation in these cases is done with a non-linear diffusion filter (Buades et al. 2006). The main drawback of these kind of iterative filters is that the final outcome is highly sensitive to the number of iterations so, this parameter may sometimes need to be controlled manually. Since in (Calderero and Caselles 2013) reasoning is produced at the pixel level, the authors note that the presence of noise and edge blur may disturb the final result. Nonetheless, results in both systems show that low level features are useful to recover global depth maps, see Fig. 5.3.

In this thesis, we are going to assess all the potential weaknesses of the previously mentioned approaches. Three different systems are proposed based on the same prin-

ciples:

- There is no scene a priori knowledge. No assumptions can be made about the type of the scene so that the system can accept any kind of input.
- Low level cues are used instead of high level information. Working with low level information allows to have more cues of depth than simple semantic priors which strongly condition the final structure estimation.
- Region based image representations allow to deal, among other things, with blur and noise without preprocessing the image. Moreover, if the region representation is organized in a hierarchical structure, information appearing at different scales can be encoded naturally.

The general system architecture was shown in Fig. 2.14. Here we show a particularization for the proposed systems for single images 5.4. From an input image signal, a region representation by mean of a Binary Partition Tree (BPT) is built and then post-processed to obtain a suitable segmentation. During this process, low level cues such as T-junctions and convexity are estimated and used to infer a consistent global depth ordering. The following section reviews the state of the art of first stage of the system: hierarchical region representations of images.

5.2 Hierarchical Representation of Images

5.2.1 State of the Art

In most image processing applications, an image is viewed as a set of pixels placed on a planar grid. This low level and unstructured representation only offers the possibility to use simple algorithms due to the large number of pixels composing the whole image. Moreover, the representation does not describe the spatial composition and does not provide support to easily handle semantic notion. In the recent years, there has been an increasing interest to consider the image as a set of superposed regions. Thus, a region-based representation has to be computed from the pixel level.

Since the information in an image may be present at different scales, the image representation should be able to deal with different levels of detail. This characteristic is obtained by constructing a hierarchical set of regions. To generate such regions, two main approaches handle the problem either with a top-down or a bottom-up perspective, see Fig. 5.5. The former initially considers the image as a unique region and splits

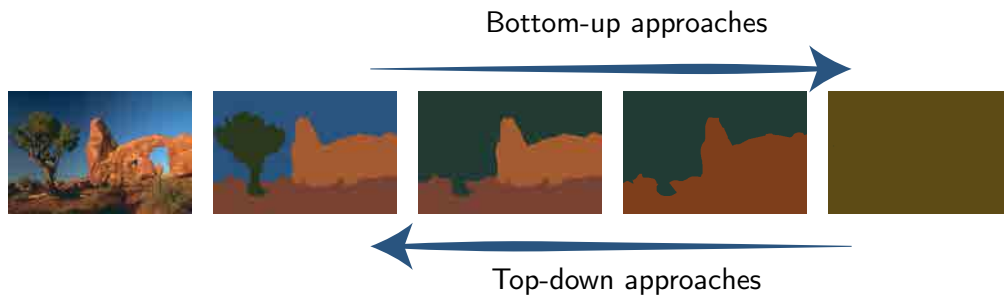


Figure 5.5: Differences between bottom-up and top-down approaches. From the original image on the left, bottom up approaches successively merge small regions to create bigger ones. On the contrary, top-down approaches split bigger regions to create finer partitions.

iteratively the newly created regions to obtain the final partition. The latter considers the pixels as a starting point and the final regions will grow from these initial seeds. Among the literature, examples of these systems can be found in (Colantoni and Laget 1997; Tremeau and Colantoni 2000; Pardas et al. 1996).

Some hierarchical segmentation methods extend their “flat” homologue by successive application of the algorithm to produce coarser partitions at each step. For example, (Paris and Durand 2007) extends the Mean-Shift (Comaniciu and Meer 2002), (Cour and Benezit 2005) applies spectral clustering across scales and (Grundmann et al. 2010) extends the Efficient-Graph Based method (Felzenszwalb and Huttenlocher 2004) to produce hierarchies for video sequences. Although these approaches actually produce a hierarchy of regions, they generate very unbalanced trees in which a parent region can have a variable number of child nodes. Sometimes a given structure is desirable, and as such, methods (P. Salembier and Garrido 2000; Alpert et al. 2007; Arbeláez et al. 2011) propose to generate binary trees to represent an image.

The simplest form of a tree, known as the Binary Partition Tree (BPT), was proposed in (P. Salembier and Garrido 2000) and proved to be fast and efficient. The BPT is a structured representation of the image regions that can be obtained from an initial partition using a simple bottom-up merging approach. At each iteration, two adjacent regions are merged to form a parent region containing the two merged ones. The pair of regions to be merged are chosen according to a similarity measure. When the BPT is constructed, the leaves of the tree represent the regions belonging to the initial par-

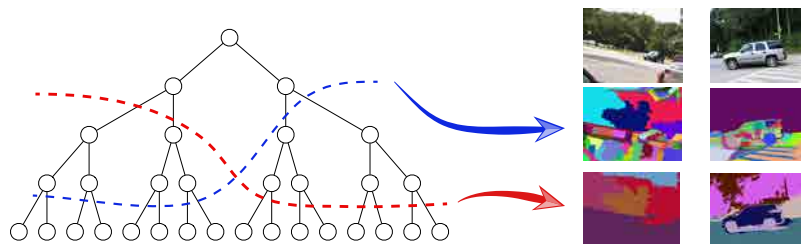


Figure 5.6: Example of different prunings of a BPT. Depending on which nodes are selected (blue or red lines) the produced partition is different. Selecting one as a segmentation for a particular application can be done using tree cuts, see Sec. 5.3

tion and the root node refers to the entire image. The remaining tree nodes represent the intermediate regions formed due to the merging process used to construct the BPT. Prior to BPT definition, an image model is needed. Basically, the image model is the pixel type, also known as color space. To construct a BPT, four region concepts must be presented:

Models: Since the BPT is a region-based representation, the region and image models should be clearly defined. A region by itself is a set of 4-connected pixels. Common choices to represent regions are color and contour information, but these may vary for each implementation.

Adjacency: Adjacent regions are region sharing at least one pixel edge. Edges are considered to be the points between two pixels. Region adjacency is then defined as 4-connected. There may be other possibilities, such as an 8-connected adjacency, but they are not considered in this work.

Hierarchy: The hierarchy of regions is defined as the parent region to be the union of its two child regions. This *parent* relationship can be extended to more than one level, relating a node with all of its descendants and vice versa. This is a key concept for a BPT because the hierarchical organization allows to look at the image with different resolutions.

Similarity: A metric should be defined to compare adjacent regions. This metric will vary depending on the chosen region models and should somehow measure the similarity between two regions.

Generally, as the hierarchy cannot be found by a global minimization of a given criterion, bottom up approaches proceed iteratively to build the tree. At each iteration, the two most similar neighboring regions are merged, creating a new parent region. The parent region preserves the adjacency relations of both sons with other regions. This process is repeated until only one region representing the whole image is left. This strategy proved to give the best result until the date on image segmentation. It should be noted that the tree is not a segmentation of the image per se, but only a region based representation which contains many segmentations. Formally, any subtree maintaining the original root can be translated to its corresponding segmentation. That is, the leaves of a BPT or a pruned BPT represent a partition as shown in Fig. 5.6. Since BPTs are constructed by iteratively optimizing a local criteria, a post-processing step generally helps to retrieve better partitions than the ones generated by the merging sequence. This fact has been observed by several works such as (P. Salembier and Garrido 2000), (Serra et al. 2012) and (Y. Xu et al. 2012) for example. In the following sections, two different strategies to create a BPT of the image are presented and in Sec. 5.3 a general overview of techniques to retrieve better partitions from trees are discussed in Sec. 5.3.

5.2.2 The Monocular Depth BPT

5.2.2.1 Color Space

When considering the image color space, several choices are possible. Usually, images are stored using the *RGB* color space but processing within this space is not the optimal choice. The *RGB* model presents high correlation between channels, so redundant information is processed three times. To eliminate this spectral redundancy, another color space may be used. The two more popular ones are the *YUV* and the *CIE Lab*, (Paschos 2001).

Although both color spaces are suitable to deal with uncorrelated channels (luminance and chrominance) there is an important difference: the *Lab* color space is perceptually defined. The numeric distance in the *RGB* or the *YUV* colorspace do not correspond to the visual color distance seen by humans: two pixels with different *RGB* or *YUV* values may seem equal to the human eye (Sharma 2002). The *Lab* color space was designed to solve this problem: the numerical distance between two colors is directly proportional to the perceptual difference between this pair of colors.

The difference between pixel values is defined as the euclidean distance between color vectors. Although the *CIE Lab* standard is carefully revised, there are strong criticisms,

specially about the distance and the distinguishable threshold. First, the distance was modified in (Robertson 1990) and results showed that some compensation was needed to maintain the perceptual correspondence between pair of colors. Second, the Just Noticeable Difference (JND) in (Sharma 2002) was said to be 2.3 instead of 1 as the standard (CIELAB standard colour image data 2010) proposed. In practice, these two modifications are subtle changes and they do not have much influence when comparing colors. As a result, the *CIE Lab* color space is used in this thesis to represent the color in the images.

5.2.2.2 Region Model

Generally, image regions are modeled using color characteristics. For example, in (Vilaplana et al. 2008) regions were modeled by their color mean while in (Calderero and Marques 2010) the region model was built around three mono-dimensional channel histograms. (Dimiccoli 2009) also modeled region with channel histograms, but their estimation was performed using ideas from the non-local means algorithm proposed in (Buades et al. 2005). In this thesis, a further extension is used and an adaptive multidimensional histogram is proposed which proved to give better results on segmentation benchmarks, see Sec. 5.3.

To represent the color distribution of the regions, one could choose among several possibilities. In contrast to (Calderero and Marques 2010; Vilaplana et al. 2008), the model chosen for this thesis is a single multidimensional histogram. Although 3D-histograms do not lose any color information, their representation is very costly in memory usage. As a result, it is unfeasible to work with a complete three dimensional representation.

To overcome the memory limitations, regions are modeled using a few representative colors (signatures) (Ruzon and Tomasi 2001). Following the MPEG-7 standard, 8 dominant colors are a good choice to represent a whole image (Manjunath et al. 1998). Therefore, the same number is chosen to describe each region, but depending on the region color homogeneity a lower number may suffice.

Hierarchical Signature Estimation Each color signature s_i is characterized by a set of ordered pairs $\{(p_1, c_1), (p_2, c_2) \dots (p_n, c_n)\}$ with n being at most 8. Each pair i is composed of a representative color vector c_i and its probability of appearance p_i .

Due to the hierarchical nature of the BPT regions, the most representative colors for each region may be estimated using different approaches. The challenge of finding the

representative colors can be seen as a quantization problem. From the initial image regions it is fairly easy to find illustrative colors. If the initial regions corresponds to individual pixels, their dominant color is simply the pixel color. If bigger regions are considered, dominant colors are obtained using a k-means clustering approach. But problems arise when a merging occurs. Due to the huge amount of initial regions, an approximate solution is proposed: When two regions are merged, a new signature is created for the parent region by joining the two underlying signatures. If the number of representative colors exceeds the maximum (that is 8, here), only the 8 colors with more presence (higher probabilities) are selected. If two colors i and j are very close according to $d_{ij} = (p_i + p_j) \times c_{ij}$ ($d_{ij} < 0.1$), they are merged and replaced by their weighted average. c_{ij} is defined perceptually as in (Shepard 1987):

$$c_{ij} = 1 - e^{-\frac{\Delta_{ij}}{\gamma}} \quad (5.1)$$

With Δ_{ij} being the euclidean distance between *Lab*-colors c_i and c_j . The decay parameter γ is set to 14.0 as in (Ruzon and Tomasi 2001). The proposed histogram simplification represents each region by at most its 8 most representative colors. The advantages over the color mean region model are obvious: the mean color is a particularization of the signature when at most one dominant color is allowed. Permitting more colors in the representation grants a more accurate representation of textures and thus, similarity measures with two regions having different color distributions but similar means would lead to different results. The advantages over the mono-dimensional histogram region model are twofold:

- 3D histogram exploits channel correlation. The mono-dimensional model is completely valid only when the color channels are independent. If they are not, some information about color is lost. It happens that, in the CIE *Lab* color model, luminance is indeed independent, although the two chroma channels present some relationship. Therefore, by using the full histogram, this dependence may be exploited
- a 3D histogram is generally sparse. By using an adaptive approach, the effect of noise can be reduced and only the most representative colors are stored.

5.2.2.3 Region distance

The order in which these regions are merged to build the BPT is given by a similarity measure. Usually, this measure is based on low-level features of the regions such as

color, area, or shape. In this thesis, however, depth information is also introduced to contribute to the final region distance. The overall distance between two adjacent regions R_1 and R_2 is a contribution of all these features:

$$d(R_1, R_2) = d_a \times (\alpha(1 - (1 - d_c) \times d_d) + (1 - \alpha)d_s) \quad (5.2)$$

d_a stands for the area distance. d_c and d_s are the color and shape measures respectively. α is the weighting factor between shape and color. Its value was experimentally set to $\alpha = 0.7$. d_d is the depth measure introduced to weight color distance. These four contributions (area, color, shape and depth) are considered to be key characteristics to define regions. Color is the most important feature but, the exclusive use of color distances lead to regions with unnatural shapes so a measure evaluating the region contour is introduced. In practice, objects in the real world have more or less compact and round shapes. Moreover, relevant objects in a scene present similar areas so a term addressing region size is also included. Since the goal of this thesis is to estimate depth planes, the inclusion of a depth measure attempts to differentiate different levels of depth during the BPT construction.

Color Criterion Histograms are a way to represent probability density functions (*pdf*). In this case, they are applied to the represent the color/intensity distribution of the pixels. Therefore, the problem to compare two region colors is equivalent to compare two color histograms. There is a wide repertory of measures that can be considered, ranging from L_p norms to ground distance measures like the histogram intersection (Puzicha et al. 1997). All the possible distances can be classified into two different types: bin-to-bin or cross-bin distances. The former type includes L_p norms, χ^2 distance, Kullback-Leibler and Bhattacharyya divergences among others. The key characteristic is that the distance $d(\mathbf{h}, \mathbf{g})$ of two N bin histograms $\mathbf{h} = [h_1, \dots, h_N]$ and $\mathbf{g} = [g_1, \dots, g_N]$ is computed by examining the distance at each bin locally:

$$d(\mathbf{h}, \mathbf{g}) = \tilde{d}(f(h_1, g_1), \dots, f(h_N, g_N)) \quad (5.3)$$

Where $f(h_i, g_i)$ is a function that compares two bin values. The function \tilde{d} aggregates all the local calculations by a linear combination maybe followed by an exponentiation (as in the L_p norms). In contrast to bin-to-bin distance, cross-bin distances allow (non linear) combinations of several bins to compute the final distance value.

The limitation of bin-to-bin distances is that the underlying purpose of histogram comparison is to obey some *perceptual* principles. For instance, two colors that are perceived very differently (e.g. black and white) must have a higher distance than

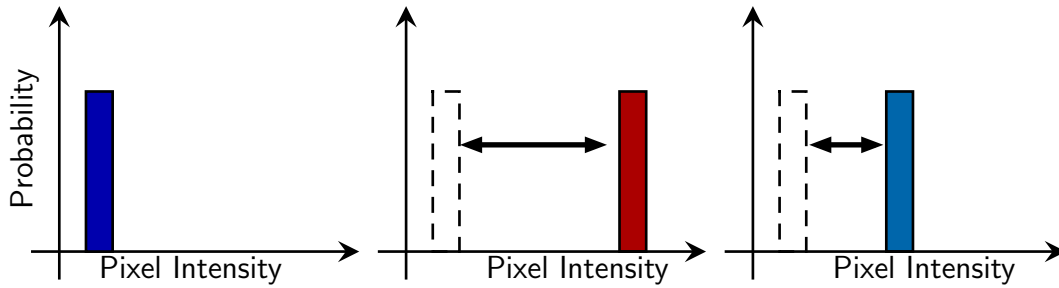


Figure 5.7: Effect of bin-to-bin distances. The second and the third histogram have the same distance with the first histogram although the colors are perceptually very different. While bin-to-bin distances say that they are equally different, cross bin distances solve this problem by assigning a bin-to-bin cost.

two similar colors (e.g blue and turquoise). This intuitive reasoning is not fulfilled in bin-to-bin distances. If, for example, an L_2 distance is computed to compare the left histogram with the center and right histograms in Fig. 5.7, the same result is obtained. Perceptually, however, the red color is much more different to the dark blue than the light blue.

This limitation can be overcome by using the so called cross-bin distances, such as the Diffusion Distance (Ling and Okada 2006) or the Earth Mover’s Distance (EMD). The Diffusion Distance measures the distance by iteratively computing the energy of the convolution of a Gaussian Kernel with the bin-to-bin histogram difference. It turns out that this process, which has the same behavior as the heat propagation in a medium, can be seen as an approximation of the EMD. The EMD was first presented as a transportation problem (Hillier and G. J. Lieberman 1990). In this thesis, it is used to compare two *pdfs*. The EMD is already presented in Sec. 3.1 in T-junction color confidence calculation but, since it is a key concept for the BPT construction, a more thorough explanation follows.

The EMD measures the amount of probability mass that has to be moved, to convert one histogram h into another g following some cross-bin unit costs. Although it is a good measure for image query, first used in (Rubner et al. 1998), it has not been used for complete image segmentation nor BPT construction. In (Ruzon and Tomasi 2001) a simplified version of the EMD is used for corner and junction detection.

The EMD distance can be defined between two signatures h and g with number of bins N_h, N_g . Each signature consists of a set of pairs (h_i, c_i^h) and (g_i, c_i^g) , where the value for bin i , h_i, g_i is defined to be the probability to observe the color c_i^h and c_i^g

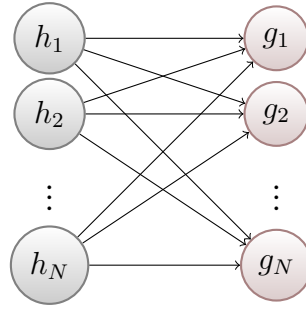


Figure 5.8: Graph representing the EMD problem. The arrows represent the flow from/to the bins and the nodes are the bins themselves

respectively. Since histograms represent a *pdf*, $\sum_{i=1}^{N_h} h_i = 1$, $\sum_{j=1}^{N_g} g_j = 1$.

The problem is how to transform histogram \mathbf{h} into \mathbf{g} . That is, some probability mass of \mathbf{h} should be displaced to form \mathbf{g} . The amount of probability mass displaced from one bin i to another bin j is represented by a flow f_{ij} . The unit cost of this displacement is given by c_{ij} . In Fig. 5.8 the graph representing the transformation of \mathbf{h} into \mathbf{g} is represented. The mathematical formulation of the EMD can be written as follows:

$$EMD(\mathbf{h}, \mathbf{g}) = \min_{f_{ij}} \frac{\sum_{i=1}^{N_h} \sum_{j=1}^{N_g} c_{ij} f_{ij}}{\sum_{i=1}^{N_g} \sum_{j=1}^{N_h} f_{ij}} \quad (5.4)$$

$$s.t. \quad f_{ij} \geq 0 \quad (5.5)$$

$$\sum_{i=1}^{N_h} f_{ij} = g_j \quad (5.6)$$

$$\sum_{j=1}^{N_g} f_{ij} \leq h_i \quad (5.7)$$

Eq. (5.5) simply states that the amount of probability mass moved should be positive. Eq. (5.6) forces that the flows going to a bin j , sum up to the value in g_j . Equation (5.7) makes sure that no more than the available probability is displaced from its original bin, h_i .

The costs c_{ij} are defined to be the costs of moving a unit of probability mass from a bin i to a bin j . Since in the concerning case the histogram bins are colors, c_{ij} are

the unit costs to transform a color c_i^h to a color c_j^g . Cross bin costs can be arbitrary positive numbers, but the usual choice is to define them to be the euclidean distance or a statistical measure. Taking advantage of the *CIE Lab* color space, the costs proposed in this scheme are perceptual. The distance between one color i from histogram h to a color j from histogram g is perceptually defined as:

$$c_{ij} = \left(1 - e^{-\frac{\Delta_{ij}}{\gamma}}\right) \quad (5.8)$$

Where Δ_{ij} is the euclidean distance between colors c_i^h and c_j^g . $\gamma = 14$ is the decay factor. The EMD is a convex optimization problem, and can be solved by linear programming algorithms such as the simplex method. Note that efficient ways to compute the EMD do exist when the costs are linear with respect to the bin distance (Ling and Okada 2006), i.e $c_{ij} \propto |i - j|$. However, since the costs defined in Eq. (5.8) are not linear, another implementation was used from (Hillier and G. J. Lieberman 1990). The EMD computation is a rather costly operation, but the use of few dominant colors on each region leads to reasonable computational times. As an important fact, the output of the EMD using the defined costs c_{ij} in Eq. (5.8) ranges in $[0; 1]$. The output 0 conforms to two completely equal regions and 1 two completely different ones.

The measure used for (5.2) for color is then

$$d_c(R_1, R_2) = EMD(\mathbf{h}, \mathbf{g}) \quad (5.9)$$

With h, g being the histograms representing regions R_1 and R_2 respectively.

Shape/Contour Criterion The contour criterion has long been introduced in image segmentation. In the early Mumford-Shah functional (Mumford and J. Shah 1989) a penalty cost on the region perimeter was introduced to find compact regions in a continuous variational framework. In (Vilaplana et al. 2008) a similar concept for BPTs was used to encourage regions to be as round as possible. The measure used was simply the increase of perimeter of the merged region with respect to the region with the largest perimeter. To adapt the contour criterion to the dynamic range of the color distance, the increase of perimeter is normalized to the largest perimeter. Define the length of the perimeters of the two regions R_1 and R_2 as P_1 and P_2 respectively. The common perimeter is $P_{1,2}$. The measure is then

$$d_s(R_1, R_2) = \max\left(0, \frac{\min(P_1, P_2) - 2P_{1,2}}{\max(P_1, P_2)}\right) \quad (5.10)$$

It is important to mention that the contour criterion should only be applied when the shapes of the regions are meaningful. In practice, the contour measure is only applied when the areas of both regions exceed a threshold (i.e. 50 pixels), but other numbers may work as well.

Area Criterion As stated above, relevant objects in the scene usually have similar sizes. It is then intuitive to introduce a measure to balance these sizes. That is, all the regions at a given iteration of the BPT construction should have approximately the same area. There exist several criteria weighting area in the BPT construction, but there is no general consensus. All of them, however, are monotonically increasing with the region areas. Here, the area contribution is defined to be

$$d_a(R_1, R_2) = \log_2(1 + \min(|R_1|, |R_2|)) \quad (5.11)$$

With $|R_1|, |R_2|$ being the respective region areas, in pixels.

Depth Criterion One of the local cues that allow to infer some depth relationships between regions are the so called T-junction points, see Sec. 2.2. These points appear where three different regions meet. To detect them in our approach, the approach of Sec. 3.1 is used. Since the proposed approach needs a local window segmentation for T-junction confidence calculation, the Region Adjacency Graph (RAG) of the BPT merging sequence is used. The RAG is a graph structure where two regions (represented by graph nodes) are connected if they share at least a common pixel edge. At each BPT construction iteration, a RAG is available. T-junctions are potentially located where a region triplet R_i, R_j and R_k is fully connected on this RAG. This also can be seen as R_i and R_j having a common neighbor R_k . The point where the three regions meet in the image, defines a possible candidate for a T-junction point. Each possible location is characterized with a probability value, measured with a confidence value, $0 \leq p \leq 1$ as explained in Sec 3.1.

Note that two adjacent regions may have more than one T-junction candidate and that each of these candidates may define two different depth planes. Fig. 5.9 shows a possible example where the regions R_i and R_j have four T-junctions in common. Note that some of the T-junctions between R_i and R_j are also T-junctions between R_i and R_k . The information relying on the T-junction candidates structure is used to modify the distance between two regions. Consider the set of T-junction candidates T between any pair of regions R_i and R_j . Following the perceptual cues exposed in Sec. 3.1, it

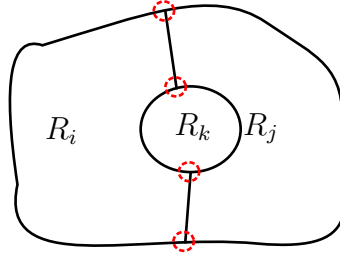


Figure 5.9: Example of two adjacent regions having more than one T-junction.

is possible to distinguish three kinds of T-junctions on the set T . Ones telling that R_i is in front of R_j , others telling R_j is in front of R_i and finally others telling that some other region is in front of R_i and R_j . Following the intuition of Sec. 3.1, where the local depth gradient of T-junctions cannot be decided with local information, a T-junction potential candidate is considered to have a normal depth gradient. That is, the region forming the largest angle is the one lying closer to the viewer.

The first two groups indicate that one of the two regions, R_i or R_j , is in a different plane than the other. The last group does not tell anything about the depth order for this pair R_i and R_j . However, for the first two groups the information is contradictory as two regions cannot be at the top of each other at the same time. Thus, one of the two suppositions (assuming a constant depth for regions) may not be true. The region depth model considered in this project assumes that no self-occlusion is present in the scene. From the confidence of each T-junction candidate, it is possible to calculate the probability that either R_i or R_j is in front of the other:

$$p^i = \left(1 - \prod_n^{N_i} (1 - p_n^i) \right) \prod_n^{N_j} (1 - p_n^j) \quad (5.12)$$

$$p^j = \left(1 - \prod_n^{N_j} (1 - p_n^j) \right) \prod_n^{N_i} (1 - p_n^i) \quad (5.13)$$

Where p_n^i is the confidence of the n -th T-junction candidate telling that R_i is in front. p_n^j is defined similarly. N_i and N_j are the number of T-junctions for each group. Let us give an intuitive interpretation of the previous expressions. The probability of R_i being in front of R_j is that at least one of the T-junctions indicating R_i is in front is true while all of the T-junctions having R_j in front are false.

With these two probabilities, the confidence difference is defined as $\delta = |p^i - p^j|$ and

the depth contribution to define the region distance is

$$d_d(R_1, R_2) = \frac{1}{1 - \delta} \quad (5.14)$$

It should be clear that $\delta = 1$ when the two values p_i and p_j differ very much and the depth order is clear and the color distance in Eq. (5.2) is increased to separate the two different depth planes. When the two values are close the modifier δ is close to zero and the color distance is not modified. This situation appears either when there are no cues that permit to order by depth the two regions or when two T-junctions give contradictory information. The intuition behind the combination of d_c and d_d during the BPT construction in Eq. (5.2) is that, the distance between two regions is increased if the junctions found between these pair of regions determine that there is a depth discontinuity. Thus, the tree is expected to be partially depth-structured.

Introducing depth distance into the BPT construction solves a ‘chicken and egg problem’. In one hand, segmentation can benefit from the knowledge of the positions of junctions, as region boundaries may not cross depth planes. On the other hand, T-junction detection and confidence estimation can benefit from segmentation as junctions appear where three regions meet. By integrating T-junction and segmentation into an iterative algorithm may help to resolve this ambiguity.

5.2.3 Ultrametric Contour Maps

Ultrametric Contour Maps (UCM) were first presented in (Arbeláez 2006) as a way to represent an image as a soft boundary map associated to a family of nested segmentations. The purpose of the original paper was not to design a new region merging approach for image segmentation, but to adapt the distance between regions so that an ultrametric property was satisfied. This change in how the distance between region was used proved to give, to the date of publication, the best results on classic segmentation datasets such as BSDS300 (D. Martin et al. 2001). The key of the algorithm is to define the distance between adjacent regions as a function of local features on the common contour. Using color and gradient contrast, the algorithm achieved state of the art performance using simple features. Later, in (Arbeláez et al. 2011) the UCM was extended to use the *gPb* contour detection, producing the best state of the art hierarchical segmentation of images. One of the proposed systems in this thesis, exposed in Sec. 5.4, include the UCM as the hierarchical representation of an image, so a brief description of the system in (Arbeláez et al. 2011) is commented in the following section.

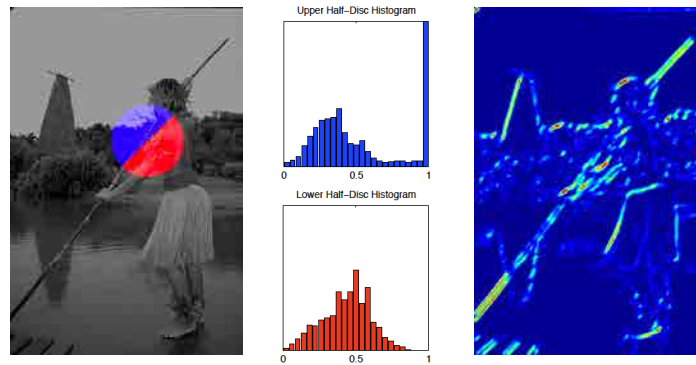


Figure 5.10: Figure extracted from (Arbeláez et al. 2011) showing the gradient computation for the brightness channel

5.2.3.1 The *gPb* contour detector

The key of the high performance of the UCM is the contour detection step, where a set of local cues at different resolutions are combined to provide global consistent contours. The process is described in the original *Pb* algorithm in (D. R. Martin et al. 2004) and it is basically a complex gradient operator on brightness, color and texture image channels. The idea is to place a disk of a fixed size at every pixel in the image and compare the histogram distribution of every channel of two disc halves using different orientations (the original implementation used 8 different angles), see Fig. 5.10.

The gradient is computed in the brightness, the two chroma, and in a texture channel. The latter channel is formed by assigning to each pixel a texton id, obtained by clustering the filter response of 17 Gaussian derivative filters with different orientations. These responses are clustered to $K = 32$ ids, and each pixel is assigned to the closest cluster center. To compare disc halves, the two distributions are compared using the χ^2 distance. The gradient computation is performed over 3 resolutions, and the responses at each pixel (x, y) are linearly combined in each gradient direction θ to form a multiscale contour detector $mPb(x, y, \theta)$.

Once a local contour strength is available for each pixel, a globalization is performed by computing pixel affinities and performing spectral clustering (Shi and Malik 2000; M. Maire et al. 2008). The difference between the spectral clustering of the normalized cuts and the *gPb* algorithm, is that the eigenvectors obtained from the spectral clustering process are not clustered. Instead, the authors propose to reshape each eigenvector

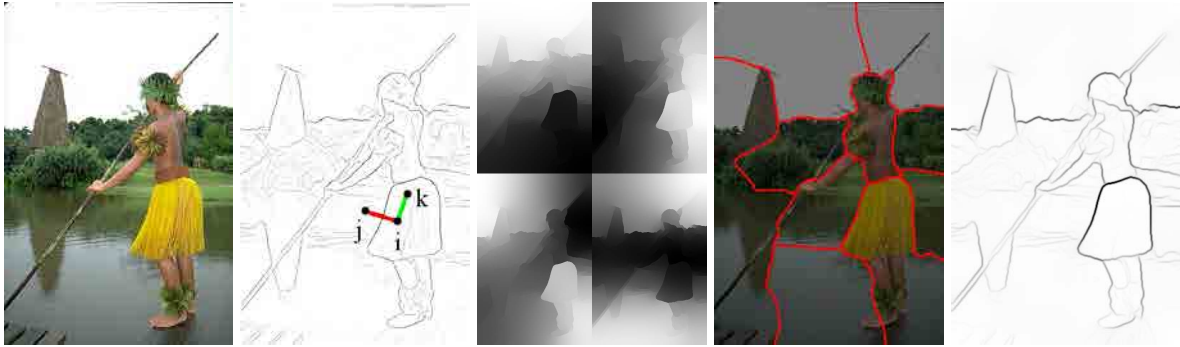


Figure 5.11: The gPb algorithm in one example image. Left: Original image. Center left: pixel affinity computation, where the green line represents high affinity and the red represent low affinity. Center: first four generalized eigenvectors of the spectral clustering process. Center right: partition of the image by clustering eigenvectors, erroneously partitioning the sky. Right: gPb signal obtained by combining eigenvectors. Image obtained from (Arbeláez et al. 2011)

to the shape of the image, compute its spatial gradient and combine them to produce a final global gradient for each orientation $sPb(x, y, \theta)$. The $sPb(x, y, \theta)$ and $mPb(x, y, \theta)$ are linearly combined to produce the global probability of boundary $gPb(x, y, \theta)$. The final contour strength of a pixel is considered to be the maximum contour strength over all directions $gPb = \max_{\theta} gPb(x, y, \theta)$. An overview of the globalization process is shown in Fig. 5.11, showing the principal steps of the algorithm.

5.2.3.2 Creating the hierarchy

The gPb signal gives a probability of boundary map, where highly confident contours have a $gPb \approx 1$. Thresholding gPb lead to non-closed contours, it does not partition the image into regions. Non closed contours may be useful for some applications, but often it is desirable to obtain regions as they can be much more informative to the viewer.

From the gPb an oriented watershed transform (OWT) is produced. The OWT is a classical morphological watershed such as (Beucher and Lantuejoul 1979; Dougherty 1992; Najman and Schmitt 1994), but the resulting contour strength is computed by averaging only the strength of the $gPb(x, y, \theta)$ on the same direction than the edge. In this way, the contours of the starting partition for the UCM match perfectly the non-max suppressed contours obtained from the gPb and the hierarchy can be created.

In contrast to (Arbelaez 2006), the distance between regions is simply defined by the mean strength of the OWT contours. The essence of the algorithm is the same as for

the BPT literature: produce a binary tree of regions, by merging the two most similar adjacent regions at a time. The key difference is that the distance between regions is based on a local contour strength, rather than a region model. Moreover, as regions are merged, contours are reweighed and the whole hierarchy can be represented by a soft boundary map.

5.2.3.3 Estimating T-Junctions

Unfortunately, the machinery underlying the *gPb-OWT-UCM* algorithm is highly coupled, with many parameters trained, and introducing new features/characteristics in one of the steps could lead to poor system performance. Moreover, since the quality of the regions produced are the best for state of the art in segmentation, we choose not to modify the UCM creation process, see results in Sec. 5.3. Instead, once the binary tree is constructed for each image, T-junctions are estimated in a top-down way, following the inverse order of the merging sequence. In this way, the process of T-junction estimation is similar to the one in the BPT creation process, but in this case distances between region are not modified by junction confidence. Note that if the distance d_d in Eq. (5.14) is introduced to the UCMs, the ultrametric property could break. Some examples of the proposed BPTs and UCMs are shown In Fig. 5.12, along with the T-junctions estimated in each case.

5.2.3.4 Differences between BPT and UCM

Minor differences of UCM and classic BPT lead to a high increase of the segmentation performance. One of the problems of some of the BPT distances is that the region model does not represent the spatial correlation of pixels within regions. That is, when considering the region color histogram, the spatial distributions of colors are lost and, for relatively large regions this can be a real drawback, see Fig. 5.13. There are other subtle differences, most of them are commented in Sec. 5.3. Since the proposed BPT and UCM hierarchical representations offer the best results of the state of the art, they are chosen as the basis for image region representation. Both trees are conceptually the same: a binary tree of regions constructed greedily using a specific region distance. Therefore, both structures can be represented similarly. To differentiate between both construction schemes, the trees constructed with the proposed distance in Eq. (5.2) will be referred as BPT, while the tree constructed with the *gPb-OWT-UCM* algorithm will be referred to as UCM.

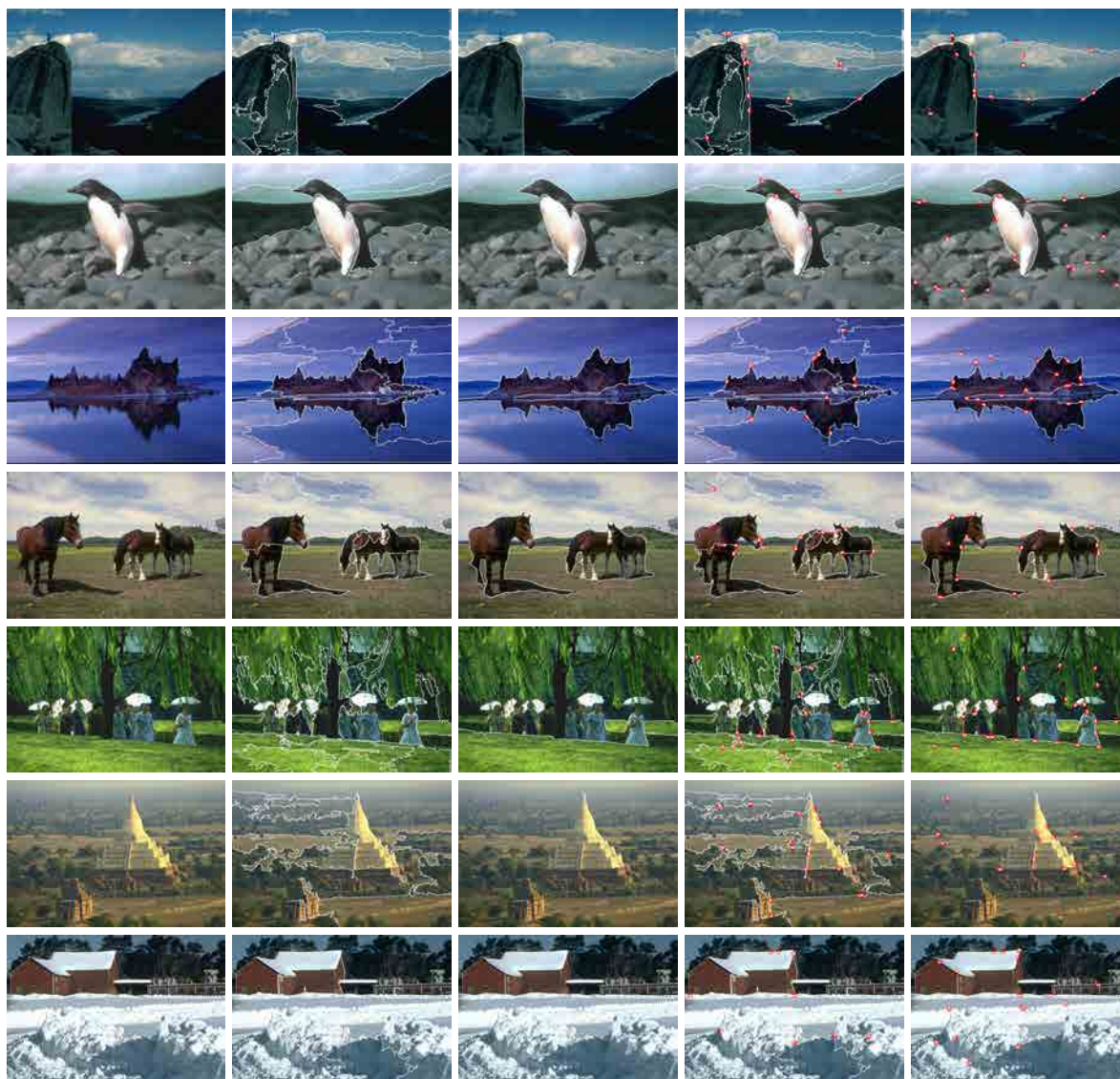


Figure 5.12: Examples of partitions and T-junctions estimated in the BPT and the UCM cases. First column: original image. Second and third columns: image with the contours of the BPT and UCM overlaid in white respectively. Fourth and fifth columns: estimated BPT and UCM junctions exceeding a confidence of 0.1 respectively. Junctions are shown in red, and the region forming the largest angle is filled in white. Partitions were extracted by minimizing a tree cut energy, see Sec. 5.3. Note that many T-junctions do not coincide with the extracted contours because the retrieved partition is too coarse in many cases.

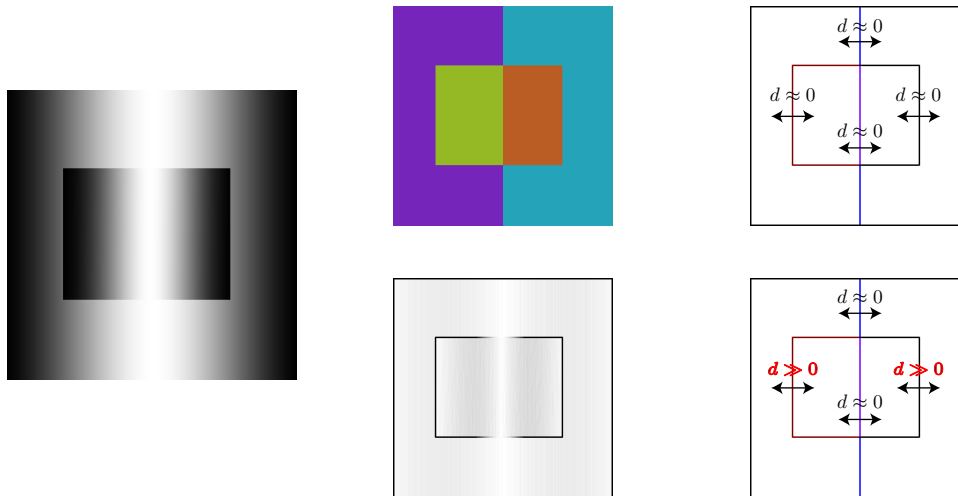


Figure 5.13: Case where the region model of BPT fails. The original left image shows two objects with same color distributions. The center top image shows the image partition at a given instant. The bottom center shows the gradient image. If only region color histogram are considered, the BPT distances will assign the same value to all distances, leading to inconsistent mergings (top right image). Instead, the UCM considers local distances so, objects are clearly differentiated by the gradient distance.

5.3 Tree Cuts

Prior to focus on the depth ordering problem concerning this thesis, it can be of interest to show the potentials of the tree structures for image segmentation. In this section we review the common techniques that may be used to extract different partitions from an image using the same tree. Independently of the distance used to create the tree, the technique extracting a partition from it can be viewed as a *tree cut*. The BPT is a particular graph where each node represents a region and the tree branches the region inclusion relationship. A partition can be naturally defined from a BPT by selecting the regions represented by the tree leaves. If this is done on the original tree, the leaves correspond to the initial partition from which the tree is constructed. However, if we prune the tree, that is if we cut branches at one location to reduce their length, a new tree, called a *pruned BPT* is created. The leaves of the pruned BPT define a non trivial partition. This pruning is a particular graph cut: if the tree root is the *source* of the graph and the leaves are connected to a *sink* node, the pruning cuts the tree in two connected components, one including the source and the other the sink. Note that following this approach, partitions observed during the merging sequence

Set of valid $\mathbf{x} = (x_1, \dots, x_7)^T$:

$$\mathbf{x}_1 = (1, 1, 1, 1, 0, 0, 0)^T$$

$$\mathbf{x}_2 = (0, 0, 1, 1, 1, 0, 0)^T$$

$$\mathbf{x}_3 = (1, 1, 0, 0, 0, 1, 0)^T$$

$$\mathbf{x}_4 = (0, 0, 0, 0, 1, 1, 0)^T$$

$$\mathbf{x}_5 = (0, 0, 0, 0, 0, 0, 1)^T$$

Invalid:

$$\mathbf{x}_I = (1, 1, 0, 0, 1, 0, 0)^T$$

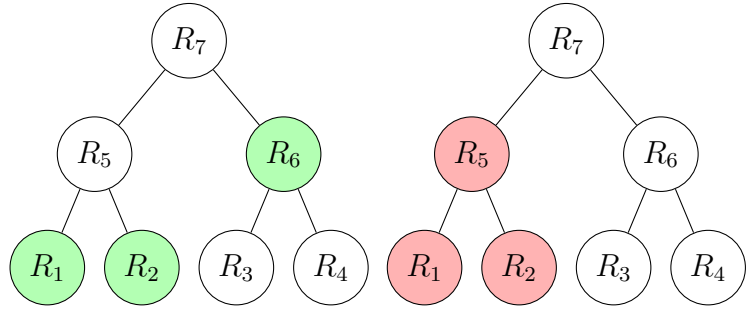


Figure 5.14: Simple BPT and associated valid vectors. Left: Set of valid partition vectors representing a pruning and an invalid partition vector. Center: BPT with green nodes indicating the cut described by \mathbf{x}_3 . Right: BPT with red nodes representing the regions described by \mathbf{x}_I which does not define a pruning.

can obviously be obtained but the interest of the pruning is that a much richer set of partitions can be extracted. Of course, the key point is to define an appropriate tree cut rule. Here, an optimum pruning based on energy minimization is proposed and three different minimization algorithms are assessed. A first formulation and a naive minimization algorithm is given in Sec. 5.3.1. In Sec. 5.3.2 we give a comparison of classical graph cuts and tree cuts, while in Sec. 5.3.4 we exploit the structure of the tree to efficiently find a solution.

5.3.1 A 0-1 Integer Programming Approach

A partition P extracted by pruning can be represented by a *partition vector* \mathbf{x} of binary variables $x_i = \{0, 1\}$ with $i = 1..N$ assigned to each BPT region R_i . If $x_i = 1$, R_i belongs to the partition, otherwise $x_i = 0$. Only a reduced subset of vectors, called *valid* vectors, actually represents a partition extracted by pruning. The problem of finding a vector \mathbf{x} by minimizing an energy can be formulated as:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} e^T \mathbf{x} \quad (5.15)$$

$$s.t \quad \mathbf{A}\mathbf{x} = 1 \quad (5.16)$$

$$\mathbf{x} = \{0, 1\}^N \quad (5.17)$$

where e is a vector containing entries e_i representing the energy associated to a region R_i and x_i . A is a matrix containing all constraints so that x represents a valid partition. A vector x is valid if one and only one region in every BPT branch involves only one $x_i = 1$. A branch is a sequence of regions from a leaf to the root of the tree. For example, the tree of Fig. 5.14 involves four branches.

Each branch l can be represented by a *branch vector* $\mathbf{b}_l = (b_1^l, \dots, b_N^l)^\top$ where $b_i^l = 1$ if region R_i is in the branch and $b_i^l = 0$ otherwise. In the example of Figure 5.14, the four branch vectors are: $\mathbf{b}_1 = (1, 0, 0, 0, 1, 0, 1)^\top$, $\mathbf{b}_2 = (0, 1, 0, 0, 1, 0, 1)^\top$, $\mathbf{b}_3 = (0, 0, 1, 0, 0, 1, 1)^\top$ and $\mathbf{b}_4 = (0, 0, 0, 1, 0, 1, 1)^\top$. With this notation, a partition vector x is valid if, for every branch l , $\mathbf{b}_l^\top x = 1$. In Figure 5.14, $x_I = (1, 1, 0, 0, 1, 0, 0)^\top$ is not valid because $\mathbf{b}_1^\top x_I = 2$. The constraint can be globally expressed as a matrix product $\mathbf{A}x$. In the case of Figure 5.14, the constraint is:

$$\mathbf{A}x = \begin{pmatrix} \mathbf{b}_1^\top \\ \mathbf{b}_4^\top \\ \mathbf{b}_3^\top \\ \mathbf{b}_4^\top \end{pmatrix} x = \mathbf{1} \quad \begin{pmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 1 \end{pmatrix} x = \mathbf{1} \quad (5.18)$$

where $\mathbf{1}$ is a vector containing all ones. The general 0-1 optimization problem are NP-hard to minimize, so standard branch and bounds techniques can be used to arrive at a global minimum. Standard solvers such as CPLEX (ILOG, Inc 2006) are suitable for this task. Although more efficient algorithms can be applied to this particular problem thanks to its structure (see the following section), the 0-1 linear programming approach formulation allows a very easy extension of the model to allow pairwise and higher order interactions between x_i 's. For example, consider the following minimization problem:

$$\mathbf{x}^* = \arg \min_{\mathbf{x}} e^\top \mathbf{x} + \mathbf{x}^\top \mathbf{Q} \mathbf{x} \quad (5.19)$$

$$s.t \quad \mathbf{A} \mathbf{x} = \mathbf{1} \quad (5.20)$$

$$\mathbf{x} = \{0, 1\}^N \quad (5.21)$$

Where \mathbf{Q} is a matrix with the Q_{ij} encoding the cost of x_i and x_j to be on the final partition. The problem formulation is still valid and minimization proceeds as before, with approximate methods. Nevertheless, the problem structure in this case does not allow for an efficient solution and all possible algorithms minimizing (5.19) have exponential complexity on the number of elements in x and the constraints. Still, there

are cases where this formulation is needed such as in (Pont-Tuset and Marqués 2012) and in (C. Xu, Whitt, et al. 2013). Since minimizing 0-1 problem can be inefficient, it is possible to exploit the problem structure to design more efficient solutions.

5.3.2 Equivalent Graph Cut Problem

The most common classes of energies that can be minimized using graph cuts are presented in (Kolmogorov and Zabih 2004). In these problems, an energy should be minimized with $\mathbf{x} = x_1, \dots, x_n$ a vector of binary variables:

$$E(x_1, \dots, x_n) = \sum E^i(x_i) + \sum_{i < j} E^{i,j}(x_i, x_j) \quad (5.22)$$

Not all kinds of energies can be modeled using a graph. Namely, only the so-called *regular* functions are graph representable. Basically, regular functions are functions which allow to construct a graph without negative edges. Since Eq.(5.22) only involves pairwise interactions, a condition for regularity must be satisfied for all pairs of nodes:

$$E^{i,j}(0, 0) + E^{i,j}(1, 1) \leq E^{i,j}(1, 0) + E^{i,j}(0, 1) \quad (5.23)$$

If condition (5.23) is satisfied over all the variables, the energy can be represented in a graph and the minimization of (5.22) can be performed efficiently using a maxflow-minicut algorithm (Cormen et al. 2001). By examining closely (5.22) it is easy to see the similarity with (5.19), but there are subtle differences that make the problem conceptually different. For instance, pairwise interactions in Eq. (5.22) normally refer to adjacency relations in the classical graph cuts applications. In the tree cuts case, since a node is only adjacent to its children, these kind of interactions refer to inclusion relations. Therefore, matrix \mathbf{Q} in Eq. (5.19) and the energy $E^{i,j}(x_i, x_j)$ do not represent the same kind of interactions. However, if pairwise interactions are dropped and (5.15) is used instead, there is a direct correspondence with vector e and the energy terms $E^i(x_i)$ and Eq. (5.15) corresponds to a regular energy.

Although the function (5.15) has the regularity property required to be graph representable, the overall optimization problem is constrained, unlike Eq. (5.22), because \mathbf{x} has to describe a valid a partition. Therefore, additional work has to be done in order to present the same problem of tree cuts but using a formulation which can be solved by graph cuts more efficiently than with 0-1 integer programming.

Beginning by the simplest, yet non-trivial, problem of optimization on a tree T_2 with two leaves shown in Fig. 5.15. The tree involves three nodes, i.e. $\mathbf{x} = [x_1, x_2, x_3]$, with

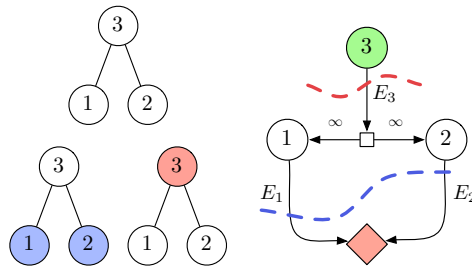


Figure 5.15: Simple case of a binary tree (left) and its corresponding \mathcal{G}_2 graph with the only two possible cuts marked in the graph (red and blue, slashed). All the other cuts in this graph have an infinite cost.

x_1, x_2 representing the leaves and x_3 the root. The only two possible solutions are $\mathbf{x}_0 = \{1, 1, 0\}$ and $\mathbf{x}_1 = \{0, 0, 1\}$ with costs $E(\mathbf{x}_0) = e_1 + e_2$ and $E(\mathbf{x}_1) = e_3$ respectively. Any other \mathbf{x} does not represent a valid partition.

In classical graph cuts problems, the vector \mathbf{x} is unconstrained, while here \mathbf{x} should correspond to a valid partition. Therefore, the problem formulation (5.15) does not correspond to a maxflow-mincut problem at first glance. To convert the proposed problem into a graph cut, we define a graph \mathcal{G}_2 where unfeasible partitions $\tilde{\mathbf{x}}$ are associated to sufficiently high energy values $E(\tilde{\mathbf{x}}) = \infty$. In this way, we allow the graph to be cut at every edge, but only valid partitions have an energy $E(\mathbf{x}) < \infty$. Therefore, the topology of \mathcal{G}_2 should be such that (Kolmogorov and Zabih 2004):

- Each valid state of \mathbf{x} is a cut of \mathcal{G}_2 with cost $E(\mathbf{x})$.
- Each non-valid state of \mathbf{x} is a cut of \mathcal{G}_2 with cost ∞ .

\mathcal{G}_2 is constructed as shown in Fig. 5.15 by introducing a new node and 3 edges. \mathcal{G}_2 has a structure very similar to that of T_2 . In (Serra et al. 2012) a similar comparison between graph cuts and energy minimization is done. In that case, however, the cut was constrained to node capacity instead of edge. To solve the maxflow problem, the root of the tree is considered to be the source node (green node), while a dummy sink node connected to all leaves is introduced at the bottom of the graph (red node). The only two bounded ($< \infty$) cuts are the ones marked in red and blue with the dotted lines. To extend the problem to deal with general binary trees, it is possible to use the inherent recursive structure of a tree as shown in the next section.

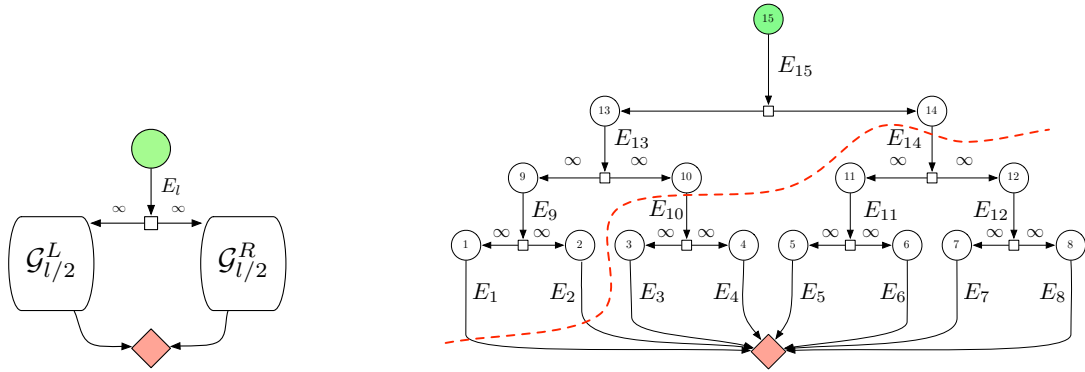


Figure 5.16: Tree cuts can be stated as maxflow problems. Left: Recursive solution to construct the graph \mathcal{G}_l . Right: constructed graph \mathcal{G}_8 for the T_8 case with one marked cut. The energy of the cut is $E_1 + E_2 + E_{10} + E_{14}$ and the partition is formed by nodes 1, 2, 10 and 14.

5.3.3 Tree cuts for general binary trees

The key to deal with arbitrary binary trees is to observe that the energy (5.15) and the constraint (5.16) can be decomposed into two independent problems. Indeed, a binary tree T_l is composed of a root node joining two binary subtrees $T_{l/2}^{L,R}$ corresponding to its Left and Right descendants. Therefore, the energy of T_l (5.15) can be decomposed as:

$$E(\mathbf{x}) = e_l x_l + \sum_{x_i \in T_{l/2}^L} e_i x_i + \sum_{x_j \in T_{l/2}^R} e_j x_j \quad (5.24)$$

$$= e_l x_l + \mathbf{e}_L^\top \mathbf{x}_L + \mathbf{e}_R^\top \mathbf{x}_R \quad (5.25)$$

where the x_l corresponds to the energy of the root. The second and third terms correspond to energies of the left and right subtrees. This energy decomposition is possible because the energy (5.15) is of class \mathcal{F}^1 (Kolmogorov and Zabih 2004) and there are no second order terms relating nodes in different subtrees.

The constraints (5.16) can be decomposed in the following way. According to the structure of the characteristic matrix, the root node corresponds to the last column of \mathbf{A} . Since the last column of \mathbf{A} only involves ones, if $x_l = 1$, then all other x_k should be zero to fulfill (5.16). On the contrary, if $x_l = 0$, then the problem consists in finding the minimum of the second and third terms of equation (5.25). If the last column is ignored, \mathbf{A} can be subdivided by rows into two disjoint matrices \mathbf{A}_L and \mathbf{A}_R , one constraining \mathbf{x}_L and the other \mathbf{x}_R . Finding the minimum for the left and right subtrees

subproblems becomes:

$$\mathbf{x}_i^* = \arg \min_{\mathbf{x}_i} \mathbf{e}_i^\top \mathbf{x}_i \quad (5.26)$$

$$s.t. \quad \mathbf{A}_i \mathbf{x}_i = 1 \quad (5.27)$$

$$\mathbf{x}_i = \{0, 1\}^N \quad (5.28)$$

where the subindex i is L or R when referring to the left and the right subtree respectively. Therefore, the two possible solutions of (5.15) are $\mathbf{x}_1^* = [0, 0, 1]$ or $\mathbf{x}_2^* = [\mathbf{x}_L^*, \mathbf{x}_R^*, 0]$:

$$\mathbf{x}^* = \begin{cases} \mathbf{x}_1^* & \text{if } e_l < \mathbf{e}_L^\top \mathbf{x}_L + \mathbf{e}_R^\top \mathbf{x}_R \\ \mathbf{x}_2^* & \text{if } e_l > \mathbf{e}_L^\top \mathbf{x}_L^* + \mathbf{e}_R^\top \mathbf{x}_R^* \end{cases} \quad (5.29)$$

This recursive definition of the optimum energy is the basis of the dynamic programming solution (P. Salembier and Garrido 2000) (which will be reviewed in the next section) and also the key to construct a more general graph cut as follows.

Since the left and right subtree problems are decoupled, we can construct \mathcal{G}_l by joining two independent graphs $\mathcal{G}_{l/2}$ and a root node. To join these three components, an edge with cost e_l is added to a dummy node which, in turn, is connected to both $\mathcal{G}_{l/2}$ with infinite cost edges. The sinks of each $\mathcal{G}_{l/2}$ are connected to a common sink. This process can be seen in the left part of Fig. 5.16 and a particular case of T_8 is shown in the right part of the figure. Since the two subgraphs are connected by infinite costs edges, the cost of the cuts for \mathcal{G}_l is either e_l or $\mathbf{e}_L^\top \mathbf{x}_L + \mathbf{e}_R^\top \mathbf{x}_R$.

Although conceptually similar, the technique presented here has a fundamental difference with classical graph cuts. As stated in (Kolmogorov and Zabih 2004), graph cuts normally assign labels to nodes depending on the component they are connected to after the cut. In this scheme, we are not interested in the nodes forming the two cut components, but on the edges of \mathcal{G}_l that form the cut of minimum cost. These edges correspond to regions forming the final partition. The maxflow-mincut algorithm can be computed in $O(|V|^3)$, much faster than the 0-1 integer programming approach which is $O(e^{|V|})$. However, in the tree cuts framework, the maxflow-mincut algorithm can be run much faster than maxflow in general graphs using dynamic programming.

5.3.4 Dynamic Programming Algorithm

The dynamic programming algorithm benefits from the fact that the energy e_i in Eq. (5.15) for a region R_i does not depend on regions $R_{j \neq i}$ and that the global energy

Algorithm 1 Optimal Partition Selection: $\text{OptimalSubTree}(\text{Region } R_i)$ contains the set of regions belonging to the subtree rooted at R_i that have been selected to be part of the partition and the sum of their associated energy

```
function OPTIMALSUBTREE(Region  $R_i$ )
   $R_l, R_r \leftarrow (\text{LEFTCHILD}(R_i), \text{RIGHTCHILD}(R_i))$ 
   $(\mathbf{o}_i, e_i) \leftarrow (R_i, e_r(R_i))$ 
   $(\mathbf{o}_l, e_l) \leftarrow \text{OPTIMALSUBTREE}(R_l)$ 
   $(\mathbf{o}_r, e_r) \leftarrow \text{OPTIMALSUBTREE}(R_r)$ 
  if  $e_i < e_r + e_l$  then
    OPTIMALSUBTREE( $R_i$ )  $\leftarrow (\mathbf{o}_i, c_i)$ 
  else
    OPTIMALSUBTREE( $R_i$ )  $\leftarrow (\mathbf{o}_l \cup \mathbf{o}_r, e_l + e_r)$ 
  end if
end function
```

is the sum of the energy values assessed on each region. Therefore, locally optimum decisions lead to global optimum. More precisely, if R_i is a region which has two child regions R_l and R_r , the local decision that has to be taken is to know whether R_i or $R_l \cup R_r$ has to belong to the partition. If e_i is smaller (larger) than $e_l + e_r$, the locally optimum solution selects R_i ($R_l \cup R_r$). The complete tree is analyzed in a bottom-up fashion (from the leaves to the root) to define the complete partition as outlined in Algorithm 1. A formal proof of dynamic programming optimization in trees is given in (Serra et al. 2012) and it is also used in (P. Salembier and Garrido 2000) to find a rate-distortion ratio for a coding system. The algorithm complexity is $O(|V|)$ as each node is only examined once and, therefore, this should be the strategy to optimize energies described by Eq. (5.15).

5.3.5 Are Tree Cuts Useful?

In the previous sections several strategies to extract partitions from binary trees were exposed. For these strategies to be useful, the quality of the retrieved partition should be higher than partitions obtained by trivial methods such as partitions found in the merging sequence or cutting the tree at specified levels. The final objective of tree cuts is to retrieve partitions of better quality for a specific applications and, since depth ordering is rather specific, a general and public segmentation benchmarking is chosen. Although the experiment setup is to evaluate the segmentation quality, tree cuts will also be used to retrieve partitions for depth order. To this end, the following experiment is designed. Image segmentation minimizing the Mumford-Shah (MS) func-

tional (Pock et al. 2009) has been very popular since its first publication in (Mumford and J. Shah 1989). We propose to assess the tree quality by minimizing a MS functional on an image I constrained to the tree, adapting the region energy (5.15) to be:

$$e_i = \sum_{\mathbf{p} \in R_i} |\boldsymbol{\mu}_i - I(\mathbf{p})|^2 + \lambda_p |\Gamma_i| \quad (5.30)$$

where $\boldsymbol{\mu}_i$ is the mean color vector of region R_i and $|\Gamma_i|$ is the length of its perimeter. λ_p is a regularizer parameter that controls the relative importance of the perimeter with respect to the squared error. Taking a closer look at (5.30), one can observe that there exists a compromise between both energy terms. For very oversegmented solutions (small λ_o), the squared error will be small, although the total number of contours points will be high. On the contrary, for subsegmented solutions (high λ_o), the number of contours points will be low at the expenses of a higher squared error.

Three hierarchical segmentation approaches to evaluate the energy (5.30) are selected. The algorithms are the Binary Partition Tree constructed with the Normalized Weighted euclidean distance between models with Contour Complexity (NWMC) (Vilaplana et al. 2008), with the Independent Identically Distributed - Kullback Liebler (IID-KL) (Calderero and Marques 2010) distances and the proposed distance for monocular depth in Sec. 5.2.2, which is referred to as monocular depth (MD). Additionally, the UCM technique is also included in the evaluation. The dataset is the Berkeley Segmentation Dataset (BSDS500) and the reported results are on the test subset (200 images). The tree cut framework is evaluated using two classes of experiments. First, we assess the improvement of the MS functional in an unsupervised way. Second, we introduce the human marked segmentations available on the BSDS500 dataset and evaluate the subjective quality of the contours and regions produced.

5.3.5.1 Unsupervised assessment

Fig. 5.17 shows the results of the minimization as a function of the regularization parameter λ_p . The left plot compares the total energy of the optimum tree cut with the optimum partition found through the merging sequence for a given λ_p . In absolute values, the NWMC technique is the one offering best results. This is understandable as the NWMC hierarchy is constructed using a region distance which is closely related to the Mumford-Shah functional. Beyond this conclusion, it is important to note that the tree cut provides better partition than the merging sequence in all cases.

The right plot shows the relative improvement of the tree cut with respect to the merging sequence. There is a general trend on the behavior of the cuts on the three hier-

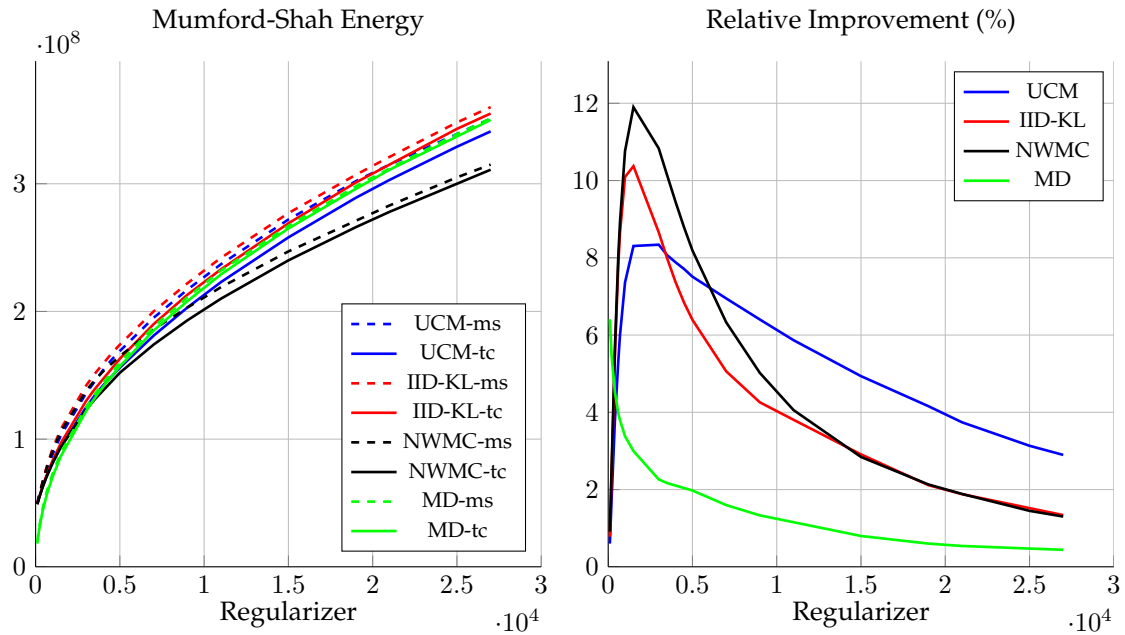


Figure 5.17: Mumford Shah error on the BSDS500 Dataset. Left: Minimum Mumford-Shah energy of the merging sequence (dashed lines) compared to the tree cut (solid lines). In all cases the tree cut helps to reduce the error. Right: Relative energy improvement depending on the regularizer.

archies. For large λ_p , partitions have few regions corresponding to cuts near the tree root. If only the last nodes of the tree are considered, there are few combinatorial possibilities and the relative improvement is low. As λ_p becomes smaller, the relative improvement augments, reaching an optimum point. Then, the relative improvement decreases because we deal with highly oversegmented partitions that are converging towards the initial partition.

5.3.5.2 Supervised assessment

It is common to evaluate algorithm results against a human annotated ground-truth database, so as to verify the *human meaning* of the results. Here, an assessment of the contour and region quality of the tree cuts is reported. The two measures are based on detection frameworks, either by detecting contour locations or regions:

Contour detection performance Contour detection performance is evaluated the bipartite matching from (D. R. Martin et al. 2004). As explained in Sec. 4.2, bipartite matching finds a one-to-one matching between the detected contours and the

groundtruth ones. Detected contour pixels that cannot be matched with a corresponding groundtruth create false positives. On the other way, groundtruth contours that are not matched with a detected contour create false negatives. Matched contours are true positives. After the matching, precision-recall measures can be computed, assessing the contour detection performance.

Region detection performance Similar to contour, image segmentation can also be assessed by considering it as a region detection problem. We apply the measure from (Pont-Tuset and Marqués 2013), where regions are matched with groundtruth annotations using the Jaccard index, as in Eq. (4.12). Unmatched groundtruth regions create false negatives, while oversegmentation (two regions matched to the same groundtruth) create false positives.

The precision-recall (PR) on contours is presented on the left part of Fig. 5.18, while the PR on regions (Pont-Tuset and Marqués 2013) is shown on the right part. Parameters involved in the measures are the same as the one used by authors in their respective papers. NWMC, IID-KL, MD and UCM results are shown together with other state of the art algorithms on image segmentation such as the Normalized Cuts (Cour and Benezit 2005) and the Mean-Shift (Comaniciu and Meer 2002).

For contour evaluation, it is common to show the PR compromise. However, since the operating point is very application dependent, a commonly agreed measure to summarize the system performance is the so called F-measure, which is the maximum harmonic mean of Precision and Recall. The higher the F-measure is, the better is the algorithm considered to behave. We also report these values in the figure. The PR of the merging sequence (dashed lines) is compared with the tree cut (solid lines of same color). It is possible to see a great improvement in both NWMC and IID-KL (0.07 and 0.04 respectively), a moderate increase in MD (0.02) although the UCM loses performance (-0.02).

The improvement in the MD, NWMC and IID-KL cases can be explained mainly by two factors. The first, and more obvious, is that the partitions generated by tree cut are better than the ones of the merging sequence. The second factor is related to how results of each individual image are *aligned* to provide a single dataset curve. If the system performance depends on a parameter value θ , each image I_i , $1 \leq i \leq N_I$ will have its optimum operating point θ_i , where the F measure is maximum for that particular image. A desirable system behavior is to have the same optimum operating point for each image, but this is practically never the case. In real scenarios, the parameter

5. DEPTH ORDERING IN STILL IMAGES

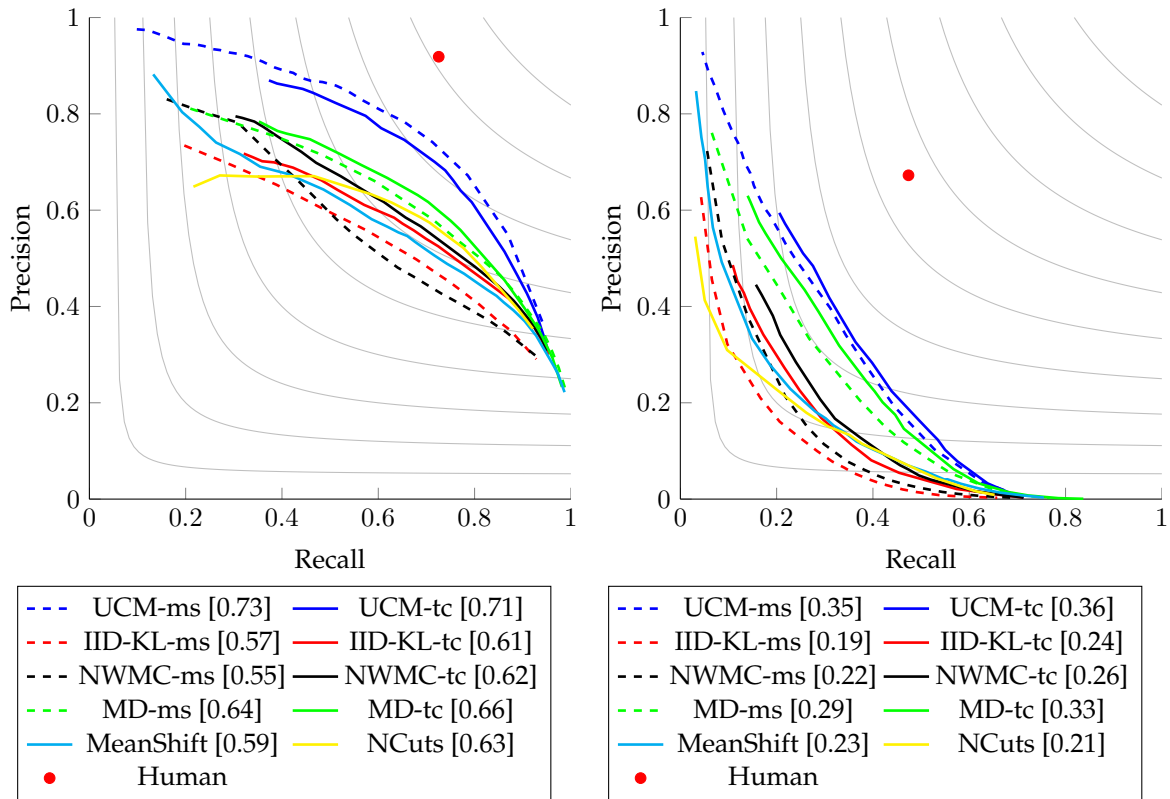


Figure 5.18: Precision-Recall (PR) on contours (left) and on regions (right) on the BSDS500 dataset. Suffixes ‘ms’ and ‘tc’ indicates results evaluating the merging sequence and the tree cuts respectively. The F-measure of each technique is shown in brackets in the legend.

θ is chosen equally for each image so as to give the best F measure in average for the whole dataset. The value of the parameter achieving such point is the Optimal Dataset Scale (ODS) θ_{ODS} . On the contrary, if the value of θ is tuned to give the optimal point in each image, the Optimal Image Scale (OIS) is achieved and θ has a different value for each image. That is, θ_{OIS}^i , for image i . Since in the OIS case the optimum is achieved for each individual image, the overall performance is always higher (or equal) than in the ODS regime, with equality only when $\theta_{OIS}^i = \theta_{ODS}$ for all images. As θ_{ODS} gets closer to θ_{OIS}^i for each image, the average system performance improves.

For the MD, NWMC and IID-KL merging sequence cases, each point of the PR curve corresponds to the average of individual results having the same number of regions. For the tree cut cases, each point corresponds to results having the same λ_p . It turns out that the ground-truth segmentations have a variable number of regions, making the *OIS* differ from the *ODS* on the merging sequence case. However, the *OIS* and *ODS*

are much closer when aligning results with λ_p , which makes the average PR curve to reach a higher F-score. Note that after the tree cut, the F-score obtained for the NWMC is competitive with the technique of (Cour and Benezit 2005). The decrease in performances of the UCM is due to the fact that these techniques are *explicitly* constructed to improve the F-score based on the gPb contours. Although the tree cut improves the MS functional, the contours of minimum MS energy partitions do not correspond very precisely with human ground-truth segmentations. This is caused by the presence of textures and color variability that cannot be captured by the mean-error model (5.30). Of course, this conclusion opens the door to the development of alternative energies.

For region quality evaluation, the PR for regions is shown in the left plot of Fig. 5.18. In this case, the tree cuts improve the F-score in all three cases, with high gains in MD, NWMC and IID-KL (0.04, 0.04 and 0.05) and a smaller improvement in the UCM (0.01). The improvement can be explained with the same reasoning as for contours: partition quality and alignment. There is also a last factor that may explain the improvement, which is related to the measure (Pont-Tuset and Marqués 2013). Contour detection is very sensitive to over/subsegmentations, making precision/recall fall rapidly if 'extra' contours are detected or some are missed. Region evaluation, on the contrary, is more insensitive to solutions having more/less regions (although it is also penalized), as long as the shape of these regions coincide sufficiently with the ground-truth partitions. UCMs are very efficient for contour detection, although their performance on object/region recognition can be improved (Carreira and Sminchisescu 2012). This can be observed by the fact that, even with a tree cut, the boundary performance decreases, although the object quality measure boosts.

The conclusion that can be drawn from the previous experiments is that the UCM is the best state of the art technique for segmentation, followed by the proposed MD trees in Sec. 5.2.2. Since the distance is very different from one approach to another, it is interesting to assess the depth ordering classification system using two different hierarchies. Nevertheless, the resulting segmentations obtained from the UCM and MD trees have to agree with low level depth cues (mainly T-junctions) so, another kind of energy has to be proposed in order to obtain such depth-oriented segmentations.

5.3.6 Depth-based Tree Cut

One immediate application of the tree cuts has been seen in the previous section by minimizing a Mumford Shah-like functional. Since tree cuts are proven to improve the quality of the obtained partitions, we can apply the same idea to generated depth

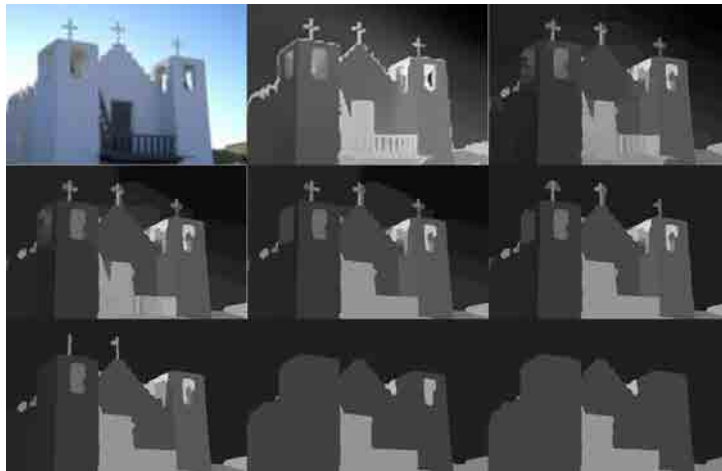


Figure 5.19: Depth estimation example combining (Calderero and Caselles 2013) and tree cuts. In scan order: original image, original depth map obtained with PO and 7 generated depth maps with tree cut techniques with increasing λ

maps from (Calderero and Caselles 2013). The mentioned work estimates a low level cue known as probability of ownership, an intrinsic property of each pixel of the image which can be directly related to the relative depth, see Sec. 3.1 on how this cue is computed.

Although the algorithm (Calderero and Caselles 2013) provides good results, it is based on processing raw color pixel information and sometimes it cannot deal with noise, blur and texture. Although the original algorithm processes data at different resolutions, edge and texture effects are often visible. A mid level representation (such as regions) is often desirable, for example, to determine objects in an image and their relative depth ordering. Here we propose to incorporate the UCM hierarchical representation (Arbeláez et al. 2011) to generate high quality regions with a constant relative depth value. The idea of the algorithm is similar to the one in (M. Maire 2010), where the UCM are used to introduce regions to the soft depth estimated map generated by angular embedding. However, the followed approach differs from the previous one as here the UCM are processed a posteriori with tree cuts to estimate homogeneous depth zones. The outline of the algorithm is as follows:

- Generate the UCM representation using the color image
- Estimate depth using (Calderero and Caselles 2013)
- Use tree cuts to obtain a depth-homogeneous partition

Since the first and second step is carefully explained in Sec. 5.2 and in (Calderero and Caselles 2013), the focus resides in the third step. The problem is very similar to the minimization of the energy Eq. (5.30), where homogeneous color zones were extracted. In this case, instead of working with color, homogeneous depth zones are extracted from the original depth map. This extraction is performed by minimizing a Mumford-Shah like functional (Mumford and J. Shah 1989) over the UCM segmentation hierarchy. To do so, the tree cuts technique is used to minimize a distortion error between the original depth maps and a depth model for each region. The model for each region is chosen to be a simple average of the original depth data, and the energy of Eq. (5.15) can be particularized for this case as:

$$e_i = \sum_{p \in R_i} \left| \hat{D}_i - D(\mathbf{p}) \right|^2 + \lambda |\Gamma|_i \quad (5.31)$$

\hat{D}_i is the mean depth value of region R_i and $D(\mathbf{p})$ is the original estimated depth value for a pixel \mathbf{p} . Γ_i is the region perimeter and λ is a parameter controlling the partition granularity. This equation is very similar to equation (5.30) used to assess the segmentation quality using tree cuts, and it exhibits the same kind of compromise: Small values of λ create fine partitions with many regions, while for larger λ coarser regions are found, see an example in Fig. 5.19. The final estimated depth partition is formed by regions with minimum tree cut energy and each region is filled with its mean depth value.

Although a more thorough evaluation is shown in Sec. 5.5, some examples of the tree cuts with different granularities are shown in Fig. 5.20. Note that the estimated original depth presents some artifacts on object edges. The depth propagation of (Calderero and Caselles 2013) uses only color, so depth in highly textured or non-homogeneous have undesired variations. Incorporating regions and controlling the regularization parameter, both the edge artifacts and the texture depth variations effects can be mitigated. This mixed approach using the probability of ownership combined with segmentation hierarchies proves to give good results on public benchmarks, see Sec. 5.5.

While the probability of ownership is an intrinsic property of each pixel and with a simple tree cut competitive results can be obtained, it is still of interest to see which are the limits of the common low-level cues: T-junctions and convexity. In the next sections a different depth ordering method dealing explicitly with T-junctions and convexity is exposed, showing how local depth relations between regions can be globally integrated to generate a global depth map. The local depth cues are explicitly estimated

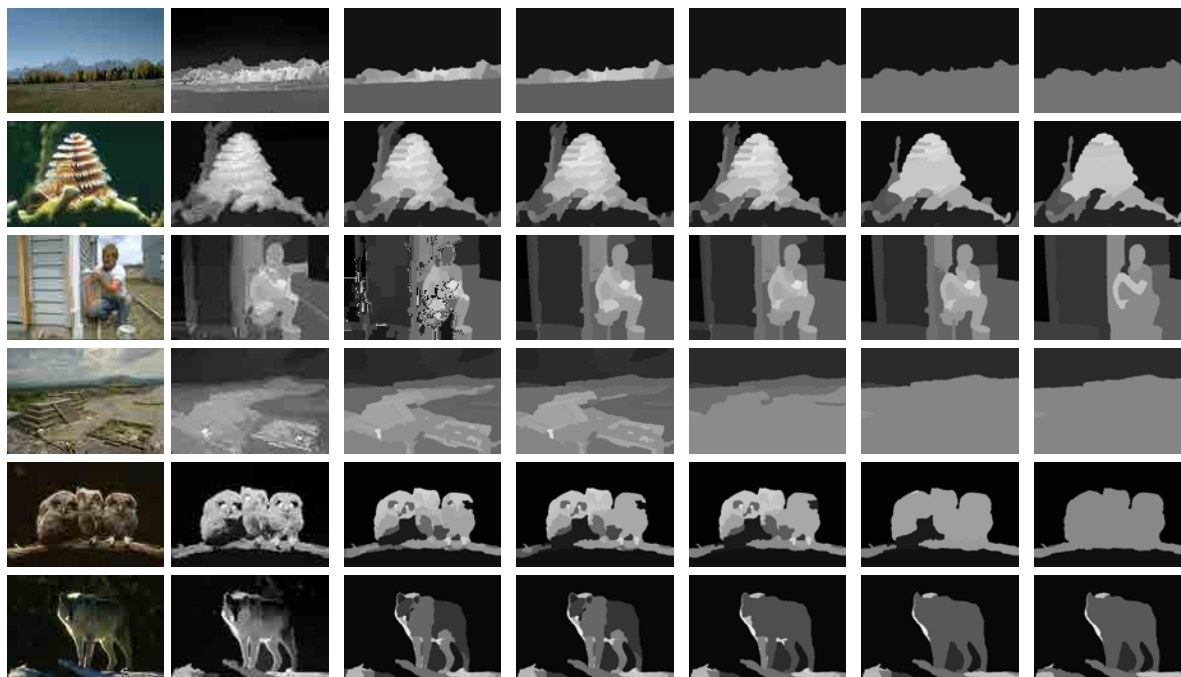


Figure 5.20: Examples of depth maps by (Calderero and Caselles 2013) and the tree cut minimization results in Eq. (5.31). The leftmost column shows the original image and second shows the original depth map from (Calderero and Caselles 2013). Columns three to seven show tree cuts with different λ for each image.

and combined with a set of different hierarchical segmentations to assess the potential limits of low level cues.

5.4 Depth Ordering

5.4.1 Occlusion Based Tree Cut

When trying to recover the depth ordering in an image, a suitable segmentation inside the tree should be found. The MS functional in the previous section allows for an automatic segmentation selection based on a compromise between color distortion and contour length. Although it proves to give good results on segmentation benchmarks, the obtained segmentation may not be compatible with the depth cues estimated during the tree construction. That is, the contours of the regions forming the partition may no correspond with T-junction coordinates. In an ideal situation, detected T-junctions would indicate a change of depth plane, so the three regions involved in the junction should form part of the final partition. Since in typical images there will be few T-

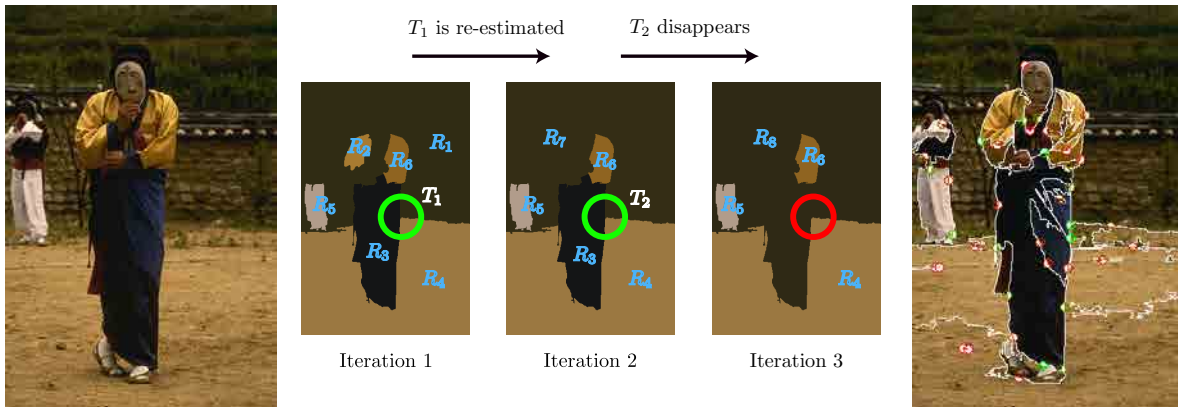


Figure 5.21: Example of T-junction estimation process. When constructing the tree, the same T-junction can be estimated multiple times. In this case, from the original image in the left, a tree is created from the image partition shown in the second image. T_1 involves R_1, R_3, R_4 . Since a merging occurs between R_2 and R_3 , the T-junction is re-estimated giving rise to a new T_2 involving R_1, R_4 and R_7 . In the next merging, since two regions involved in T_2 are merged, the T-junction disappears. In the right image, the set of T-junctions exceeding 0.01 confidence are shown. The degree of confidence is color coded, where green and red mean high and low confident scores respectively.

junctions, there exist zones in the images without any depth cue. Thus, regions not involved in any relevant junction should be discarded, as they will not have any depth relation with their neighbors.

In practical situations T-junctions have a confidence value $0 \leq p \leq 1$ which determines the degree of confidence that a given point is a T-junction point. Potentially every point in the image is a T-junction point, so each three-way region intersection is characterized with a confidence value. If the tree is built correctly, prominent T-junctions appear at the top of the tree, when objects are well represented by regions. So, intersections at the low levels of the tree should be low-confident, while T-junctions near the root of the tree should have a high confidence. So, there exists a compromise between the number of T-junctions considered and the number of regions of the final partition. For instance, a partition preserving low confidence T-junctions coordinates will be a very oversegmented partition. On the contrary, if only very high confidence T-junctions are kept, undersegmentation can occur.

To obtain a partition with a given compromise between T-junctions and number of regions, the process of T-junction estimation should be further analyzed. At a given tree building iteration, a T-junction T_i involves three regions R_1, R_2, R_3 . If the next



Figure 5.22: Examples of partitions obtained using tree cuts with the region cost (5.32). The first two images correspond to the original image and an oversegmentation showing the estimated T-junctions with the same color code as in Fig. 5.21. Partitions are obtained by increasing λ_o , so coarser partitions are obtained each time. Each region is colored with a different gray level.

merging involves $R_i, i = 1, 2, 3$ with $R_j, j \neq i, j \neq 1, 2, 3$ one of the tree region forming the T-junction changes and thus, T_i should be re-estimated creating another junction candidate T_{i+1} involving three different regions, see Fig. 5.21. Also, if a merging occurs between two regions involved in the same T-junction, the T-junction disappears for subsequent partitions produced by the merging sequence.

Since all estimations for a point do indeed refer to the same T-junction, there exists some preprocessing before the final partition can be retrieved. The idea is to group multiple T-junction estimations of the same image point and retain only the last estimation before the T-junction disappears. For a point in the image a set T of N T-junctions in the same location are estimated in increasing order of the merging: $T = \{T_1, \dots, T_N\}$ and each T_i involves regions R_1^i, R_2^i, R_3^i . T_N is thus the last estimation before the T-junction disappears and, since it is estimated when regions are larger, the estimated confidence is the most reliable among the T_i, p_N . Upwards the tree, the T-junction disappears after T_N , since two of the regions forming the T-junction are merged. The formed region contains T_N and no depth relations can be retrieved for successively parent regions.

Since this situation will be present with every T-junction candidate, we can take advantage of tree cuts, Sec. 5.3, to obtain a partition which preserves as many T-junctions as possible, while maintaining a reasonable partition complexity. Following the notation of Sec. 5.3, define the cost of a region R_i to be:

$$e_i = \sum_{T_k \in R_i} p_k + \lambda_o \quad (5.32)$$

where p_k is the confidence of the T-junction candidate T_k . The summation is performed over all T-junctions that have disappeared below region R_i on the tree. λ_o is a constant

term for every region included in a partition. There exists a compromise between the two terms of the cost e_i , since at the top of the tree many T-junctions may have disappeared ($\sum_{T_k \in R_i} p_k \gg 0$). Nevertheless, at low levels of the tree, the partition will be formed with many regions and the constant cost λ_o will dominate, see Fig. 5.22. In this way, the obtained partition contains regions belonging to estimated T-junction cues and, at the same time, the number of regions is minimized.

Once the partition is obtained, the depth relations between region are also retrieved from the estimated depth cues:

T-junctions For every three common adjacent regions the confidence of the corresponding T-junction is used to relate the two further regions to the closest one.

Convexity For every contour separating two regions, a convexity confidence is estimated as discussed in Sec. 3.1.

Since T-junctions and convexity establish a local depth order between region in the partition, they may be contradictory cues. Resolving these conflicts and arriving at a global and consistent depth ordering is done by propagating local depth relations using a Depth Order Graph.

5.4.2 The Depth Order Graph

Once the final partition P_d is obtained through the tree cut, a global ordering can be computed. The problem could be viewed as a rank aggregation problem which are used for web ranking [Dwork et al. 2001](#) or photosequencing [Basha et al. 2012](#). Here, the goal is to achieve a fully ordered list from a set of partial orders by minimizing a given cost function. Normally, rank aggregation works with fully ordered lists, where two elements cannot have the same order. Since, in a image, two different regions may be at the same depth (thus have the same order), we state the problem as a network reliability problem [Terruggia 2010](#).

There are two sets of depth cues that may contribute to determine the relative depth order between regions. For a given partition P_λ obtained with tree cuts using cost (5.32), T-junctions relations and convexity relations (estimated using algorithm in Sec. 3.1) are used to create a Depth Order Graph (DOG), see Fig 5.23. Nodes in the graph represent regions on P_λ - The depth relations are represented by directed weighted edges, going from the foreground region to the background one. Once all the edges are defined, a directed graph is obtained like the one illustrated by Fig. 5.23. A depth cue

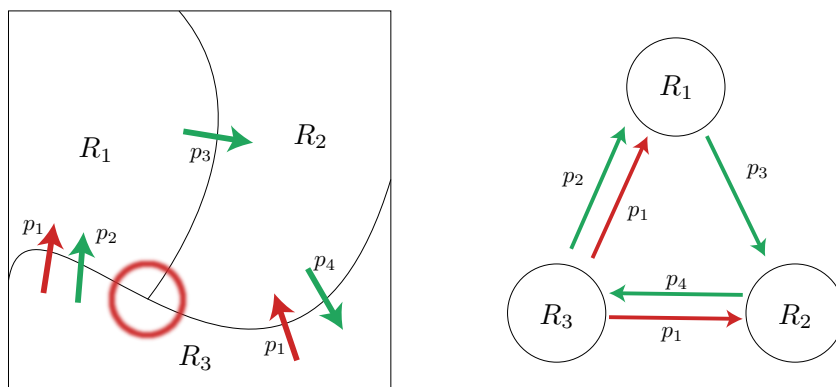


Figure 5.23: Example of DOG construction from a given partition. Left: partition and depth relations overlaid. The T-junction is marked with a red circle. Depth relations obtained from T-junction are marked in red, and convexity relations are marked in green. Right: corresponding directed graph constructed from the depth cues.

characterizes the relation between one (or more, in case of T-junctions) pair of nodes. If the cue confidence relating node R_i and node R_j is p , the directed edge weight is $p_{ij} = p$.

Since the perception of depth involves the interpretation of (sometimes) conflicting cues, the DOG may also present these conflicts. A depth conflict occurs when, due to a set of depth cues, a region can be on the top of itself. If that is the case, two regions exist, R_1 and R_2 , which are at the top of each other at the same time making depth ordering impossible. These conflicts are identified as cycles in the DOG. That is, if one can find a cycle in the graph, all the regions belonging to the cycle are classified as incompatible between them. Fig. 5.23 shows a graph with 3 region with several conflicts.

The main idea of conflict resolution is to modify/eliminate depth cues with low confidences to achieve a direct acyclic graph (DAG) so that it is possible to establish an order between nodes. To do so, a global reasoning of all the cues is performed using the principles of network reliability computation (Terruggia 2010).

The DOG is a graph with edge weights representing probabilities of precedence. That is, if only two nodes R_i and R_j were present in a DOG, and these nodes were connected by a single edge e_{ij} with weight p ; the probability that node R_i precedes R_j (the node R_i is in the foreground) would be p . In practice, more than two nodes and more than

one edge form a DOG. The DOG can be seen equivalently as a network of reliable links (Terruggia 2010) and the reliability between two nodes, in the proposed case, is called probability of precedence (PoP). The overall goal of this step is to perform a global reasoning of the DOG to eliminate cycles for a posterior depth ordering. To this purpose, the following solution is proposed:

1. Compute the PoP for every pair of regions (nodes), R_i and R_j . That is, the probability that R_i is foreground with respect to R_j , ρ_{ij} .
2. Examine all pairs ρ_{ij} and ρ_{ji} . If a cycle is present, both R_i and R_j can be foreground and a conflict exists.
3. In case of conflict, modify one of the paths from R_i to R_j or vice versa to eliminate the cycle.
4. Repeat the previous steps until a DAG is obtained and no conflict exists.

Probability of Precedence Computation To compute the probability that a region R_i precedes R_j all the paths going from the former to the latter region should be considered Galtier et al. 2005. Since edge weights represent the confidence of precedence between pair of directly connected regions (two regions A and B are directly connected if there is an edge from A to B), these weights can be used to calculate the probability of precedence of two non-directly connected regions. Simple rules exist to compute ρ_{ij} when graphs have special topologies.

Single Path If only a single path P_q exists:

$$\rho_{ij} = p(P_q) = \prod_{l=1}^L p_{l,l+1} \quad (5.33)$$

Where L is the number of edges forming the path and $p_{l,l+1}$ is the weight of the edge connecting the nodes l and $l + 1$ on the path P_q . Equation (5.33) shows that the PoP of node R_i to a node R_j with respect to P_q is just the joint probability of all the edges forming P_q . That is, for a node R_i to precede R_j , all the edges in a path P_q should be reliable.

Multiple direct edges If there exist N_E edges between R_i and R_j , ρ_{ij} is the probability that at least one edge is reliable:

$$\rho_{ij} = 1 - \prod_{l=1}^{N_E} (1 - p_{ij}^l) \quad (5.34)$$

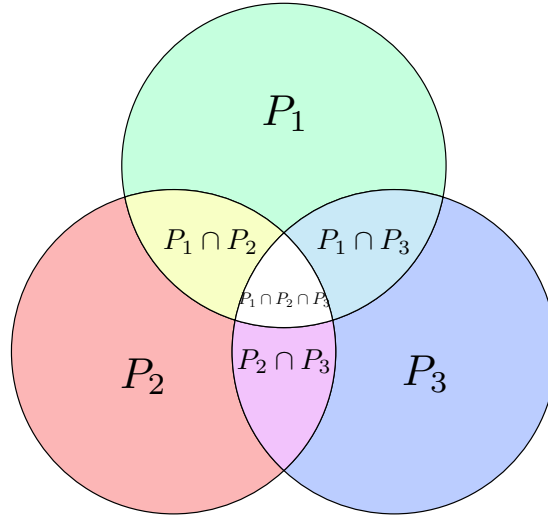


Figure 5.24: Venn diagram illustrating the inclusion-exclusion principle for three paths. Each set represents the reliability of a path. Intersection of two sets represents that both sets are reliable at the same time and, finally, the intersection of the three sets represents that all three paths are reliable.

Where p_{ij}^l is the l -th weight of the edge connecting R_i and R_j

General Topology If a set of N_P paths connect R_i and R_j the PoP ρ_{ij} is the probability that at least one of these N_P paths is reliable. This probability can be calculated by the inclusion-exclusion principle:

$$S_k = \sum_{1 \leq i_1 < \dots < i_k \leq N_P} p(P_{i_1} \cap P_{i_2} \cap \dots \cap P_{i_k}) \quad (5.35)$$

$$\rho_{ij} = p\left(\bigcup_i P_i\right) = \sum_{k=1}^{N_P} (-1)^{(k-1)} S_k \quad (5.36)$$

This can be illustrated with Venn diagrams for a small number of paths, see Fig. 5.24.

To illustrate a simple example of the inclusion-exclusion principle, ρ_{32} is computed for the DOG in Fig. 5.23. Three paths are found going from R_3 to R_2 are found:

- Path P_1 : $R_3 - R_2$. Probability: p_1
- Path P_2 : $R_3 - R_1 - R_2$. Probability (using (5.33)): $p_2 p_3$

- Path $P_3 : R_3 - R_1 - R_2$. Probability (using (5.33)): p_1p_3

The PoP of R_1 to R_3 is defined according to (5.36), for this particular case, as:

$$\begin{aligned} \rho_{13} = p(P_1 \cup P_2 \cup P_3) &= p(P_1) + p(P_2) + p(P_3) \\ &\quad - p(P_1 \cap P_2) - p(P_1 \cap P_3) - p(P_2 \cap P_3) \\ &\quad + p(P_1 \cap P_2 \cap P_3) \end{aligned} \quad (5.37)$$

That is, ρ_{13} is the probability that at least one path is reliable between R_3 and R_2 . The unary terms $p(P_1) = p_1$, $p(P_2) = p_2p_3$ and $p(P_3) = p_1p_3$ are the probabilities that a given path is reliable. The pairwise terms $p(P_1 \cap P_2) = p_1p_2p_3$, $p(P_1 \cap P_3) = p_1^2p_3$ and $p(P_2 \cap P_3) = p_1p_2p_3$ are the probability that two paths are reliable at the same time. Note that in $p(P_2 \cap P_3)$, p_3 is no square, as it represents the same edge on both paths. The last term is the probability that the three paths are reliable $p(P_1 \cap P_2 \cap P_3) = p_1^2p_2p_3$. Therefore:

$$\rho_{32} = p(P_1 \cup P_2 \cup P_3) = p_1 + p_2p_3 + p_1p_3 - p_1p_2p_3 - p_1^2p_3 - p_1p_2p_3 + p_1^2p_2p_3 \quad (5.38)$$

Observing that, even for small graphs, the PoP computation becomes computationally intensive, an approximate strategy should be designed. In an arbitrary large graph, the computation cost involving the inclusion-exclusion principle is exponentially proportional to the number of paths. Instead, the algorithm proposed here is an approximation giving an upper bound for all the pairs of nodes. To approximately compute ρ_{ij} with more than one path between nodes, consider that there are only three nodes R_i , R_j and R_k and that ρ_{ij} and ρ_{jk} are already known. Moreover, assume there is a direct edge from R_i to R_k with strength p_{ik} . An approximate PoP of node R_i to R_k is then given by equation (5.34):

$$\rho_{ik} = 1 - (1 - \rho_{ij}\rho_{jk})(1 - p_{ik}) \quad (5.39)$$

Equation (5.39) is only valid if all the paths connecting R_i , R_j and R_k are independent, although this assumption is not fulfilled in most of the practical cases. The problem of (5.39) resides in computing the values ρ_{ij} and ρ_{jk} which were assumed to be known. It is possible to iteratively compute ρ_{ij} for paths of shorter length and sequentially increase the path length. This process is performed using a modified Floyd-Warshall algorithm (Cormen et al. 2001):

```

for  $j=1 \dots |V|$  do
  for  $i=1 \dots |V|$  do

```

-	R_1	R_2	R_3
R_1	-	ρ_{12}	ρ_{13}
R_2	ρ_{21}	-	ρ_{23}
R_3	ρ_{31}	ρ_{32}	-

$$\begin{aligned}
 \rho_{12} &= p_3 \\
 \rho_{13} &= p_3 p_4 \\
 \rho_{21} &= p_4 p_1 + p_4 p_2 - p_1 p_2 p_4 \\
 \rho_{23} &= (5.38) \\
 \rho_{31} &= p_1 + p_2 - p_1 p_2 \\
 \rho_{32} &= p_4
 \end{aligned}
 \tag{5.40}$$

Table 4: Adjacency matrix representing the transitive closure of the graph in 5.23. The non-zero terms are shown as ρ_{ij} . The graph representation is shown at the bottom

```

for  $k=1 \dots |V|$  do
     $\rho_{ik}^{n+1} = \rho_{ik}^n + \rho_{ij}^n \rho_{jk}^n - \rho_{ik}^n \rho_{ij}^n \rho_{jk}^n$ 
end for
end for
end for
    
```

where $|V|$ is the number of nodes in the DOG. If the DOG contains any cycle, the path length may be infinite so, for practical reasons, the maximum path length is assumed to be the number of nodes on the DOG. The computation of all the pairs ρ_{ij} leads to a new graph which is the transitive closure of the DOG, the DOG^+ . The transitive closure of a graph G is a graph G^+ with the same nodes of G . G^+ contains a direct edge (possibly weighted) from node R_i to R_j if there exists a path P_q in G that connects both nodes. In the case exposed here, the transitive closure of the DOG contain edges with weigths ρ_{ij} . The graph G^+ allows to detect cycles easily as paths with arbitrary lengths are reduced to direct edges. It is known that identifying all cycles in a graph G is an NP problem (Pratt 1976), meaning that there is no efficient solution. Instead, making use of G^+ , cycles can be detected easily by direct comparison of ρ_{ij} and ρ_{ji} .

The building of the DOG^+ is illustrated showing the probability of precedence between nodes in table 4, as an Adjacency Matrix.

Conflict Resolution If a cycle is found, no depth ordering of the nodes is possible. Therefore, some edges should be removed. A conflict may occur mainly because of two factors. The first may be because some false alarms have been introduced in the final T-junction candidate selection and/or in the convexity reasoning. The second may be because self occlusion actually exists in the image. Assuming that self-occlusion is

rather difficult to find in natural images, the conflicts are said to come from bad depth cue selection.

The conflict resolution iteratively seeks the minimum ρ_{ij} of all the pairs of nodes causing a cycle in the DOG. Each time a conflict is found, either ρ_{ij} or ρ_{ji} must be wrongly estimated. Following an intuitive approach, the less confident depth cues should be eliminated. Therefore, the minimum of the two PoP values is considered to be wrong. Therefore, assuming that $\rho_{ij} < \rho_{ji}$, some modifications on the paths that go from R_i to R_j should be done by deleting or turning some edges (and thus possibly breaking the cycle).

Once the conflicting pair is identified, a maxflow-mincut from R_i to R_j is performed on the original DOG. The minimum cut between both nodes gives a set of conflicting edges E . It is not the unique set with this property, but since E has minimum sum over all possible cuts between R_i and R_j , the retrieved edges are likely to be the ones with lower confidences. For each depth cue creating an edge in E , the following reasoning is performed:

Convexity Cue: The cue is considered to be wrong and the corresponding edge is eliminated

T-junction Cue: According to the depth perception principles, exposed in section 2.2, the depth order indicated by a T-junction is not clear. Therefore, if the T-junction depth order has not been modified before, the occluding side is changed, inverting the depth order relationship and turning the edge's inward and outward nodes. If this modification does not solve the conflict, the cue is considered wrong and it is deleted with the corresponding edges.

Each time a modification to the DOG is done, the transitive closure is recomputed, until no cycles are found and a DAG is obtained. A possible iterative solution of conflict resolution for the graph in Fig. 5.23 is illustrated in Fig. 5.25. When all the conflicts are removed from the DOG, no cycles are present. Moreover, a DOG should have a unique order for its nodes/regions. To order the nodes on the DOG, a topological partial ordering is proposed. This ordering is a linear ordering of a graph's nodes in which each node comes before all nodes to which it has outbound edges. That is, if node R_1 has an outbound edge to node R_2 , R_1 will precede R_2 in the sorted list of nodes and R_1 will be closer to the viewer than R_2 . Since in a depth image, two different regions may not have the same depth order relationship between them, it is assumed that in

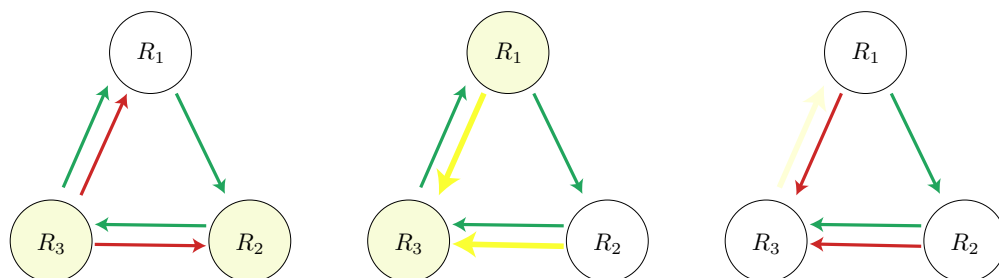


Figure 5.25: Conflict resolution for the graph in Fig. 5.23. The three iterations of the process are shown from left to right. At each iteration, the two conflicting regions are marked in yellow and the modified edges are marked in yellow in the next iteration. In the first iteration, the two edges belonging to a T-junction are reversed. Since there is still a conflict, the edge going from R_3 to R_1 is reversed. The third graph on the right shows the DAG obtained for depth ordering.

such cases both regions have the same ordinal depth. Results using different types of hierarchies are explored in the next Sec. 5.5. Quantitative and qualitative evaluation is also assessed for proposed methods and state of the art algorithms.

5.5 Results

5.5.1 Depth Annotated Dataset

There exist few public datasets incorporating relative depth ordering between objects present in images. There are more datasets devoted to video monocular depth and motion estimation, where the problem of structure recovery from a set of images is more tractable than for single images. Nevertheless, one of the most popular datasets in image segmentation, the BSDS500 (Arbeláez et al. 2011) incorporates figure/ground annotations for a subset of the images. These annotations are performed in contours, where both sides are marked either figure (closer to the viewer) or ground (further to the viewer). Although it is normally the evaluation choice for figure/ground systems, annotations may not have closed contours and no global consistency is found in several cases. Moreover, it is unclear if the term “figure” actually refers to depth or saliency properties. In some annotations, the part that was more salient to the viewer was marked as “figure” even if it was behind other objects, just because it was seman-

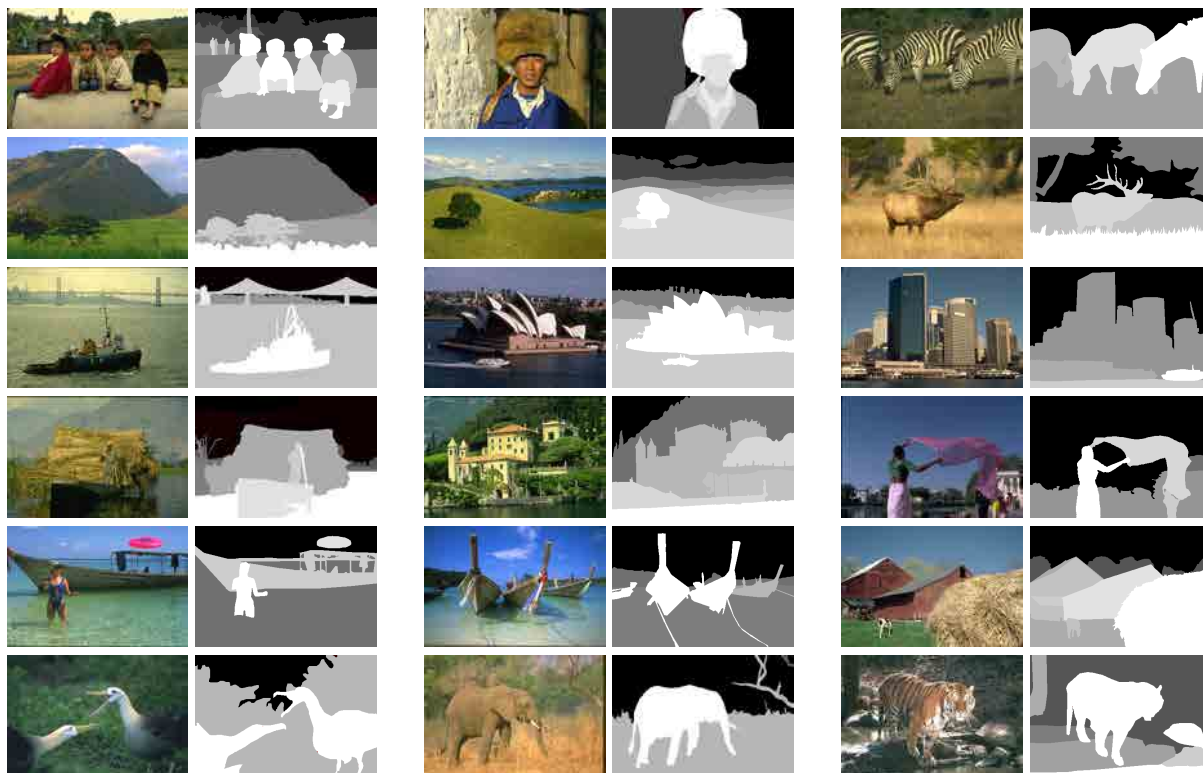


Figure 5.26: Several ground truth annotations for the BSDS500 Datasets. For each column, the original image is shown on the left and the relative depth on the right. Brighter regions are closer to the viewer.

tically more important.

To overcome all the possible limitations of the figure/ground annotations, a new benchmark was created, based on region relative depths. For each image of the BSDS500 dataset one of the human created segmentations was chosen and their regions were assigned a relative depth order. Regions are assigned a constant depth value, with no smooth depth gradient. This may be a limitation when a region spans through different depths (such as an horizontal ground), but it gives a clearer insight on the objects present in the image and their relative position on the scene. Examples can be seen in Fig. 5.26.

5.5.2 Quantitative evaluation

To evaluate the proposed depth ordering system, the LDC and the GDC measures proposed in Sec. 4.2 are evaluated. The following proposed approaches are subject to

this evaluation:

- The proposed monocular depth (MD) estimation ordering scheme based on explicit detection of T-junctions and convexity, using three different hierarchies:
 - The proposed BPT construction scheme, with the pixels as initial partition (method name: BPT+MD)
 - The proposed BPT, with a groundtruth segmentation as initial partition (method name: GT+MD)
 - The UCM technique from ([Arbeláez et al. 2011](#)) (method name: UCM+MD)
- The proposed technique integrating the UCM hierarchy and the PO depth generated depth maps (method name: UCM+PO)

Additionally, techniques from the state of the art are also included in the evaluation:

- The technique from ([M. Maire 2010](#)) using angular embedding for figure/ground segregation (method name: AE)
- The learning based approach from ([Saxena et al. 2005](#)) (method name: LD)
- The raw-probability of ownership approach from ([Calderero and Caselles 2013](#)) (method name: PO)

Results on both LDC and GDC measures are presented in Fig. 5.27. Before commenting the results, some nomenclature follows. Each method provides either a surface or a line in the precision-recall (PR) plane. The higher points in each plot will be referred to as “detection score” as the LCD and GDC measure in these points do not take into account depth classification but only correct detections. The lower points for each graph will be known as “classification score” for a similar reason, as both measures only count correct detection and correct classifications, see Sec. 4 for a review of the PRC framework.

The first thing to notice is that LDC gives higher precision-recall scores than GDC, basically for two causes. First, as already stated in ([Pont-Tuset and Marqués 2013](#)) detecting high quality region is a much more difficult problem than detecting contours. A random contour detection system may have a non-0 detection score on contours, but it will surely fail on detecting regions. For a more thorough explanation refer to ([Pont-Tuset and Marqués 2013](#)). Second, classification scores are higher because a

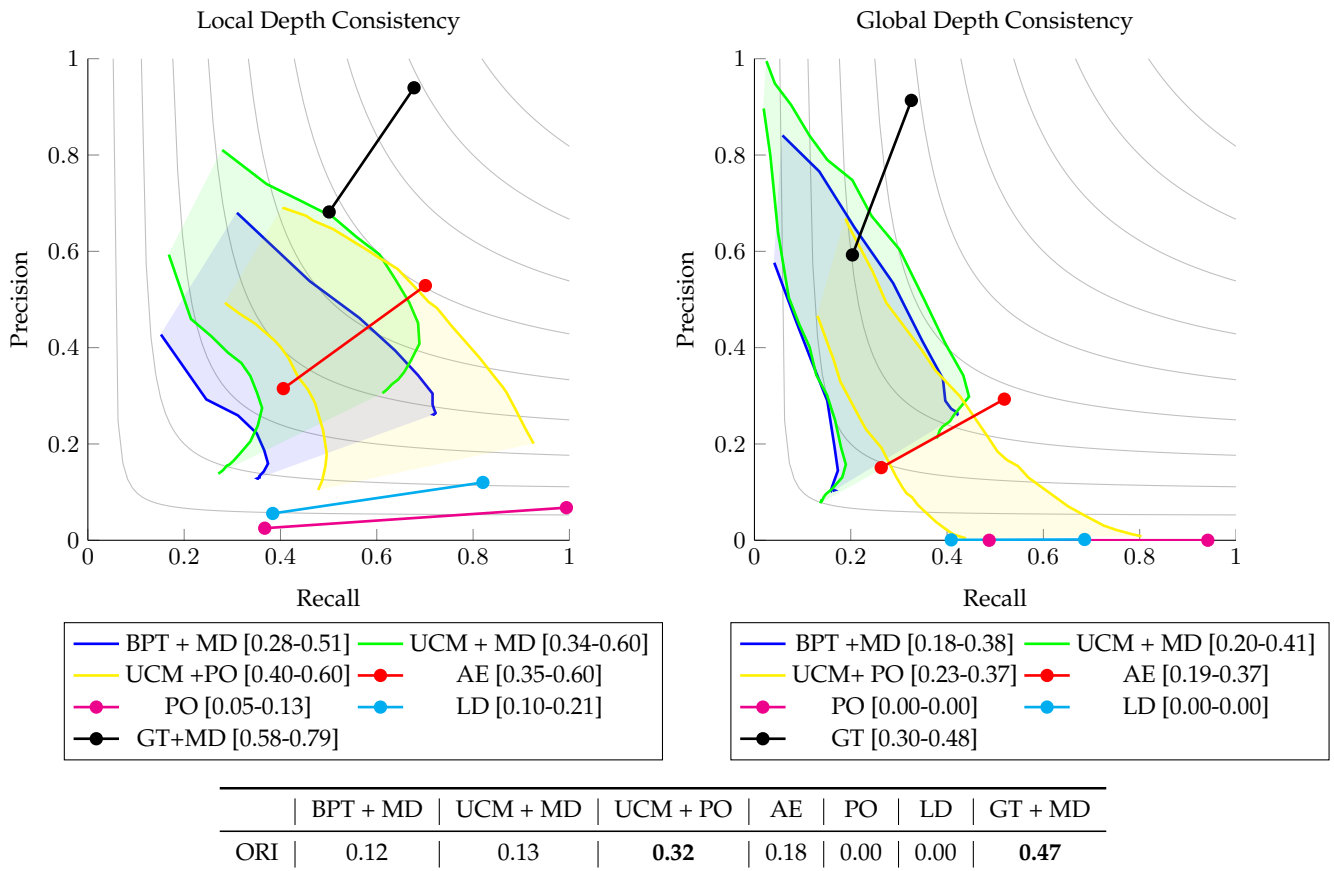


Figure 5.27: LDC and GDC measures for state of the art system. F_{max} and F_{min} measures for point each method are shown in brackets in both cases. The ORI is shown also shown a the bottom table.

global depth consistency is much harder to achieve than local depth gradient at contour points. The LDC measure is a good measure when a moderate oversegmentation is found. For solutions with many detected contours, the classification score suffers from the fact that the bipartite assignment of contours matches random pairs, as they are many potential candidates for a groundtruth contour, giving a poor score. This fact can be seen in the ORI measure for systems PO and LD, where a score of 0 is assigned. This behavior, however, has relative importance because the best F-measures of systems appear on moderate levels of detected recalls where the bipartite matching performs correctly.

LDC performance Analyzing in more detail the LDC error figures it is possible to see that, obviously, the system using the groundtruth segmentation gives the best detection and classification scores. Therefore, the GT+MD presents the best F-measure

for detection, classification and the best ori index (0.47). Since in practical cases the groundtruth segmentations will not be available, the GT+MD can be used to assess the limit of T-junction and convexity cues capacity to give insight about the relative depth order. That is, a system using occlusion cues will not be able to recover always a correct depth, even if a perfect segmentation is available. For systems with no groundtruth segmentation, the best algorithms on detection give similar scores close to $F = 0.6$. That is, for AE, UCM+PO and UCM+MD the detection is similar because the machinery underlying the contour detection step is based on the same algorithm, the gPb , see Sec. 5.2. For systems working at the pixel level, (or very oversegmented partitions), PO and LD, a high recall is observed but with a very low precision (< 0.1). It can be seen that the UCM+PO slowly converges to the PO points for points with high recall. As the granularity of the partitions of the UCM+PO increase, regions become smaller, eventually matching the PO partitions.

For classification scores, the UCM+PO systems is the best system, as it gives higher F-measures ($F = 0.4$) for these points. The system closer to UCM+PO is AE, giving $F = 0.35$ followed by the UCM+MD approach $F = 0.34$. Clearly, the system using BPT is lower in both detection and classification, as the quality of the contours of the BPT are lower than the ones in UCM, see Sec. 5.2. For the UCM/BPT+MD and UCM+PO plots, there is a general trend that for points with low recall both detection and classification scores converge. Usually, operating points with low recall take into consideration only high confidence boundaries and, as stated in (M. R. Maire 2009), the depth organization for higher confidence boundaries is easier than for ambiguous, low confident, contours.

Overall, the best system for the LDC seems to be the UCM+PO approach, giving the best F-measures and, consequently, the best $ORI = 0.32$ way above its next competitor, AE with $ORI = 0.18$. Nevertheless, all figures are very far from their ideal counterparts. For instance, the $ORI_{max} = 1$ and the best system to the date has $ORI = 0.32$, showing much room for improvement in depth ordering. This somehow contradicts figure/ground accuracy performance of the state of the art, where scores of nearly 80% are achieved nowadays, see (D. Hoiem et al. 2011). Here it is shown that there is still a huge gap between computers and humans.

GDC performance Measuring a global depth ordering is much harder than measure a local depth gradient, as the PRC comparison between LDC and GDC shows. For the GT-MD method, for example, the maximum recall of 0.5 shows that, in average, half of the depth transitions are missed. That is, due to the tree cut process, normally small region are discarded, missing true depth transitions. Logically, the GT+MD method

is the one having best figures. For other systems, detection scores are much lower than in the LDC case with the method UCM+MD giving the best F-measure with $F = 0.38$, although UCM+PO, AE and BPT+MD have similar numbers. Although PO and LD provide extremely oversegmented solutions, they are still not able to capture all the transitions between groundtruth regions, since the recall of both system is not 1. For classification it is possible to see that the UCM+PO gives the best score with $F = 0.23$, making it the system with less inconsistent detections and having a better overall depth interpretation of images. This can also be seen by comparing the “width” of the produced region for each plot. If the plot is thin, the system has few inconsistent detections. On the contrary, if the plot is wide, inconsistent detections play an important role.

If one system should be chosen, the UCM+PO should be used. It is the system giving best results due to great consistency on depth ordering. Its ORI index is the best for completely unsupervised systems and it has good overall performance in either contour and region benchmarks. For a more complete comparison and a part from the LDC and GDC measures, it is important to see qualitatively if systems give visually pleasant results and work in different situations.

5.5.3 Qualitative evaluation

Regardless of the numeric figures that a system gives, it is also important to evaluate if obtained results are meaningful to the human eye. As a first assessment, a comparison of the different operating points of the systems BPT/UCM+MD and UCM+PO is shown in Fig. 5.28. In this figure, three different solutions are shown for six images. Solutions were generated by varying the regularization parameter λ_o in the tree cuts energy (5.32). The first solution for each system is the one having more regions and it clearly show lots of false contours and spurious depth transitions. As the regularizer increases, less regions are obtained, eventually reaching an optimum operating point for these images. For the highest regularizer, the systems may be seen as a kind of foreground-background segregation system, as the system tries to separate the front most plane with the rest of the image.

However, normally a natural image can be divided in a few planes, so the optimum regularizer may be the one giving a reasonable segmentation (possibly with little over/sub segmentation) with few region. Images with 5-10 planes usually give a good depth impression, and in Fig 5.28 they are shown in the second row for each method.

5. DEPTH ORDERING IN STILL IMAGES

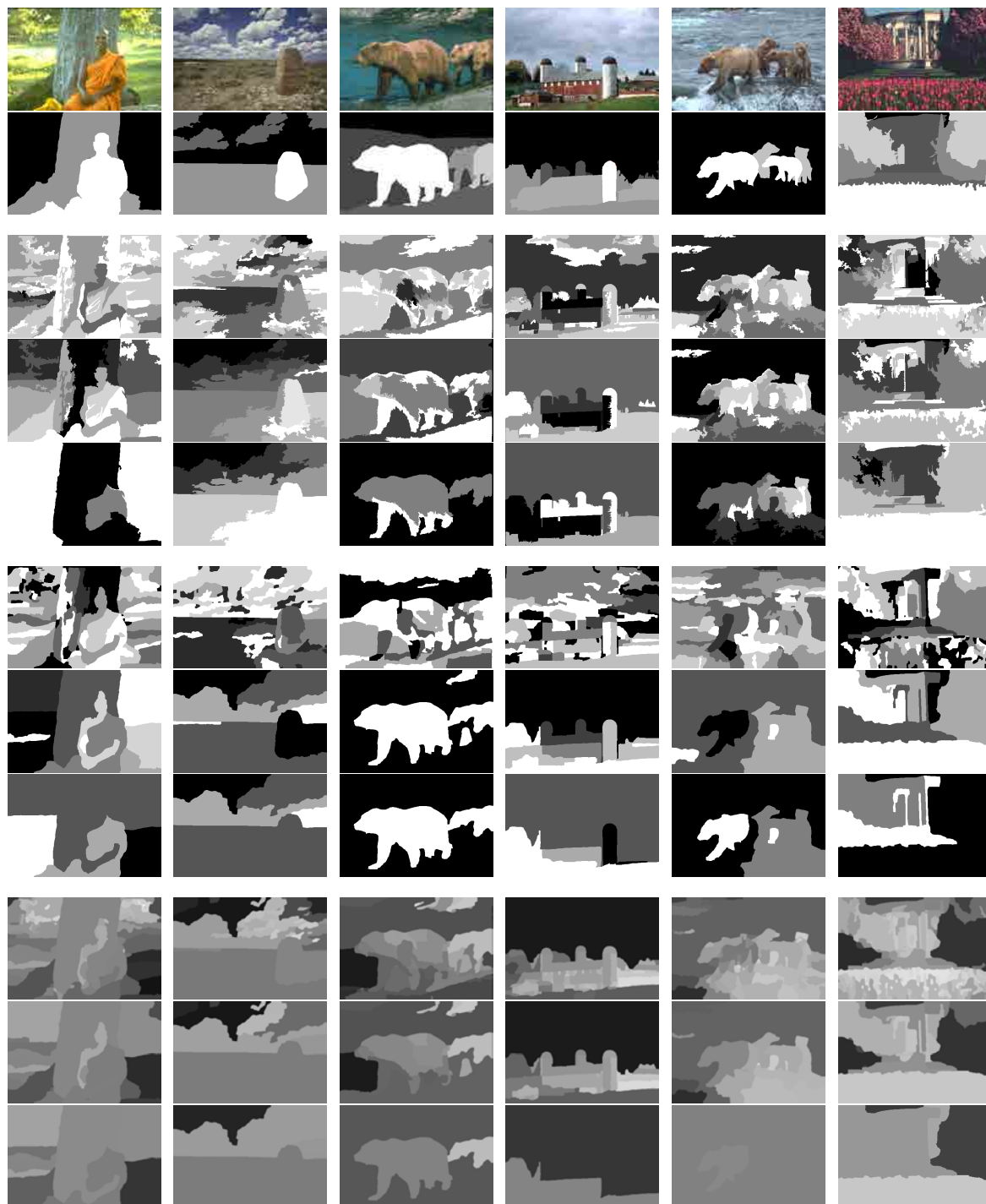


Figure 5.28: Depth ordering examples for a variety of methods. For each row: 1) Original image 2) groundtruth depth annotation 3-5) BPT+MD results for different partition granularities 6-8) UCM+MD results for different partition granularities 9-12) UCM+PO results for different partition granularities.

Although the depth ordering in these cases is not perfect, the overall depth structure of the image can be easily interpreted.

The proposed systems BPT/UCM+MD can also be compared visually with systems using high level information such as LD (Saxena et al. 2005) or the system (D. Hoiem et al. 2011), devoted to detect occlusion boundaries (OB). Additionally, they are compared with systems AE and PO which use only local depth cues. Both high level systems use high level features, as well as low level color and texture cues. They rely heavily on surface layouts and the position of the pixels in the image, inferring a similar depth structure for each image. An important difference with systems only using occlusion cues is that these system can estimate a given orientation for every region, thus producing smooth depth transitions. This is the case for (D. Hoiem et al. 2011), where horizontal surfaces are considered to fade from close to far away depths. Fig. 5.29 shows a comparison over several images. As stated before, relying on high level features, the OB and LD methods attempt to fit the learned model to each image, giving a very similar depth impression for each image. Nevertheless, when the model and the input image correlate, the estimated depth maps are of high quality, specially in (D. Hoiem et al. 2011). Since the LD method was trained mainly on landscape image, the depth layout for each image is very similar, with ground regions being closer to the viewer. Nevertheless, when the high level information is wrong, generated depth maps miss much of the image structure, with unacceptable result. In the proposed approaches, even if some depth cue estimation is wrong the system is able to compensate it and the overall depth structure for image is somehow captured.

Approaches AE and PO offer also good performance over a variety of situations. Nevertheless, the PO algorithm has spurious responses due to working at the pixel level (effect that is compensated by UCM+PO) and it has fuzzy boundaries in some cases. For instance, it creates a imaginary edges near object boundaries. The AE results correlate pretty much with the UCM+MD results as they use the same kind of features and thus the performance is similar (as shown in the LDC and GDC measures).

The fact that the proposed algorithms do not make assumptions on the type of image can be seen in results in 5.30 in comparison with OB and LD. The systems are presented a bunch of different situations (landscape, close photo, high depth range, strong foreground/background separation) for depth estimation. Obviously, trusting only pixel information, without any previous knowledge of the scene can be limiting, but it also has its positive points. For instance, the input of the system can be arbitrary images, assuming always that some kind of considered occlusion cue is present. This

5. DEPTH ORDERING IN STILL IMAGES

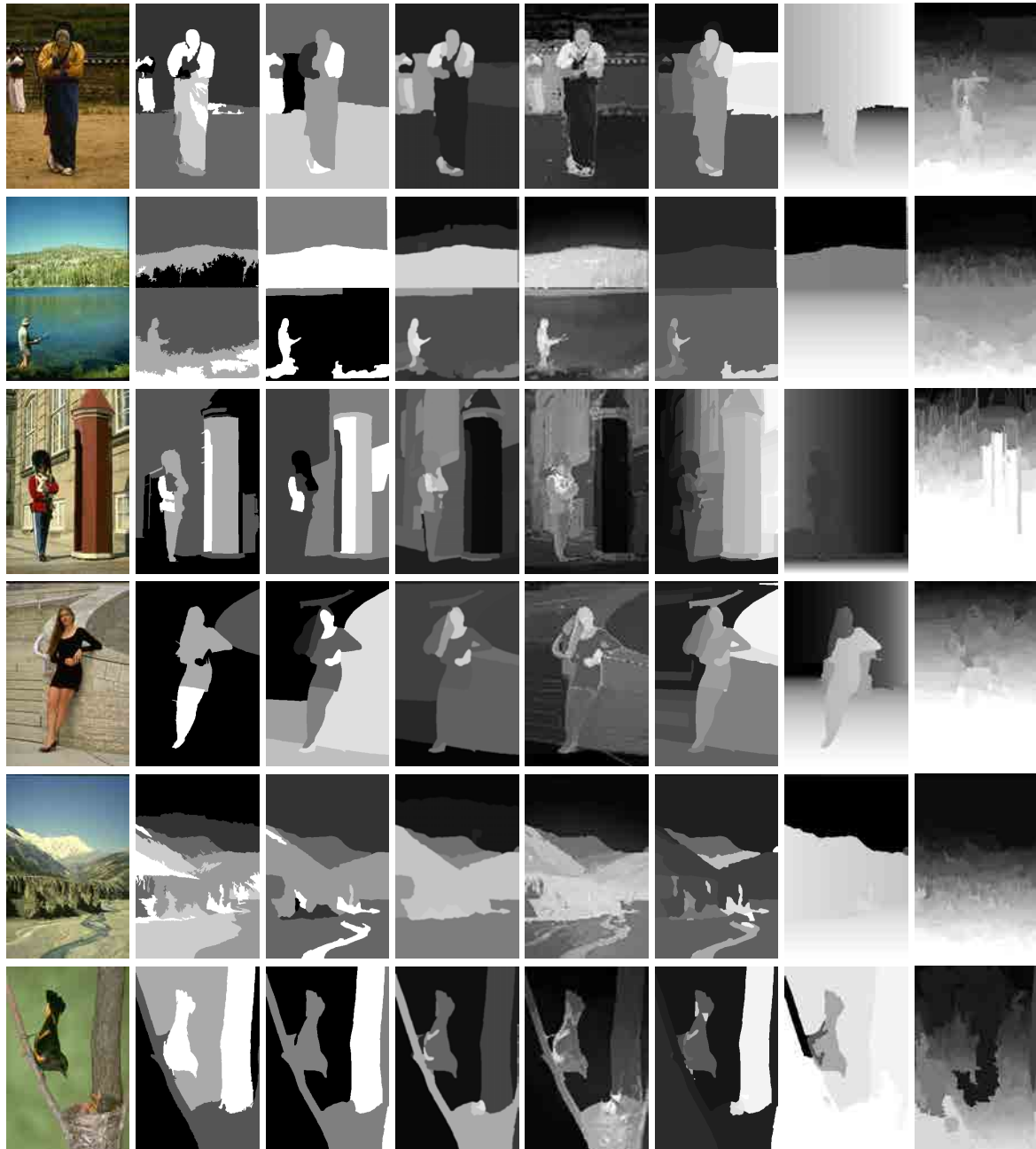


Figure 5.29: From left to right. Original image, depth estimation from BPT+MD, UCM+MD, UCM+PO, PO, AE, OB and LD methods.

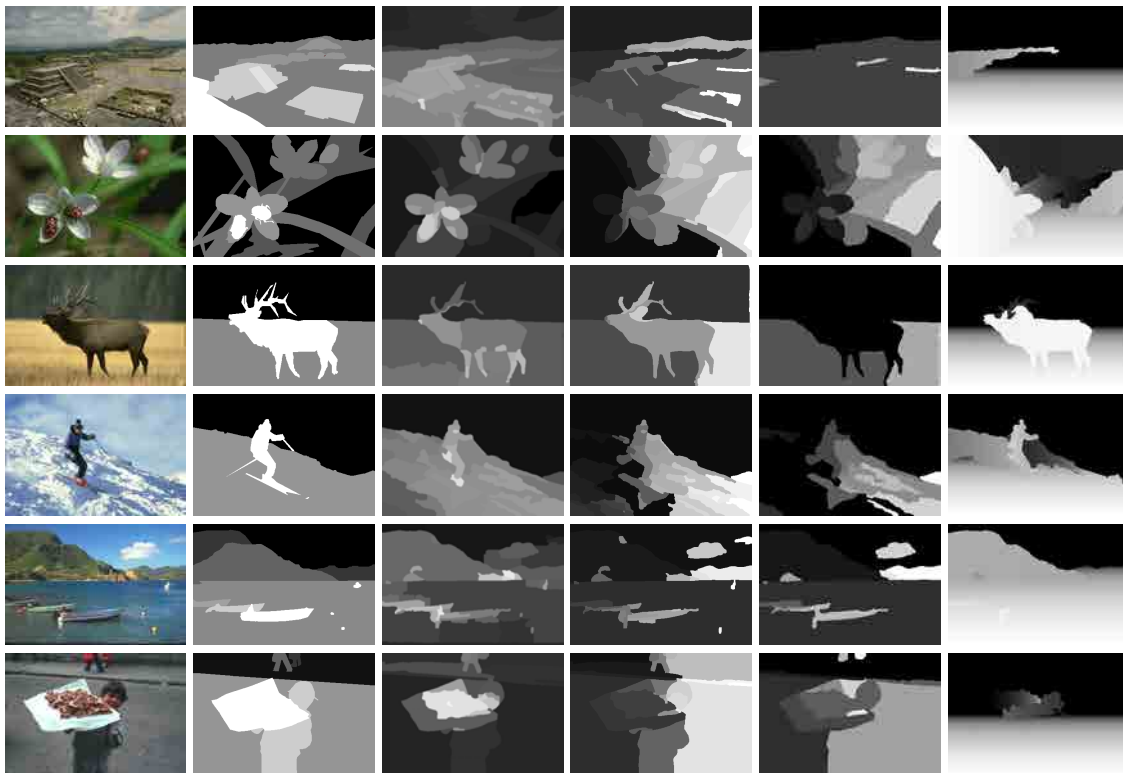


Figure 5.30: From left to right. Original image, groundtruth relative depth order and results from the UCM+PO, the AE, the UCM+MD and the OB methods. Note how occlusion or low level based systems are able to deal with different situations, where high level approaches fail.

makes the algorithm work in more situations such as a landscape, an office or a portrait. Since occlusion cues are known to be almost omnipresent (McDermott 2004), it is guaranteed that the proposed system can accept almost any kind of input images.

The systems also presents some weaknesses. First, there may exist some low level cues which do not conform with the assumed model. This case is specially seen in T-junctions in textured regions, and where convexity does not offer a good depth cue (take, for example, holes). Second, the constant depth model for each region can be limiting for some applications, as all the surfaces of the scene are considered to be parallel to the camera view plane. If a complete depth map has to be retrieved, this model is insufficient since it does not permit to have oriented surfaces which can indeed exist in normal situations. However, for many applications, depth ordering can be sufficient. For instance, if depth ordering is available, with some little user interactions, an

5. DEPTH ORDERING IN STILL IMAGES



Figure 5.31: Depth ordering using user information. From left to right, for both columns. Original image, image with user defined markers and retrieved depth ordered partition for the BPT+MD.

approximate depth map may be available.

Extending the system to accept user input User interaction can naturally be integrated in the working flow and can be used to improve the quality of the depth order map. With very little modification, the described unsupervised systems can be adapted to accept user input. If the user introduces some depth markers in the images, the given information can be used to force some depth relations. There could be many situations where this extension is desirable. For example, user information may overcome some system limitations. Moreover, user may be interested in accurately ordering some parts of the image, leaving all the other regions to be ordered automatically. Since the proposed system is originally designed to perform in an unsupervised way, unlike (Phan et al. 2011), it is able to infer extra depth planes other than the ones introduced by the user. To illustrate the idea, the approach BPT+MD is extended to accept user depth markers. Any of the proposed approaches could be extended (UCM+MD, GT+MD), but the system behavior would be the same with different degrees of user interaction. Markers can be simply defined by roughly marking areas of the image with gray levels. To integrate this information with the depth ordering stage, two little changes are proposed: one concerns the initial BPT tree cut and the other the DOG construction.

Initial BPT pruning A part from preserving the most important T-junction at the ini-

tial partition, the pruning must also preserve user input markers at different regions.

Depth ordering Each pair markers from two different regions introduce a fully confident depth relation. That is, these edges are assigned the maximum confidence $p = 1$, making sure that no edge is deleted in the conflict resolution step and the final depth ordered partition contains all the user markers.

Examples of the system accepting user interaction are shown in Figure 5.31 showing that, with little user information, accurate orderings can be obtained.

6 Depth Ordering in Single Frames of Video Sequences

6.1 State of the Art

Depth perception in human vision for vision sequences relies on several depth cues. When only one point of view is available, disparity cannot be used to infer depth but only monocular depth cues can be identified for structure estimation. There is a huge difference when facing static images or video sequences, since when the temporal dimension is present a set of totally different cues arise. In static images, only T-junctions or convexity cues can be used for occlusion estimation. Nevertheless, as shown in Sec. 5.4, the performance of these kind of cues is limited, and humans should make use of other, higher level, cues in order to infer depth. In video sequences, motion information can also be used to get depth information. For example, occlusion of moving objects, size change or motion parallax are used to structure the scene (Ono et al. 1986). The difference between still and dynamic cues is that motion is much more reliable than static cues. State of the art results on depth estimation or figure ground labeling for motion sequences achieve a much better accuracy with rather simple approaches, while results in static images are far from ideal even with complex approaches.

Nowadays, motivated by the film industry, many research works are focusing on depth maps generation for video sequences. Most approaches make use of several viewpoints to compute the disparity, but shooting or recording scenes with synchronized video cameras adds an extra cost which sometimes, is not affordable. Additionally, camera synchronization can be sometimes impossible to achieve, thus introducing some drift in the disparity estimation. Moreover, one critical issue is the large

Contributions on Depth Ordering on Frames

- G. Palou and P. Salembier. "2.1 Depth Estimation of Frames in Image Sequences Using Motion Occlusions." In: *ECCV Workshops*. Firenze, Italy, 2012
- G. Palou and P. Salembier. "Depth ordering on image sequences using motion occlusions". In: *IEEE ICIP*. Orlando, FL, USA, 2012
- G. Palou and P. Salembier. "Depth order estimation for video frames using motion occlusions". In: *IET Computer Vision* 2013

amount of material that has already been acquired in the past as monocular sequences and needs to be converted to some extent to a 3D format. In such cases, depth information can only be inferred through monocular cues. The film industry is seriously tackling this problem. For example, Disney or Microsoft have designed supervised systems supporting the creation of depth maps for monocular sequences (Ward et al. 2011; Wang et al. 2011). These systems rely heavily on human interaction. However, there is a clear interest in defining unsupervised systems because of their reduced cost in time and money resources.

Since in video there is much more information, estimating the depth from sequences seems to be theoretically easier than estimating depth in single images (although in practice, computational resources are a bottleneck). As a consequence, there is a lot of literature referring to estimate depth structure from whole sequences, but very little on estimating depth for single images. In this section an intermediate approach is presented: estimating depth on single images within video sequences. By reducing the problem to a single frame, the algorithm does not suffer from the computational complexity of processing a whole sequence but it can still benefit from the fact that motion can be estimated with the surrounding frames. State of the art depth ordering systems for frames (or, in practice, very short video sequences) include (Bergen and Meyer 2000) in which a layered representation of a sequence is obtained by finding occlusions between pair of regions. However, the final depth order is obtained by a simple aggregation of local cues with no global reasoning. As a result, the final map is not globally consistent. In (Turetken and Alatan 2009) a forward warping of a computed segmentation is used to determine which layers overlap other layers in the following frames. Pairwise layer relations are then used to construct a depth graph for a global depth ordering reasoning. The depth reasoning step resembles the one proposed Sec. 5.4, where cycles in the graph are identified and some edges are eliminated. The described process is though much more local, and it is performed by examining cycles in the graph in an arbitrary order.

The approach of (Chang et al. 2006), restricts itself to motion parallax depth cues, estimating a layered representation by exploiting the difference of horizontal movements. The authors estimate linear trajectories which they then assign to different layers according to their amount of displacement. The approach involves a lot of processing, such as image transformation to the frequency domain and convolving with a 3D filter. Moreover, the approach processes pixels individually and lacks the concept of regions. Therefore, the resulting partitions involve many small regions and the decision process is not robust. A different approach to estimate depth from short videos is the

work from (Karsch et al. 2012), which attempts to find a full depth map by matching parts of the input video to similar videos and then by propagating depth information to unmatched regions. This kind depth by example-based approach works well for known scenes but its generalization to arbitrary scenes (or even different points of view of the same scene) is very difficult. References (Li et al. 2006; G. Zhang, Jia, T.-T. Wong, et al. 2009) attempt to retrieve a full depth map from a monocular image sequence by exploiting structure from motion in short sequences. However, they involve important assumptions and restrictions about the scene structure that may not be fulfilled in many typical situations. Structure from motion will be assessed more deeply in Chapter. 7.

There exists another kind of approaches, which do no attempt to retrieve a depth ordered partition, but only recover the occlusion boundaries for a given frame. They can be seen as the homologue of figure/ground labeling for single images, although they sometimes involve the processing of several images. For example, the work from (He and Yuille 2010) assumes that the scene is still and assume that occlusion boundaries appear on strong motion and color gradient. They do not attempt to provide a local depth ordering at contours, but the algorithm only outputs a confidence map showing the localization of occlusion boundaries. Following a similar line of work, the approach in (Sundberg et al. 2011) extends the gPb approach (Arbeláez et al. 2011) on single images to single frames. Basically, they introduce the temporal gradient (forward and backward frame difference) as features to the contour gradient estimator. A globalization step of the local gradient detector is performed using spectral clustering, and the region representation using UCMs is kept exactly the same as in (Arbeláez et al. 2011). Results show that introducing dynamic features greatly help the performance on contour detection. As an additional step, the authors show results for figure ground labeling based on optical flow, claiming much better performance than using only static cues. The main drawback of this scheme is that the relative depth is assigned based on a set of local characteristics of the contour and the approach avoids a global reasoning on the depth structure of the scene.

The proposed approach for depth estimation in frames is to use motion occlusion to determine the depth order within a frame given its previous and next frames. No assumptions on the scene stillness are made other than either the camera or the objects are moving. When objects move relatively to the camera, background areas may appear and disappear, providing a reliable cue to determine the depth order. Note that motion occlusion appears when the apparent motion of two overlapping

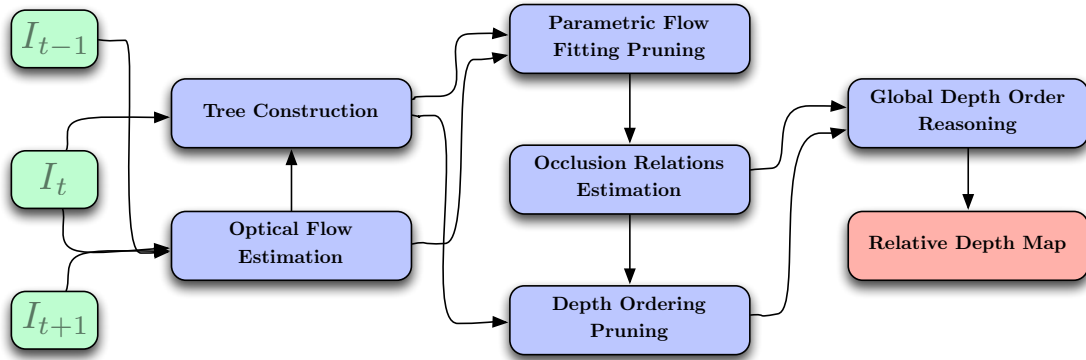


Figure 6.1: Proposed system architecture. Three consecutive frames are used to estimate a depth order map. The system involves an optical flow estimation step and a tree construction. Then, two pruning (graph cut) strategies are applied to extract one partition providing a region-based representation of the optical flow and a second partition involving regions that can be depth ordered. Finally, a global reasoning is used to define a consistent depth order map.

objects/regions is different. This situation occurs either when:

- The real motion of the two objects is different (e.g. two cars in a road)
- The scene is static and the object depths are different (e.g. a building occluding the sky)

To exploit this idea, the system first computes the forward and backward optical flows (*Optical Flow Estimation* block of Figure 6.1). Then, a hierarchical region-based representation of the image is computed and stored in a Binary Partition Tree, BPT (*Tree Construction* block). The goal of this representation is to support robust estimation and global reasoning about relative depth. The use of such representation is essential in our approach. Two ways to construct this representation for frames are explored in this work: one BPT based on color, shape and motion features and one based on Ultrametric Contour Map (UCM) (Arbeláez et al. 2011). The created trees are used to retrieve two partitions using tree cut techniques exposed in Sec 5.3. The first partition allows to fit parametric flow models to regions, finding reliable flow values at occlusion points (*Parametric Flow Fitting Pruning* block) and then obtaining occlusion relations. The second partition is obtained by exploiting these occlusion relations and defines regions that can be depth ordered (*Depth ordering Pruning* block). Since occlusion relations provide depth relations between pair of regions, a final step is needed

to ensure global consistency and to obtain a final depth order map. Besides the algorithm definition, this thesis compares the performance of static versus dynamic cues, showing that motion occlusions are a much reliable cue for depth ordering on video frames than junctions or convexity cues.

6.2 Hierarchical Representation of Frames

6.2.1 State of the Art

As stated in the previous section, there exists little literature specific for hierarchical segmentation on single frames inside video sequences. Most approaches either tackle segmentation in single image or full sequence segmentation, but only few address single frames. The image segmentation state of the art was exposed in Sec. 5.2 and the case for full sequences will be exposed in Sec. 7.2. Therefore, the only approach working explicitly with a hierarchical representation for single frames found to the date is (Sundberg et al. 2011).

Authors extend the segmentation tools for static images in (Arbeláez et al. 2011) to include the motion gradient as an additional channel to brightness, color and texture, see Sec. 5.2. The motion gradient is computed by taking the difference of the current image with the following and previous frames and combining both outputs. If $MG^+(x, y) = I(x, y, t + 1) - I(x, y, t)$ and $MG^-(x, y) = I(x, y, t - 1) - I(x, y, t)$ are the forward and backward motion gradient respectively, both MG^+, MG^- should have high absolute values in the location of the edges of the current frame. The two measures are combined to produce a motion gradient image $MG = (MG^+MG^-)^{(1/2)}$ which is then incorporated to the machinery of the gPb contour detector. The produced soft contour map, the $gPb + \delta$, is used to create a first Ultrametric Contour Map (UCM) to determine which boundaries are actual occlusion boundaries. For each region in this first UCM, the authors reestimate the probability of occlusion boundary using optical flow features and static boundary cues with a linear classifier. The output of this classifier is then feed to another to provide the final hierarchical region representation of the single frame.

This approach is, to the date, the only approach specialized to represent single frames in video sequences as a set of hierarchically ordered partitions. Although the approach gives good results, the creation of the hierarchy comprises many steps, and motion information seems to play a secondary role in favor of static cues for segmentation. Extending the BPT approach for single images, we propose a simpler method that

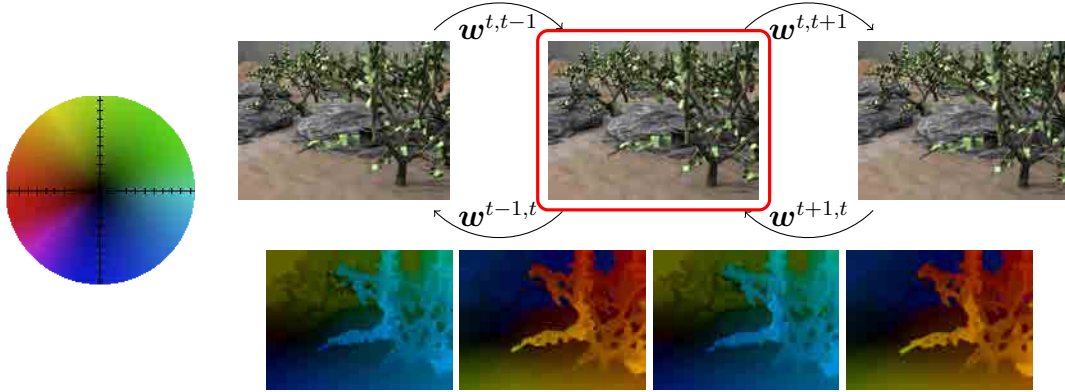


Figure 6.2: Optical flow computation using the next and previous frames. Left: color code used to represent optical flow values. Three consecutive frames are presented in the top row, I_{t-1} , I_t in red and I_{t+1} . In the bottom row, from left to right, the flows $w^{t-1,t}$, $w^{t,t-1}$, $w^{t,t+1}$, $w^{t+1,t}$ are shown.

integrates motion information from the beginning of the tree creation process. Once the tree is created, we process it using tree cuts and make use of motion occlusion cues to estimate the relative depth order between the obtained regions.

6.2.2 Proposed BPT for Frames

Although the hierarchical representation is constructed for a single frame, information from adjacent frames can be used to provide dynamic information such as motion. In (Sundberg et al. 2011) a quantitative evaluation is performed, measuring how the size of the temporal window affects the segmentation performance. Surprisingly, extending to more than two frames (forward and backward) does not help much for segmentation, so gathering information from the previous as next frames proves to be sufficient for most of the cases.

To this end, prior to segmenting the frame, the optical flow between forward and backward frames is computed. The technique described in (Brox and Malik 2011) was used, due to its compromise between simplicity and accuracy. As shown in Fig. 6.2, for each frame I_t the previous I_{t-1} and following I_{t+1} frames are used. With three frames, forward flows $w^{t-1,t}$, $w^{t,t+1}$ can be estimated. $w^{t-1,t}$ maps each pixel of I_{t-1} to one pixel in I_t . Similarly, $w^{t,t+1}$ maps each pixel in I_t to one in I_{t+1} . Additionally, backward flows $w^{t,t-1}$, $w^{t+1,t}$ are also estimated as shown in Fig. 6.2.

Once optical flows are computed, the creation of the segmentation hierarchy can benefit from motion information. For the purposes of this work, two possible trees have

been considered: a BPT approach created using a similar strategy than for single images, see Sec. 5.2 and the UCM technique proposed in (Arbeláez et al. 2011). The only difference between the proposed BPT for single images and the proposed BPT for frames, is that motion information can be introduced to the region distance and region model. During merging iterations, the BPT uses a combination of color, shape and motion information to determine the region similarity. By contrast, the UCM considers the mean strength of the common contour between R_i and R_j estimated using the gPb . The formal expressions for both trees are:

$$d_{BPT}(R_1, R_2) = d_{area}(\alpha d_{cm} + (1 - \alpha)d_{shape}) \quad (6.1)$$

$$d_{UCM}(R_1, R_2) = \sum_{x \in \Gamma_{ij}} \frac{gPb(x)}{|\Gamma_{ij}|} \quad (6.2)$$

Where R_1, R_2 are two arbitrary adjacent regions. d_{area} , d_{shape} , d_{cm} are the area, contour and color+motion contributions to $d(R_1, R_2)$ respectively. α is a weighting factor between shape and color. d_{area} is defined as for single images in Eq. (5.11). For the reader convenience, the expression is:

$$d_{area} = \log(1 + \min(A_1, A_2)) \quad (6.3)$$

With A_1 and A_2 being R_1 and R_2 area respectively. Also, as in single images in Eq. (5.10), d_{shape} is defined as the relative increase of perimeter of the parent region with respect to the biggest one:

$$d_{shape} = \max\left(0, \frac{\min(P_1, P_2) - 2P_{1,2}}{\max(P_1, P_2)}\right) \quad (6.4)$$

With P_1, P_2 and $P_{1,2}$ being R_1, R_2 and the common perimeters respectively. Each frame I_t is represented with seven channels: three for the *CIE Lab* color space and four for motion information. The four motion channels correspond to horizontal and vertical fields for the forward $\mathbf{w}^{t,t+1} = (u, v)^{t,t+1}$ and backward $\mathbf{w}^{t,t-1} = (u, v)^{t,t-1}$ optical flows. The color model for each region is the same as for static images: a full 3D adaptive histogram represented by the 8 most representative colors. For the motion channels a similar approach is followed, and the motion model consists of a joint 4D histogram represented by the 8 most representative motions in the region. The way in which the adaptive motion histogram is estimated is the same as in the color case, see Sec. 5.2. To compare color and motion models for two adjacent regions, two EMD distances are used:

$$d_c = EMD(\mathbf{s}_{c1}, \mathbf{s}_{c2}) \quad (6.5)$$

$$d_m = EMD(\mathbf{s}_{m1}, \mathbf{s}_{m2}) \quad (6.6)$$

Where the distance d_c compares the two color signatures s_{c1}, s_{c2} for both regions and the distance d_m compares the motion s_{m1}, s_{m2} signatures. The definition of cross bin costs for color are the same than for single images. According to (Shepard 1987), this way of comparing perceptually two magnitudes can be used in many cases, such as shape or size. Therefore, it is assumed that differences in motion also obey a 'perceptual law' and the cross bins unit cost between two motion can also be expressed as:

$$c_{ij} = 1 - e^{-\Delta_{ij}/\gamma} \quad (6.7)$$

with Δ_{ij} as the euclidean distance between color or motion vectors and γ an a priori set parameter which, in the case of color was fixed to 14 (Ruzon and Tomasi 2001). γ can be fixed for color because colors have an absolute value, but motion can be almost arbitrary in the scene. For example, a motion difference of two pixels can be very representative in very still scenes. However, this same difference in scenes with objects moving very fast may be due to optical flow estimation error. To avoid setting a fixed parameter for motion, the maximum and minimum motion are found in the frame:

$$\mathbf{m}_{max} = \max_{q,p} |\mathbf{w}^{t,q}(\mathbf{p})| \quad (6.8)$$

$$\mathbf{m}_{min} = \min_{q,p} |\mathbf{w}^{t,q}(\mathbf{p})| \quad (6.9)$$

where $q = t + 1, t - 1$ and \mathbf{p} can be every point in the image. The decay parameter for (6.7) is set to $\gamma = 0.25(\mathbf{m}_{max} - \mathbf{m}_{min})$. In this way, whether the scene has large or small motions, motion differences are scaled properly.

6.2.3 The UCM for Frames

Since the code to build the hierarchy in (Sundberg et al. 2011) could not be accessed, experiments were irreproducible without the original code due to the complexity of the algorithm. Therefore, the classic $gPb - OWT - UCM$ chain of (Arbeláez et al. 2011) is used to produce a binary partition tree, as in Sec. 5.2. Although motion cannot be exploited during the segmentation process using this particular technique, the quality of the regions is comparable (or even better than the BPT approach), so its use can be justified, see Sec. 6.4.

6.3 Depth Ordering

Proceeding similarly as in the single image case, when the tree is constructed it is further analyzed to obtain proper partitions which represent objects as accurately as possible. In the case of single frames, there is an extra difficulty regarding the estimation of motion occlusions. As explained in Sec. 3.2, the optical flow field in occluded areas is normally unreliable, unless the estimation algorithm deals explicitly with this case. Therefore, prior to depth ordering, reliable flow field should be somehow provided for motion occlusion estimation.

The idea of the depth ordering step of the algorithm is to perform two tree cuts. The first cut is performed to identify homogeneous flow zones so that occlusion areas contain valid flow values. From the modeled flows, the occlusion points and relations are computed to provide local depth relationships. The second and final cut is the cut generating the final depth partition. The concept of the final tree cut is similar to the one for similar images: preserve as many low level depth cues as possible while maintaining a reasonable partition granularity. In this case, a local depth cue is considered to be an occlusion relation estimated from flow values. An occlusion relation is composed by two pixels: the occluded p_u and the occluding p_o .

6.3.1 Tree Cut for Parametric Flow Fitting

The process to estimate motion occlusions was shown in Sec. 3.2. For simplicity matters, the occlusion estimation process assumed that a partition was available for parametric flow fitting. Since in practical cases the partition is not available and should be found, here an approach based on tree cuts is proposed to find a segmentation of the frame. In Sec. 3.2 the chosen flow model to fit to each region was a projective flow model (Kanatani 1988) is used. For the reader's comprehension, a short explanation follows. The flows $\tilde{w}_{R_i}^{t,q} = (\tilde{u}, \tilde{v})$ with $q = t \pm 1$, associated to region R_i are expressed as a quadratic model on the x and y coordinates:

$$\begin{aligned}\tilde{u}(x, y) &= a_1 + a_2x + a_3y + a_7x^2 + a_8xy \\ \tilde{v}(x, y) &= a_4 + a_5x + a_6y + a_7xy + a_8y^2\end{aligned}\tag{6.10}$$

where $(x, y) \in R$. As said, in Sec. 3.2, this parametric model assumes that objects can be approximated as rigid planar surfaces moving and rotating with respect to the camera. As objects normally lie far from the camera, this assumption often holds.

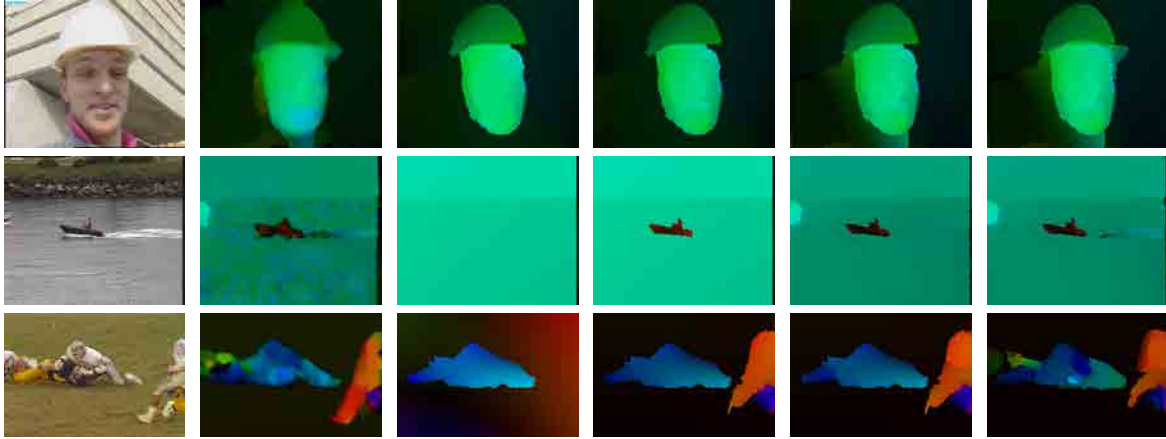


Figure 6.3: Three examples of flow fitting with different partition granularities. From left to right: reference frame, raw estimated optical flows, and four flow fittings from coarser to finer partitions.

To limit the computational load, this flow fitting is applied on the tree nodes that are close to the tree root. Typically, the nodes corresponding to the last thousand merging steps are kept and the remaining nodes corresponding to earlier merging steps are discarded. Once the parametric flow is estimated for each region, a partition P_f representing the regions that best fit to these models is computed using tree cuts, and the energy (5.15) is adapted as:

$$e_i = \sum_{q=t\pm 1} \sum_{x,y \in R_i} |\mathbf{w}^{t,q}(x,y) - \tilde{\mathbf{w}}_{R_i}^{t,q}(x,y)| + \lambda_f \quad (6.11)$$

The constant λ_f can be varied to control the degree of coarseness of the obtained partition. Higher values of λ will provide coarser partitions. As it is an intermediate step of the system, the value was fixed to $\lambda_f = 4 \times 10^3$. It was found experimentally and proved not to be crucial for the overall system performance. A few examples are shown in Fig. 6.3 by varying λ_f , showing that fitted flows correspond to independent moving objects.

6.3.2 Occlusion relation estimation

Motion occlusions are estimated according to the algorithm exposed in Sec. 3.2 with the partition and a few particular examples can be seen in Fig. 6.4. Note that the system is able to handle a wide variety of situations close views, arbitrary landscapes and scenes with few and large movements.

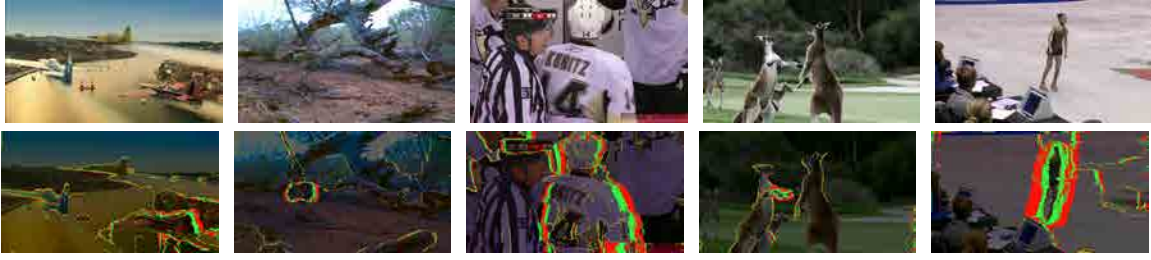


Figure 6.4: Motion occlusions examples. First row: original reference frames. Second row: occlusion relations marked for each frame. Occluded and disoccluded points are marked in red, while occluding points are marked in green.

6.3.3 The Depth Order Graph for Frames

6.3.3.1 Depth Ordering Tree Cut

Equation (3.43) creates a set of pixel pairs (p_u, p_o) for which a depth information is available. If both pixels belong to the same region, they are discarded but if they belong to two different regions, we can conclude that there is one evidence that the two regions belong to different depth planes. In the context of regions described by hierarchical representations, if we deal with regions that are close to the root, many (p_u, p_o) pairs are discarded because regions are very large. By contrast, if regions are close to the leaves, many (p_u, p_o) pairs will be preserved.

To extract from the BPT a partition P_d involving regions that can be depth ordered, an tree cuts strategy is again used. Here the energy to be optimized should be a compromise between the number of occlusion relations, that is of (p_u, p_o) pairs, that are kept and the simplicity of the partition in terms of region number. As a result, the pruning is done particularizing the energy (5.15) as:

$$e_i = \sum_{(p_u, p_o) \in R_i} \frac{1}{N_o} + \lambda_o \quad (6.12)$$

where N_o is the total number of estimated occlusion relations.

6.3.4 Final Depth Ordering

A similar strategy as for single images is followed. The only change is how the weight of the DOG are defined. We review here the main steps of global depth reasoning, but a detailed explanation is given in Sec. 5.4.

A graph $G = (V, E)$ is constructed where vertices V represent the regions of P_d . A directed edge $e_i = (a, b, p_i)$ is defined between node a and node b if there are occlusion

relations between region R_a and region R_b . The weight of e_i is $p_i = N_{ab}/N_o$ where N_{ab} is the number of pixels from R_a which have been estimated as occluding pixels of R_b and N_o is the total number of occluding pixels. The graph G can be seen as a network of (un)reliable links, with the edge $e_i = (a, b, p_i)$ connecting a and b with probability p_i . In this context, a precedes b in depth (a is in front of b) with probability p_i . For two arbitrary nodes of G , the probability of precedence (PoP) can be computed even if there are no edges directly connecting them. If there exists more than one path from a node a to b , the probability of “ a to precede b ” is called ρ_{ab} and is the probability that at least one path between a and b is reliable. ρ_{ab} can be computed by complete state enumeration and the inclusion-exclusion principle (Terruggia 2010).

6.4 Results

6.4.1 Quantitative Evaluation

Due to the lack of state of the art methods for single frames, there is only one possible benchmark on f/g labeling: (Sundberg et al. 2011). Nevertheless, the method could not be accessed and its implementation is not reproducible to a full extent. Therefore, comparison is done only within variants of the system (basically varying the type of hierarchy used) and the system of depth ordering for single images. The three compared systems are:

- The proposed BPT with motion information, estimating depth with motion occlusions (BPT+MO)
- The proposed UCM without motion information, but estimating depth with motion occlusions (UCM+MO)
- The proposed BPT+MD method of Sec. 5.5.
- The proposed UCM+MD method of Sec. 5.5.

The chosen dataset for comparison is the Berkeley Motion Segmentation Dataset (BMSD), proposed by (Sundberg et al. 2011), the only dataset of sufficient visual quality that can be found public. The BMSD is particularly challenging, as there exists a large variety of situations: small/large movements, low contrast scenes, blurring, atmospheric artifacts. This dataset also provides with a segmentation of some reference frames in the sequence and the figure/ground markers for the boundaries. Since the proposed

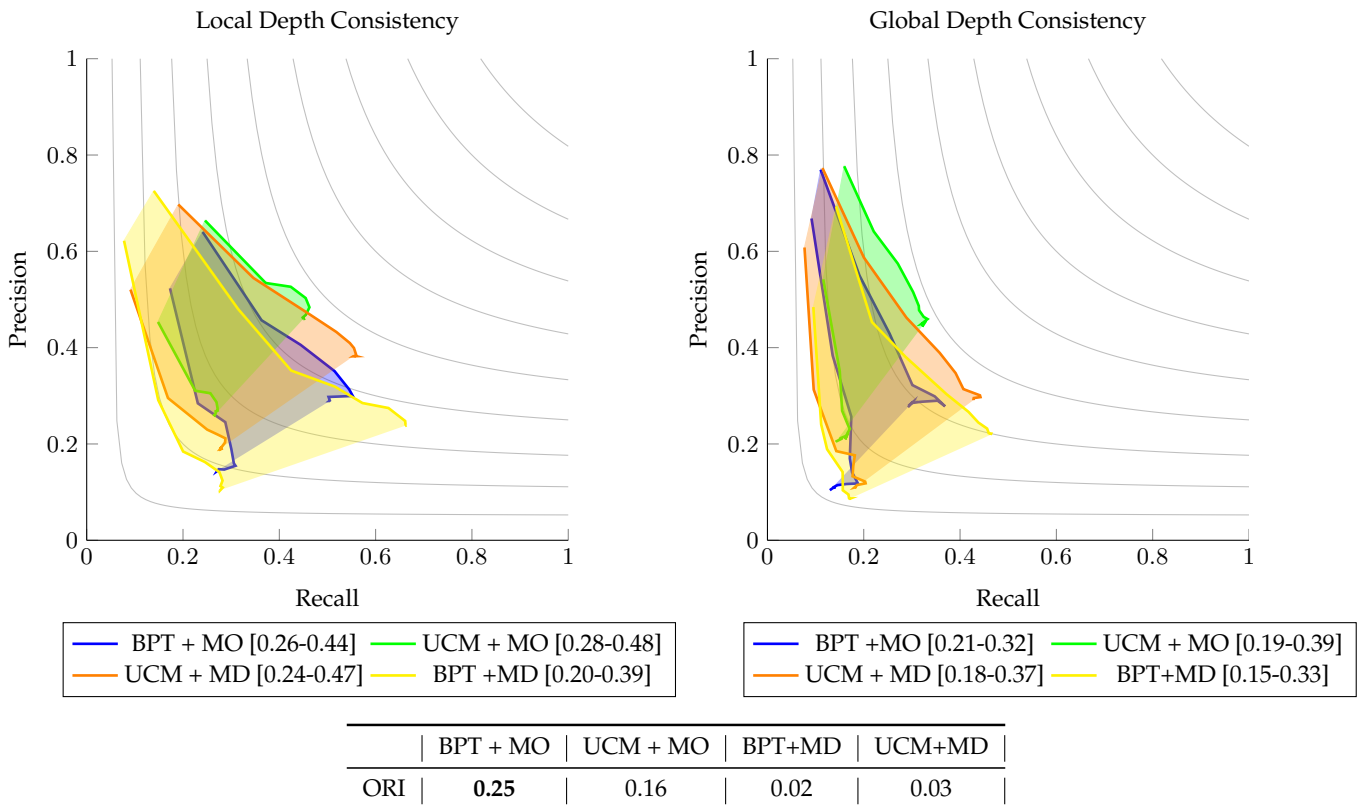


Figure 6.5: Results of the LDC (left) and the GDC (right) measures for the BMDS dataset. The ORI measures of the three systems are shown in the bottom table.

systems need relative depth annotations, the figure/ground labels were adapted to produce depth ordered partitions.

Results on the LDC and GDC measures follow the case for single images: a global consistency is much harder to obtain than a local estimation of depth gradient. One of the reasons to include the BPT+MD and UCM+MD methods into the evaluation of motion cues is to compare the reliability of both static and dynamic cues. From the LDC and the GDC measures it is possible to see that the region for the BPT+MD is much wider than the systems BPT/UCM+MO, meaning that there exist more inconsistent detections in the estimation using static cues. That is, static cues are less reliable than dynamic ones. The reason for this behavior comes basically from two factors:

- T-junctions / convexity cues do not always agree at low-level with the correct depth. See the performance of these cues with groundtruth segmentations in single images in Sec. 5.5.

- The estimation of motion occlusion is based on optical flow. Nowadays, the flow estimation problem is very mature and there exists high quality algorithm able to estimate it with high reliability. Junction estimation, on the other hand, is still a non-solved problem.

This difference can also be seen in the ORI index, where the BPT/UCM+MD cannot deal with such a challenging dataset (BMDS). It should be said that the purpose of this dataset was to segment object according to their motion, so there are many sequences where there exists strong color camouflage and low contrast which is captured by dynamic cues but not by static characteristics because objects move differently. Both systems BPT+MO and UCM+MO show that motion occlusions are much more reliable by presenting ORI indexes of 0.25 and 0.16 respectively.

LDC performance Although motion is introduced in the BPT+MO segmentation tree construction, it doesn't seem to be able to reach the UCM+PO detection performance ($F=0.44$ versus $F=0.48$ for the BPT and UCM respectively). However, the BPT+MD has even a lower score ($F=0.39$) meaning that motion actually does help into the BPT creation process. The differences with the UCM+MO are the additional texture and multiscale features the algorithm ([Arbeláez et al. 2011](#)) introduces to the detection of the gradient. Although the BPT is a rather simple approach both in its concept and the kind of features used to compute region models, it is able to reach competitive results with state of the art algorithms. Classification scores are clearly higher for the systems using motion occlusions, meaning that these kind of cues can be more trusted than static ones. Such effect can be clearly seen when comparing th UCM+MO and the UCM+MD, where the same hierarchy is used but only the type of monocular cues used are changed. Although the detection score of both system are similar (due to differences in the tree cut process), it can be seen that the classification score for the UCM+MD is lower ($F=0.24$ versus $F=28$), showing that motion occlusions are indeed more reliable than static cues. On the four systems there is a clear behavior: as recall gets lower, depth assignments on confident contour become easier. That is, motion occlusions on confident contours are clear and indicate the correct depth order, giving fewer inconsistent detections. The lower score of the UCM+MO with respect BPT+MO may be due to the fact that the UCM tree discards motion for its construction. Although the UCM+MO detection performance is higher that the BPT+MO, the classification performance is similar for both systems (the lower part of the region) and therefore the ORI index is higher for the BPT+MO. This may be caused by the fact that the UCM is able to capture more boundaries thanks to its high-performance

contour detector, but these contours do not contain motion information. This causes the system to make random guesses on the depth assignment.

GDC performance The three systems present similar global detection scores, although the BPT+MD shows much more inconsistency as its region is much wider. The two best systems, the BPT+MO and UCM+PO provides similar performances for both detection and classification. Overall, either the BPT+MO or the UCM+MO could be chosen as the reference system for depth ordering on single frames. If better segmentation is desired the UCM+PO should be the choice, but if better accuracy on the depth ordering is a must, the BPT+MO should be the reference.

6.4.2 Qualitative Evaluation

To show the advantages and limitations of using motion occlusions, visual results are shown for two datasets. In addition to the BMDS, results from the Carnegie Mellon Dataset (CMU) (Stein 2008) are compared. The CMU dataset contains short video sequences, and some of them have poor quality and have compression (blocking) artifacts. So, as stated also in (Sundberg et al. 2011), results on this dataset are only compared qualitatively. Due to the poor quality of the CMU dataset, instead of choosing three consecutive frames, the first, the reference and the last frame of the short sequence are chosen. Each sequence is about 8 frames long, and this setup was designed so as to generate larger motion between images which could, to some extent, overcome the distortions due to compression artifacts.

Results of depth ordering can be seen in both Fig. 6.6 and Fig. 6.7, showing that motion occlusions may work over a variety of situations: static scenes, moving foregrounds, moving background or even multiple moving objects. Moreover, the algorithm does not assume anything about the kind of observed scene nor the kind of objects. Non rigid and deformable objects are treated naturally.

Motion occlusions are thus a good depth cue to determine the relative order between objects on the scene, although it presents some limitations. First, motion occlusions work with all translational and certain types of rotational motion. More formally, as stated in (Meinhardt-Llopis et al. 2011), object rotations with respect to an axis not perpendicular to the image plane can create self occlusions. When objects rotate, parts of it disappear behind them, and optical flow estimation algorithms cannot handle correctly these cases.

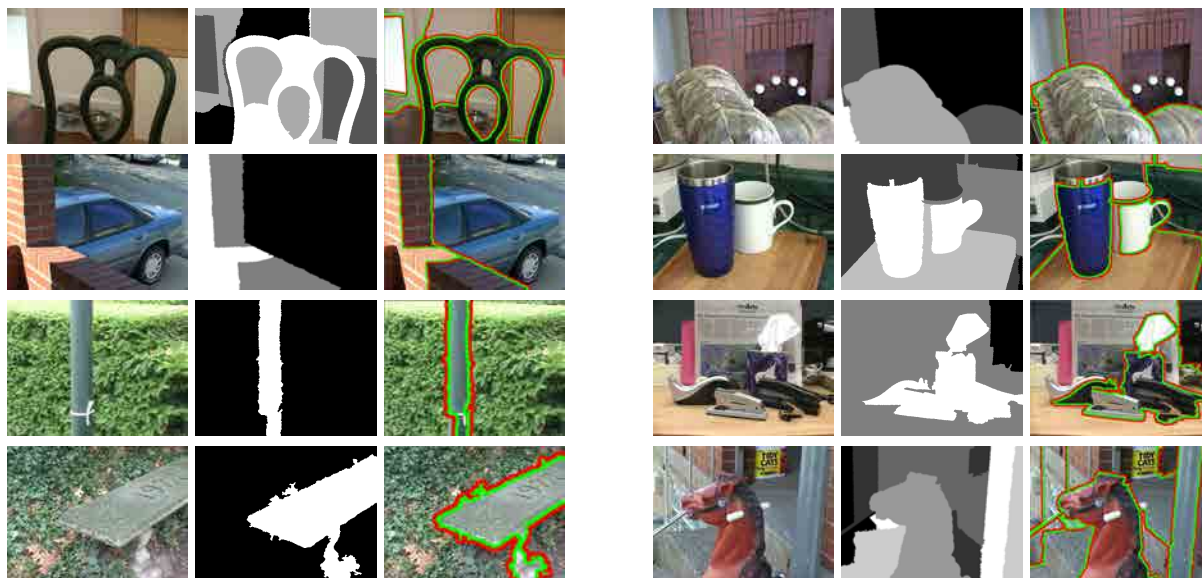


Figure 6.6: Results on the CMU dataset for the BPT+MO case. From left to right, for the two columns. 1) Keyframe image and 2) Image showing the estimated depth partition, with white regions meaning closer and black meaning further. 4) Figure/ground assignment on contours, with green and red overlaid marking figure and ground regions, respectively.

In practical situations, object rotation can be considered negligible from frame to frame as objects lie far from the viewer. However, if rotation is strongly observed, the assumption about motion occlusions do not fulfill and thus, the algorithm breaks. For instance, if an objects rotates, points disappearing behind the same objects will seem to be occluded by itself. Nevertheless, this strong rotations appear hardly and the algorithm is able to overcome the small orientation changes.

Second, the smoothness enforces by the optical flow estimation algorithm causes two kinds of (related) effects on both segmentation and occlusion estimation. Due to the aperture problem, the estimation of the flow field is governed by a smoothness term which, sometimes, can produced oversmoothed flow fields, the contours of which do not coincide with real color edges. Small details and sharp and curved edges may be missed sometimes by flow estimation and, most of the time, optical flow and color edges do not coincide spatially, see Fig. 6.8. Since occlusions appear at motion edges, the introduction of segmentation to occlusion detection played a key role on the performance of the system. Additionally, the optical flow algorithm cannot handle very big occlusion regions. As the occluding region grows, instability of the flow estimation

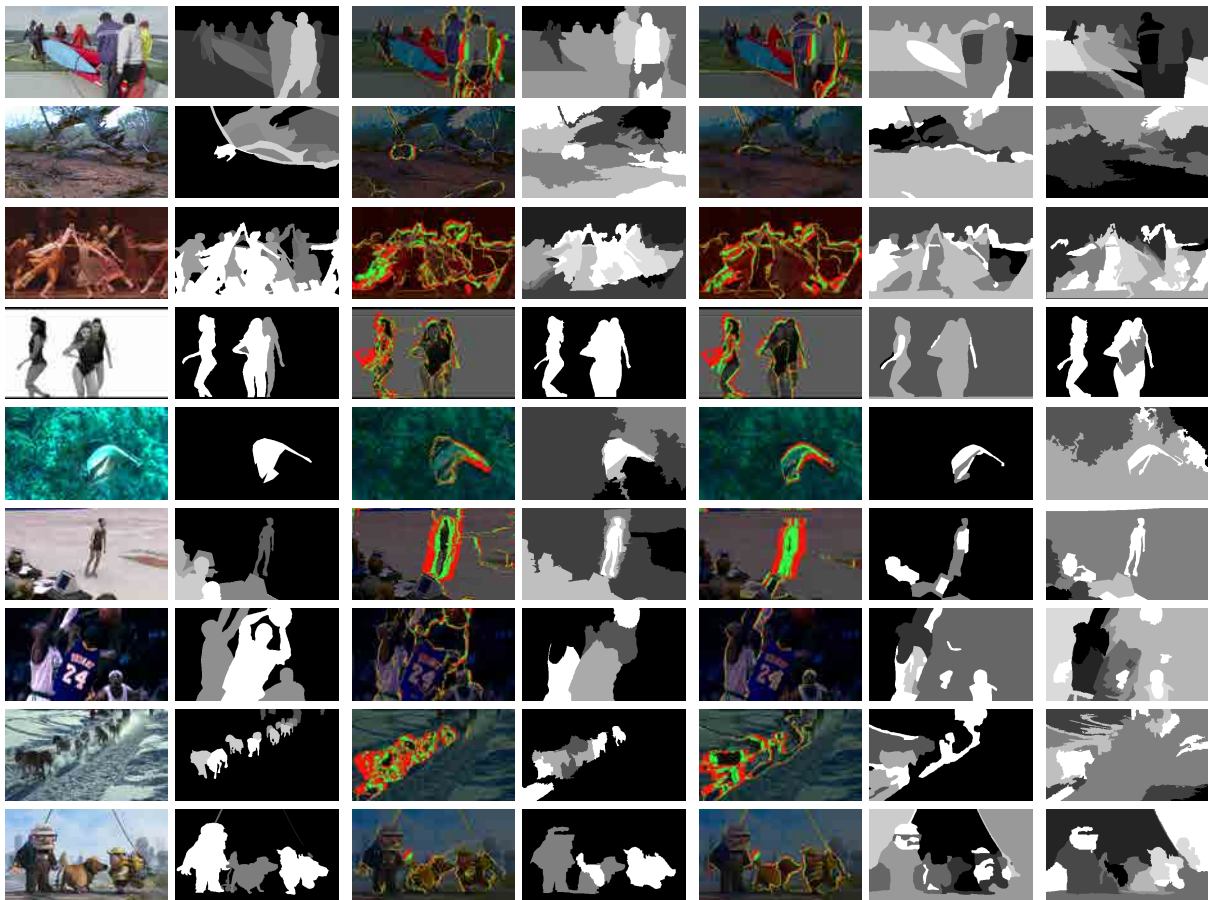


Figure 6.7: Results on some of the sequences of the BDS dataset. For each row, from left to right: original image, groundtruth, occlusions for BPT+MO, depth from BPT+MO, occlusions from UCM+MO, depth from UCM+PO and depth from BPT+MD.

increases, leading to poorer segmentation performance, as well as an increase of the error in flow model estimation. Therefore, if very large displacements are observed, the algorithm is likely to fail on these cases. Nevertheless, if the frame rate is high enough, only small movements will be observed and the algorithm will be able to correctly retrieve occlusion information.

When relative depth from single frames is obtained, an immediate extension is to apply the method for full video sequences. The next chapter exposes the advantages and the challenges of estimating depth in full video sequences, and presents an approach to estimate depth from a series of frames.



Figure 6.8: Figure showing how in the left reference frame and in the optical flow estimated field (center), the edges do not coincide. To show it, both images are overlapped in the right image.

7 Depth Ordering of Video Sequences

7.1 State of the Art

In the last section we proposed a system able to infer relative depth from a video frame. Although this task is computationally less expensive than doing the same thing for full video sequences, the literature on the first problem is much more abundant. The data that should be processed for video is one order of magnitude higher than that for single frames/images but, in contrast, the amount of information related to a spatio-temporal signal is also much richer. For this reason researchers tackled the depth estimation problem on videos much before than depth for single images and frames. Having a series of temporal consecutive frames provide many cues unavailable for single frames (not to mention single images), and a lot of redundancy is present so systems are less error prone. Of course, this comes with the price of the amount of data to be processed.

One of the key aspects of video is that motion can be estimated in many ways. One of these approaches, as commented in Sec. 6.1, is optical flow. Flow fields provide a dense mapping between two frames, where each pixel in a reference frame has a correspondence to another point in a referred frame. Another way to estimate motion is to adopt a sparse approach, identifying several key points in each frame and then match them to previous/next frames. Since key points are points with a visible structure, they appear sparsely located in frames, mainly in corners and edges, (Shi and Tomasi 1994). Matched pairs of key points allow to estimate a general motion from frame to frame which, under some assumptions, can provide cues of the depth structure of the observed scene.

Motion and structure are closely related since the first appearance of flow algorithms

Contributions on Depth Ordering on Video

- G. Palou and P. Salembier. “Hierarchical Video Representation with Trajectory Binary Partition Tree”. In: *IEEE CVPR*. Portland, OR, USA, 2013
- G. Palou and P. Salembier. “Hierarchical Video Representation with Trajectory Binary Partition Tree and its Applications”. In: *IEEE TPAMI, in peer review* 2013

(B. K. P. Horn and Schunk 1981). Few years later, (Koenderink, Van Doorn, et al. 1991) was one of the first approaches exploiting motion to reconstruct simple structures by relating points within different frames. Two years later, (Cipolla et al. 1993) presented a system recovering the structure of hand gestures on controlled environments by exploiting motion parallax. With the increasing of computer power, in (Szeliski 1996), the authors proposed to stitch the frames of video sequences into a composite mosaic, showing that the relative position of images provide cues to, at least, infer a projective structure of the observed environment (see Sec. 8 for a detailed explanation on projective geometry). In the same year, (Triggs 1996) proposed a method to relate a set of tracked points visible throughout the sequence with the camera projection matrices. Apart from the approach in (Szeliski 1996), all the other approaches did not work with real sequences.

Experiments of (Kanatani 1988) showed that the apparent motion (optical flow) and the structure of objects is closely related. The form of simple objects (planes and spheres) and their generating apparent motion was explored showing that, for example, planar surfaces under rigid motion can be expressed as a second order parametric models. However, it wasn't until (Kanatani, Shimizu, et al. 2000) that motion and structure were closely related, when the optical flow fields were used to estimate the fundamental matrix between two views (R. Hartley and Zisserman 2004). Subsequent works kept relating multiview geometry and optical flow (Wedel, Pock, et al. 2008), or even combining both approaches (Mainberger et al. 2008), improving the reconstruction of objects from a set of views. More recent approaches showed that indeed, under static scene assumptions, motion and disparity are equivalent and dense depth maps can be estimation using only motion information (G. Zhang, Jia, T. T. Wong, et al. 2008; G. Zhang, Jia, T.-T. Wong, et al. 2009).

Most of the approaches exploiting motion to estimate depth are known as structure-from-motion systems. They all follow a similar chain of processing: 1) key point extraction and tracking, 2) initial sparse structure estimation and 3) dense reconstruction upgrade. In the first stage of these systems, a set of characteristic points are identified and tracked across frames. In the second, the 3D position of these points is triangulated, providing a sparse three dimensional structure. The last step interpolates the obtained depth to build dense surfaces and depth maps for visualization. For some applications, obtaining a sparse structure of the scene is sufficient. For example, in these approaches, (Davison and Murray 1998) was one of the first real time systems to track a set of features and position them in a 3D space for autonomous vehicle navigation. These approaches, known as Simultaneous Localization And Mapping (SLAM)

have evolved since the 90s with (Davison, Reid, et al. 2007). Recently, GPU computing power allows to obtain an upgraded version of SLAM systems. These approaches, commonly known as D-SLAM (D for Dense), are characterized by their capacity to obtain a real time and dense structure from a video sequence (Stühmer et al. 2010). Still, the previous approach needed a lot of computing power, and its implementation is restricted to devices with high parallelism capacity, such as GPU. Still nowadays, in the research community, the common approach to follow is to identify a set of sparse points, triangulate their position from a set of views to obtain their 3D structure and then interpolate to obtain a dense surface reconstruction.

There are many approaches exploiting these ideas, but only the most characteristics will be referred here. For a full survey on the methods, see (Ponce, Forsyth, et al. 2011) which presents a detailed explanation and many state of the art algorithms. One of the major works in this field, is the PhD thesis (Pollefeys 1999), which summarizes most of the structure from motion approaches. Moreover, systems proposed in the thesis were improved in (Pollefeys et al. 2004), which is nowadays the reference for all the structure from motion algorithms. The authors in the paper propose a robust system which is able to estimate a dense surface from a static scene. All the steps of the systems are carefully explained and some of them are incorporated into the book (R. Hartley and Zisserman 2004). Until the date, this book is considered to be state of the art algorithms of typical structure from motion. A posterior system, (Li et al. 2006) attempted to merge monocular depth cues to the classic architecture of structure from motion. In a first step, a classic structure from motion system is run and when a motion degeneracy is found, a depth from monocular static cues is applied to the system. A motion degeneracy is a particular motion that does not allow to recover structure from motion (Yan and Pollefeys 2006). Although the authors devise the whole system structure, the analysis of the depth by static cues (occlusion, convexity, etc) is left as a future work.

Many structure from motion algorithms suppose that the observed scene is totally static and can thus be represented as rigid objects. This essentially makes video sequences and multiview systems totally equivalent. So, taking pictures at different time instants is not a problem. The only thing that may be different from (multiview) stereo systems is that the relative position of the cameras and their calibration matrices may be unknown and should be found by autocalibration (O. D. Faugeras et al. 1992). However, if objects move, the structure recovery task is much more difficult and many researches adopt a layered approach to represent depth for video sequences with arbitrary moving objects.

One of the first approaches exploiting layers, ([Ayer and Sawhney 1995](#)), imposes a parametric model of the motion to each layer and then proposes an expectation-maximization (EM) algorithm to iteratively estimate the number of objects and the pixels belonging to each object. Although the system does not provide relative depth ordering between objects, it is significant as one of the first works proposing motion segmentation on video sequences. A similar work ([Torr et al. 1999](#)) proposes to represent a sequence as a collection of approximately planar layers that are arbitrarily positioned and oriented in the scene. The steps of the algorithm are like in ([Ayer and Sawhney 1995](#)), where a EM algorithm is used to estimate the number of planar objects and which pixels belong to each object. Other approaches appeared in the following years, such as ([Jojic and Frey 2001](#)). Still, the common point between all of these approaches is that they try to 1) determine the number of objects in the scene 2) assign to each pixel an owner object. Since this is a kind of 'chicken and egg' problem, most people propose to solve it by generalized expectation-maximization techniques, as the articles here cited.

A different approach is taken in ([Bergen and Meyer 2000](#)) where a first motion estimation and segmentation algorithm is proposed. As a post processing step, the authors analyze the motion estimation errors to derive from them a relative depth order. This idea is similar to the approach for depth ordering in single frames, see Sec. 6.3, where occlusions are detected due to some failure of optical flow bijective properties. The approach is novel in the sense in which it exploits motion errors, but the depth ordering part does not enforce global consistency. A much later but also similar approach, ([Turetken and Alatan 2009](#)), does enforce global consistency by eliminating iteratively cycles on the generated depth order graph. Occlusions in this case are obtained by forward warping previously segmented region from the reference frame.

Other approaches aim to first obtain a correct segmentation of a video sequence, and then reason about occlusions ([Konrad and Ristivojevic 2003](#)). By using graph cuts ([Kolmogorov and Zabih 2001](#)), the authors estimate disparity in video sequences incorporating occlusion information. Although occlusion is integrated within the structure-from-motion algorithm in this approach, only the structure static scenes with moving cameras can be correctly retrieved. Other approaches prefer to obtain layered depth representation: ([Chang et al. 2006](#)) exploits motion parallax and detects movement and occlusions using a multidimensional filter. Similarly, a layered representation is obtained in ([P. Smith et al. 2004](#)) by tracking edges to detect the occluding and the occluded sides. Nevertheless, this approach needs to know in advance the number of frames to be processed and the number of depth layers present.

One of the most recent approaches ([Lezama et al. 2011](#)) attempts to obtain a full dense

video segmentation by incorporating depth order into the segmentation information. The algorithm establishes points trajectories from optical flow which then clusters with a predefined number of clusters. The clustering method is done by minimizing an energy using occlusions, obtaining a depth ordered sparse segmentation. Although layered approaches can estimate the relative depth of objects in moving environments, they have a drawback: they cannot capture the 3D structure of each individual object/region. This fact may not be that serious for single images and frames, where this information is hard to extract, but obtaining absolute depth from video sequences is proven to be feasible.

Although later than structure from motion algorithms or layered representations, hybrid approaches attempting to estimate the 3D structure in spite of having multiple moving objects appeared several years ago. One of the first approaches ([Costeira and Kanade 1998](#)) generalized the concept of projective factorization, allowing to introduce multiple rigid motions to the same scene. The problem of this approach is that obtaining the number of different motions is hard and sometimes the algebra lying behind the approach is very sensitive to noise. Similarly, in ([Fitzgibbon and Zisserman 2000](#)) a RANSAC approach is proposed to detect the number of different motions. Once motions are detected, point 3D position is triangulated if a sufficient number of matches and views are present to perform a self calibration of the camera. Rather than proposing an iterative procedure to detect the number of independent motions, in ([Vidal et al. 2006](#)) a different approach is taken. The concept of the fundamental matrix, which relates the points seen from two views, is extended to include an arbitrary number of motions. Although theoretically feasible, the proposed approach has difficulties on estimating the number of motions in practical cases when noise is present.

These three approaches ([Costeira and Kanade 1998](#); [Fitzgibbon and Zisserman 2000](#); [Vidal et al. 2006](#)) offered a sparse structure associated to interest points in environments with multiple moving objects, but the dense structure of the scene is not retrieved. Recent approaches such as ([G. Zhang, Jia, Hua, et al. 2011](#)) offers the possibility to represent a video sequence with two independent moving layers. The difference between this approach and the previously cited ones is that the structure of each layer is estimated so they are not considered planar views. Similarly, in ([Karsch et al. 2012](#)) the depth of an input video is estimated by searching similarly structured videos on a groundtruth dataset. Although technique works well for stationary video with some moving objects, it seems that the system is not scalable to obtain depth for arbitrary scenes and camera movements.

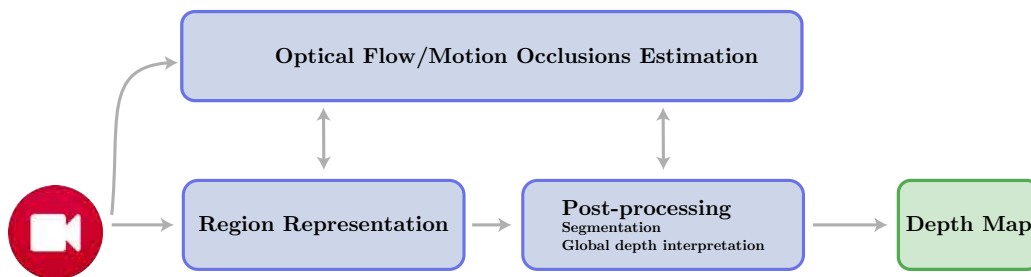


Figure 7.1: Particularization of the system architecture in Fig. 2.14 for the video case

In this thesis we address the principal problems of the literature on depth estimation in video sequences. First, we propose a new video representation by extending the Binary Partition Tree of single images/frames to provide a basis for region multiresolution analysis of video sequences. Next, and following the proposed algorithm for depth ordering for single frames, we propose a depth ordering algorithm for full video sequences without knowing the type of scene present. A particularization of the schema in Fig. 2.14 can be found in Fig. 7.1 for the video case. In a final section we address the problem of structure from motion using optical and, supposing that the input scene is static, we estimate dense depth maps and full 3D structure using only optical flow information.

7.2 Hierarchical Representation of Video Sequences

7.2.1 State of the Art

Since normally video is seen as a sequence of temporally related images, video processing algorithms are often an extension of image processing techniques. For example, the well known efficient graph based image segmentation (Felzenszwalb and Huttenlocher 2004) (GB) is proposed in (Grundmann et al. 2010) (GBH) by extending the algorithm to create a hierarchy of partitions. A mean-shift algorithm (Paris and Durand 2007) is also adapted for temporal sequences in (Paris 2008) (Meanshift). The approaches (C. Fowlkes et al. 2004) (Nyström) using normalized cuts and (Corso et al. 2008) (SWA) proved to be scalable in complexity when the time dimension is added. The basic part of these algorithms is to consider the original pixel grid as a graph, where nodes represent image pixels and weighted edges represent connectivity. After

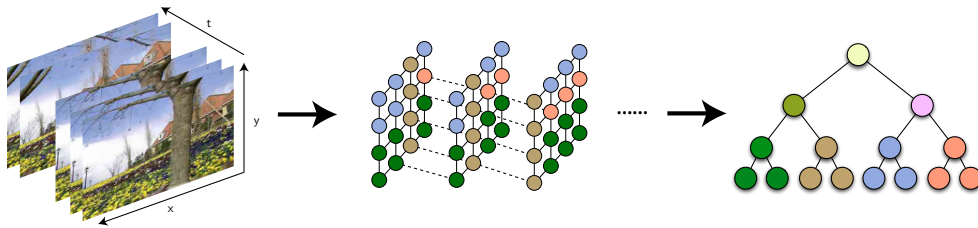


Figure 7.2: Outline of the proposed approach. Starting from the video frames, the first step identifies reliable trajectories between frames representing long term spatiotemporal coherent part of the scene (shown as dashed lines). Then, the algorithm constructs a Trajectory Binary Partition Tree by iteratively merging neighboring trajectories and builds a hierarchical representation of the entire sequence. Note: The node color approximately represents the mean color of the trajectory regions.

the original representation is transformed into a graph, the algorithms group nodes with some clustering techniques such as graph cuts, normalized cuts or greedy merging techniques. Since the image graph representation can be easily extended to more than 2 dimensions, the extension to video is normally done by treating the temporal dimension as a third spatial dimension. As a result, pixel connectivity is extended from the classic 4 neighborhood in images to 6 (Corso et al. 2008) or 26 (Grundmann et al. 2010) neighborhood in both spatial and temporal dimensions. As a result, 3D image segmentation (Wirjadi 2007) and video segmentation essentially become equivalent. The principal problem with these approaches is that the temporal dimension is treated equally as the spatial ones, even if time has different statistics and behavior than space.

Motion is the principal cue that can be extracted once the temporal dimension is available and it can be used alone to provide a sparse segmentation of a moving scene with several objects (Costeira and Kanade 1998; S. Rao et al. 2010; Brox and Malik 2010). Approaches commented in the last section that estimate depth layers in video are closely related with motion estimation algorithms. Depth ordering on video sequences can be seen as a particularization of motion segmentation. Although not exactly equivalent, depth layered representations of videos can be sometimes by concatenating two stages. A first motion segmentations of the video, and a second ordering between the obtained motions.

Motion segmentation algorithms are built under the suppositions that some points in the scene are being tracked over time. Since, until today, not all points can be reliably

tracked, tracking algorithms, known as feature trackers, operate on a subset of points which exhibit a visible and easy identifiable structure over time (Shi and Tomasi 1994). With the appearance of the first dense optical flow estimation algorithms (B. K. P. Horn and Schunk 1981; Black and Anandan 1993), motion was available for all the pixels in each frame. However, these algorithms solved the aperture problem (Nakayama and Silverman 1988) by 'filling' homogeneous zones with smoothness priors. More recent and realtime techniques (Wedel, Meißner, et al. 2009) refine the smoothness priors, but the aperture problem persists. Essentially, the set of points in which dense flow is reliable is similar (although denser) to the set of feature trackers. A current state of the art tracker (Sundaram, Brox, et al. 2010) uses frame-to-frame optical flow to estimate reliable flow regions, creating long term trajectories over time and showing better coverage and performance than feature-based detectors available to date.

If a scene contains several moving objects, chances are that tracked points of the same objects move in a similar way. Examining motion differences between tracks is known as motion segmentation, and many algorithms exist to detect moving entities in video sequences. The most used approaches are graph cuts (Xiao and M. Shah 2005), normalized cuts (Brox and Malik 2010) and low-rank factorization methods (S. Rao et al. 2010; Yan and Pollefeys 2006). The principal limitation of these approaches are that they operate on a sparse subset of points, not providing a dense segmentation of the scene, therefore object boundaries are not available. Sparse representation may be sufficient for some applications such as tracking or activity recognition, but a dense coverage provides more information. It is also possible to combine static segmentation cues (basically color) with optical flow to produce dense segmentation on videos. The most common technique is to use first a motion segmentation for a sparse representation, and then upgrade its density using color (Ochs and Brox 2011). Techniques such as (Ogale et al. 2005; Sun et al. 2010) have been proposed to produce a dense segmentation coverage using dense optical flow and by explicitly treating occlusion. The work (Sundaram and Keutzer 2011) uses temporal frame-to-frame information and the gPb contour detection algorithm (Arbeláez et al. 2011) to compute voxel-based affinities and relate pixels between frames. Affinities are then clustered using normalized cuts (Shi and Malik 2000) and a segmentation is produced by means of ultrametric contour maps (Arbeláez et al. 2011) on the resulting segments. Apart from its computational cost (5 minutes on a cluster of 34 GPUs), the algorithm does not take advantage of long term information introduced by trajectories but relies on the globalization capacity of normalized cuts to propagate motion information. By contrast, the work (Lezama et al. 2011) uses the tracked points (Sundaram, Brox, et al. 2010) to

propose a semi-supervised clustering and uses the obtained labels to produce a dense segmentation. However, the number of objects should be known in advance and, in practice, this information is most of the time unknown. In the work of (Dorea et al. 2009) a proposed extension to binary partition trees is proposed, but the algorithm involves a series of bottom-up mergings and top-down splits to achieve short and long term region coherence. In the proposed extension of the BPT a much natural and simpler approach is followed.

7.2.2 Proposed Hierarchical Video Representation

In this thesis, a scheme is presented so that it integrates the advantages of color and motion segmentation in the same process to produce a hierarchical region representation such as a BPT for video sequences. We discuss a completely unsupervised way to introduce long term motion information and spatial segmentation in a single scheme by extending the BPT algorithm (P. Salembier and Garrido 2000) to video, but taking special care of temporal information. The main difference with the original BPT approach used either in still images and single frames, see Sec. 5.2 and Sec. 6.2 respectively, concerns the elementary units that are iteratively merged. Instead of iteratively merging neighboring pixels, here neighboring trajectories are merged forming a Trajectory BPT (TBPT). The approach is outlined in Fig. 7.2. The system assumes that dense forward and backward optical flow information is available. To run the experiments, the same optical flow estimation than in Chapter 6 is used (Brox and Malik 2011), but other approaches could work as well. Prior to the Trajectory BPT computation, reliable trajectories are defined throughout the sequence using (Sundaram, Brox, et al. 2010) and then spatially quantized to produce the initial partition used as starting point for the BPT algorithm. Unlike (Lezama et al. 2011), trajectories are introduced in a fully unsupervised manner, without prior clustering into a predefined number of classes. The Trajectory BPT is then computed and, at each iteration, the two most similar trajectories are merged. While other approaches consider that color and motion information can be represented in the same way, we take advantage of motion segmentation clustering techniques to design an appropriate color and motion models and similarity measures. We show how the generated hierarchy offer competitive results in comparison with the state of the art segmentation algorithms.

There are three main contributions of this thesis regarding segmentation in video sequences. First, a simple and efficient region merging approach to generate a hierarchy representing whole video sequences. is designed. This task is performed by extending

the BPT algorithm introducing the temporal dimension. the video is represented as a binary tree of trajectory regions (which are set of spatially neighboring trajectories). Second, differently as the state of the art on video segmentation, which considers that video can be treated equivalently as a 3D volume with color and motion, specific color (spatial) and motion (temporal) models for regions resulting from the merging process are devised. Third, during the tree creation process, motion and space are specially addressed separately for segmentation, designing a coherent distance measure which exploits advantages of motion and color segmentation at the same time.

7.2.3 Trajectory Estimation

Point trajectories are a reliable way to propagate long term information along an image sequence. Optical flow based tracking (Sundaram, Brox, et al. 2010), in contrast to descriptor based tracking (Davison, Reid, et al. 2007), provides a denser coverage. In a nutshell, the tracking algorithm (Sundaram, Brox, et al. 2010) finds reliable starting points for trajectories and tracks them from frame to frame using the estimated optical flow (Brox and Malik 2011) until the flow reliability falls below a given threshold. Reliable optical flow estimates can be found at points fulfilling the following three conditions: 1) they have a visible spatiotemporal structure in their neighborhood 2) they do not become occluded 3) they are not on a motion boundary. State of the art optical flow estimation exhibit similar behavior on 'easy' points (Mac Aodha et al. 2013), making the performance of the trajectory tracking stable regardless of the used algorithm. Trajectories obtained with (Sundaram, Brox, et al. 2010) are used as a starting point for the tree creation process. Initial estimates should be quantized to the closest pixel so as to produce an initial partition. Since the flow reliability is used to define the initial trajectories and also to measure the distance between trajectory regions during the TBPT creation process, we present the three reliability notions (Sundaram, Brox, et al. 2010) for a point $\mathbf{p} = (x, y, t)$ in the video.

7.2.3.1 Structure reliability

Optical flow estimation algorithms rely on color and gradient consistency with an additional smoothness term to cope with the aperture problem (Nakayama and Silverman 1988). On points where a strong visible structure is present, mainly corners, junctions and textured regions; the aperture problem is easily handled. According to (Harris and Stephens 1988), points with a visible structure can be found by means of the second eigenvalue λ_2 of the structure tensor: $J_s = K_s * (\nabla I \nabla I^\top)$. $\nabla I = [I_x, I_y, I_t]^\top$ de-

notes a spatio-temporal gradient, K_s is a Gaussian kernel of standard deviation $\sigma = 1$ and the operator $*$ denotes the convolution. A similar approach is used in (Sundaram, Brox, et al. 2010) to set a hard threshold on whether or not a tracking trajectory should be started at a given point. In this thesis, the structure reliability is defined for each point in the video (not only for starting trajectories), and can be expressed as:

$$\rho_s(\mathbf{p}) = 1 - \exp\left(-\lambda_2(\mathbf{p})/\widehat{\lambda}_2(t)\right) \quad (7.1)$$

where $\widehat{\lambda}_2(t)$ is the average second eigenvalue of the current frame. ρ_s behaves like a Harris detector (Harris and Stephens 1988), with $\rho_s \approx 1$ in corners and junctions and with $\rho_s \approx 0$ in points in homogeneous zones.

7.2.3.2 Occlusion reliability

As commented in motion occlusion estimation in Sec. 3.2, object motion make points of the background disappear (occlusions) and appear (disocclusions). To check if a point becomes occluded, a forward-backward consistency is proposed, but an available partition and flow models should be available. Since during the creation of the TBPT partitions change at each iteration (and thus flow models fitted to the regions), a less computationally intense approach is followed. Instead of using regions, an occlusion flow confidence is obtained using the raw optical flow estimation, discarding region information.

Assume that $\mathbf{w}^{t,t+1}(\mathbf{p}) = (u(\mathbf{p}), v(\mathbf{p}))$ is the forward motion field. The backward flow field corresponding to \mathbf{p} is $\mathbf{w}^{t+1,t}(\tilde{\mathbf{p}})$ where $\tilde{\mathbf{p}} = (x + u(\mathbf{p}), y + v(\mathbf{p}), t + 1)$. The flow reliability according to the forward-backward consistency is defined as:

$$\rho_o(\mathbf{p}) = \exp\left(-\frac{|\mathbf{w}^{t,t+1}(\mathbf{p}) + \mathbf{w}^{t+1,t}(\tilde{\mathbf{p}})|^2}{0.01(|\mathbf{w}^{t,t+1}(\mathbf{p})|^2 + |\mathbf{w}^{t+1,t}(\tilde{\mathbf{p}})|^2) + 0.5}\right) \quad (7.2)$$

In the case of non occlusion, $\rho_o \approx 1$, as the forward and the backward flows compensate ($\mathbf{w}(\mathbf{p}) \approx -\bar{\mathbf{w}}(\bar{\mathbf{p}})$). In (Sundaram, Brox, et al. 2010), $\rho_o \approx 0$ indicates that \mathbf{p} is being occluded and thus the tracking should be stopped.

7.2.3.3 Motion boundary reliability

At points with strong motion gradients the flow contains motions from two different objects. Therefore, in motion edges, the estimated flow may not correspond to the true motion of the objects, so their corresponding reliability should be low. A possible

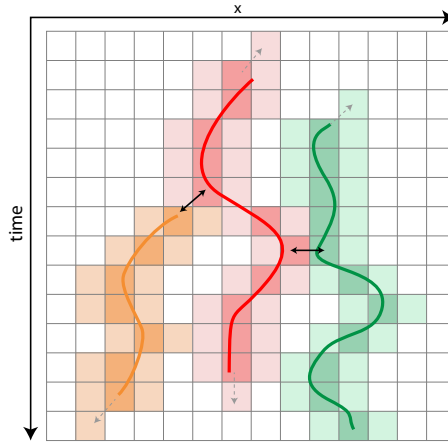


Figure 7.3: Horizontal cut of a video sequence. Estimated trajectories with sub-pixel accuracy are shown with red, green and orange curves. Quantized trajectories correspond to voxels filled with dark colors whereas adjacent voxels are indicated in light colors. Adjacency relations are represented by two-way arrows.

way to tackle these situations can be by measuring the gradient magnitude in both the horizontal and vertical flows:

$$\rho_{mb}(\mathbf{p}) = \exp\left(-\frac{|\nabla u(\mathbf{p})|^2 + |\nabla v(\mathbf{p})|^2}{0.01|\mathbf{w}(\mathbf{p})|^2 + 0.002}\right) \quad (7.3)$$

Where $\nabla u(\mathbf{p}) = \left(\frac{\partial u}{\partial x}, \frac{\partial u}{\partial y}\right)(\mathbf{p})$ is the gradient operation on the horizontal flow component $u(\mathbf{p})$. The gradient on $v(\mathbf{p})$ is defined similarly. If any of ρ_{sr} , ρ_o or ρ_{mb} falls below a given threshold, the tracked trajectory stops. The threshold values used here are the same as in (Sundaram, Brox, et al. 2010). The motion estimation algorithm (Brox and Malik 2011) provides sub-pixel accuracy on flow values so bilinear interpolation is used to track points where the flow falls in-between pixels. The trajectory can be expressed as a sequence of points $P = \{(x_t, y_t, t), \dots, (x_{t+l-1}, y_{t+l-1}, t + l - 1)\}$. Once the complete trajectory is computed with sub-pixel accuracy, each point location is quantized to the closest spatial integer position for each frame: $P_Q = \text{round}(P)$, see Fig. 7.3 for examples of quantization. We found very important to perform the whole tracking process with sub-pixel accuracy prior to quantization, specially in scenes with small displacements. In average, around 10% of voxels belongs to a trajectory of length higher than 2. Examples of points belonging to trajectories can be seen in Fig. 7.4.

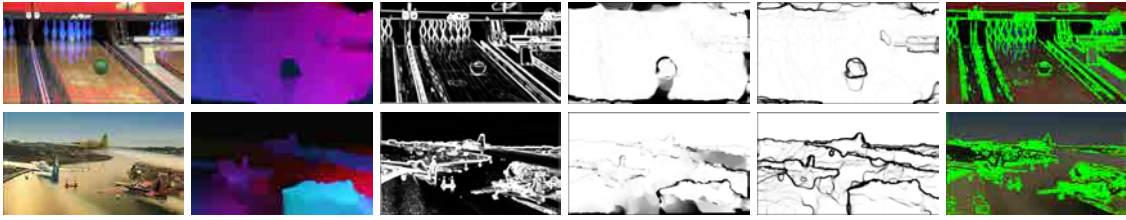


Figure 7.4: Example of flow reliability computation for two frames on two sequences of the BDS dataset (Sundberg et al. 2011). For each column, left to right: reference frame, forward motion frame where intensity and color indicate flow magnitude and direction respectively; structure, occlusion and variation reliability, with white values indicating high reliability. The last column shows points (in green) where a trajectory of length ≥ 2 is present.

7.2.4 Trajectory Binary Partition Tree

For every bottom-up (merging) segmentation approach, an initial partition should be provided. In image segmentation it is common to use either the initial partition the pixels (Felzenszwalb and Huttenlocher 2004; Vilaplana et al. 2008) or an oversegmented (supervoxel) partition (Arbeláez et al. 2011). In video, rather than supervoxels, the state of the art approaches is to begin with the 3D-partition defined by individual voxels (Grundmann et al. 2010). In this thesis, we adopt an hybrid approach between voxels and supervoxels, considering the initial partition as the quantized trajectories defined as described in Sec. 7.2.3. Trajectories allow to both introduce long term information and reduce the computational overload by reducing the number of starting regions.

The regions forming the initial partition are the trajectories as well as the non-tracked points which are considered trajectories of length 1 in the sequel. Then, a region adjacency graph is created by considering 4-connectivity intra-frame and 2 connectivity inter-frame: two trajectories are adjacent if they are neighbors in the same frame or their forward or backward motion endpoints coincide, see Fig. 7.3 for examples.

As a classic merging segmentation approach, the TBPT follows a greedy strategy to create the region hierarchy. It is constructed by iteratively merging the two most similar adjacent trajectories until only one region R_N is left. As adjacent trajectories are grouped together, they form what can be called *trajectory regions*. To decide whether or not two trajectories should be merged at a given iteration, internal characteristics of these regions should be used to differentiate them. In video streams, the two most important cues to do so are color (as for image segmentation) and motion. In the pro-

posed approach, a color model and a motion model are designed for each region. At each merging step, the models are used to define a region distance to determine the two most similar regions to merge. When the merging occurs, a new trajectory region is created, its color and motion models are updated and new distances are computed. The region models and distances are discussed in detail in the next section.

The iterative approach allows to define a more precise way to construct a hierarchy of partitions for videos than (Grundmann et al. 2010; Corso et al. 2008), where the partition coarseness of each level on the hierarchy is defined by a user defined parameter. The algorithm proceeds as in the BPT for single images and frames, iterating until one region R_N representing the whole video is left. As exposed in Sec. 5.3, the tree can be seen as a ‘container’ of many more partitions than the ones formed during the merging sequence. Using the tree cuts, it is possible to process the tree as in single images to obtain partitions of better quality.

7.2.4.1 Trajectory Region Model

Whether a given segmentation algorithm works with trajectories or other kind of regions, there are many ways to model the partitions elements and to define distance between these elements, such as the proposed BPTs for images in Sec. 5.2 or in Sec. 6.2. Motion segmentation algorithms dealing with trajectories such as (Brox and Malik 2010; S. Rao et al. 2010) only use motion information to define similarity between elements, while other systems such as (Paris and Durand 2007) (implementation by (C. Xu and Corso 2012)) rely only on region color characteristics. We adopt here an hybrid approach as in (Grundmann et al. 2010; Lezama et al. 2011), noting that color is the most discriminative cue for segmentation and motion allows to introduce dynamic information to the process.

Trajectories produced by (Brox and Malik 2010) can be as long as the entire video sequence if no occlusion occurs. This provides a very stable starting point for the merging process. However, prior to define the color and motion model, it is important to differentiate between spatial and temporal diversity:

- Objects tend to involve rich color distributions that are stable over time.
- Object motion tends to be spatially simple (uniform translation, rotation or zoom for example), but changing over time.

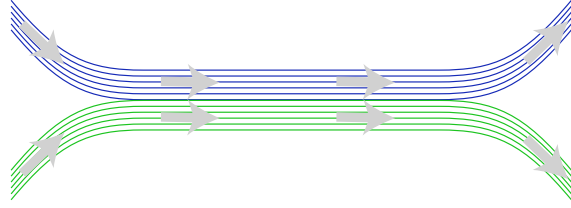


Figure 7.5: Importance of modeling the temporal evolution of trajectories. Even if objects share the same motion in some frames, they can be identified as different entities when motion is different at a time instant.

Therefore, color presents high spatial but low temporal diversity, while motion characteristics are the opposite. This encourages the use of different models for color and motion.

Color Model Color stability over time is, in fact, an assumption made by the optical flow estimation algorithm. Therefore, it is reasonable to assume that an image region can be represented with few colors regardless of its temporal span. Therefore we consider the trajectory region color model to be an adaptive histogram (signature) described by at most $n = 8$ dominant colors in the *CIE – Lab* color space. The signature of a region R is a set of pairs $s_R = \{(p_1^R, c_1^R), \dots, (p_i^R, c_i^R)\}, i \leq n$, where c_i^R is a representative color and $0 < p_i^R \leq 1$ its corresponding percentage of occurrence. This representation the same color representation as for images and frames, and for more details the reader is referred to Sec. 5.2 and Sec. 6.2.

Motion Model Object motion can be easily described between two consecutive frames. Typically, motion between frames is composed of piecewise-smooth regions. However, in spite of this spatial simplicity, object motion can change over time (unlike color). Therefore, the most important role of the motion model is to capture the different motions across frames and to preserve the order in which they appear. Fig. 7.5 illustrates the importance of modeling the temporal evolution of motion and therefore why models based on motion histogram should be avoided.

Therefore, the motion of each trajectory region R is represented by a set of motion vectors $m_R = \{\hat{w}_t^R, \hat{w}_{t+1}^R, \dots, \hat{w}_{t+l-1}^R\}$ where \hat{w}_t^R is the mean motion vector of the trajectory region at a given time instant t . Although the mean motion can be sufficient for oversegmentations, representing body motions with their mean can be somewhat limiting. For instance, motions such as zoom or rotation cannot be represented by a

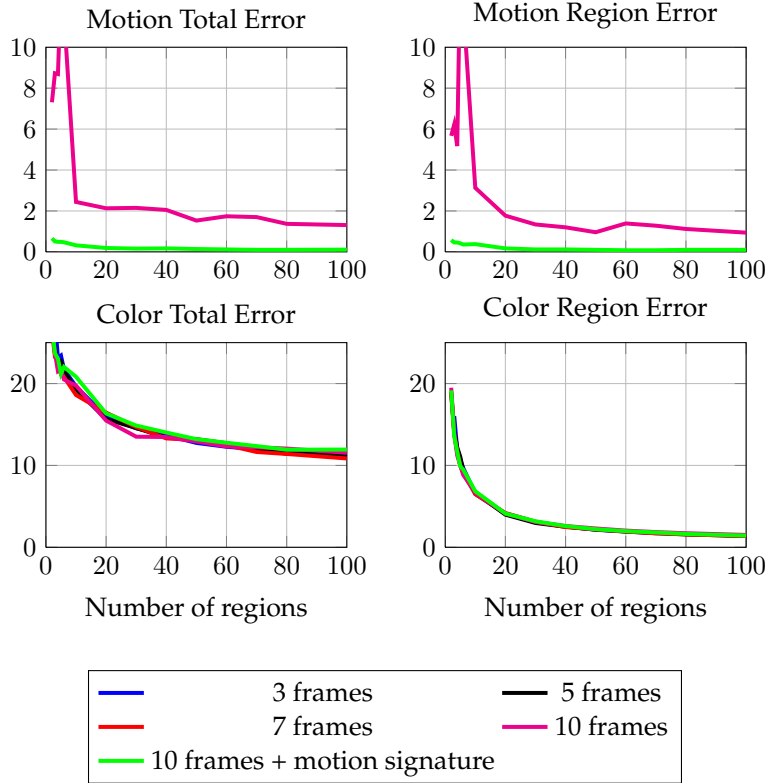


Figure 7.6: Figure showing the validity of the model for different cases of video length. The green curve shows the errors for the extended model.

single value. To cope with these cases, instead of the mean, an adaptive histogram with $n = 8$ is used at each frame when there is a small number of regions left to create the complete tree (200 is used in this work).

Validity of the model To prove that the color and motion models are scalable to videos of arbitrary length, short sequences of lengths 3,5,7 and 10 frames of the MOSEG dataset (Brox and Malik 2010) are segmented. At each merging step, the color and the motion errors for each pixel $\mathbf{p} \in R_i$ are computed:

$$CE(\mathbf{p}) = |\pi(I(\mathbf{p})) - I(\mathbf{p})| \quad (7.4)$$

$$ME(\mathbf{p}) = \left| \frac{\hat{\mathbf{w}}_t^{R_i} - \mathbf{w}(\mathbf{p})}{\mathbf{w}(\mathbf{p})} \right| \quad (7.5)$$

where $\pi(c)$ assigns the color c to their closest color cluster center in the region where \mathbf{p} belongs. CE and ME are averaged across pixels or averaged with the number of regions at each merging step. Results of the color and motion errors are shown in

Fig. 7.6, proving that color and motion errors behave as expected. Obviously, as the number of regions of the tree decreases, color and motion models exhibit higher error figures. There are no significant differences on the color error (bottom row) regardless of the different video length, confirming that 1) color is maintained within a region across time 2) small color variations are handled by allowing adaptive histograms. Since motion models already take the video length into account, only the 10 frames case is shown. Note that when extending the model, if the number number of colors in the histogram are doubled to 16 (green curve), the color error does not decrease. However, implementing a motion histogram instead of the mean allows to represent much more accurately the motion in regions.

7.2.4.2 Trajectory Region Distance

The merging sequence of the BPT is defined by a similarity measure between neighboring trajectory regions based on several distance notions.

Color Distance The distance used is the same as in single images and frames, see Sec. 5.2 and Sec. 6.2. The chosen distance is EMD, using also the same parameters as in the previous sections:

$$d_c(s_1, s_2) = EMD(s_1, s_2) \quad (7.6)$$

Motion Distance Even if two objects share the same motion during a long period of time, as soon as they move differently, they can be assigned to two different entities (Brox and Malik 2010). A good example can be found in Fig. 7.5, where two different objects meet at a given instant, then move together for some time and finally they split. In this example, it is clear that the motion distance should not consider a global motion model, but rather motion differences at each time instant. Similar to (Brox and Malik 2010), two adjacent trajectories are as different as their maximum motion difference at a given time instant:

$$d_m(m_1, m_2) = \max_{t \in T} 1 - \exp\left(-\frac{\rho_t \|\hat{\mathbf{w}}_t^1 - \hat{\mathbf{w}}_t^2\|}{\gamma_m}\right) \quad (7.7)$$

where T is the common period of time of both trajectories. The coefficient $\gamma_m = 4$ acts similarly to γ_c in Eq.(5.8), defining a soft threshold. Unlike color, which has bounded values for each channel, motion magnitude is very sequence dependent. However, we found that a displacement difference of four pixels is sufficient to the human eye. An

important factor in Eq.(7.7) is ρ_t which measures the intra-frame flow reliability:

$$\rho_t = \min_{\substack{i=1,2 \\ q=s,v,mb}} \widehat{\rho}_q^i(t) \quad (7.8)$$

For each frame, ρ_t is set to the minimum of the three reliabilities (structure, occlusion and motion boundary) of the two trajectories $i = 1, 2$ at each frame. At the last merging steps of the BPT, trajectory regions may be composed of many pixels of the same frame. Therefore, for each trajectory, the mean value of the structure $\widehat{\rho}_s^i(t)$, occlusion $\widehat{\rho}_o^i(t)$, and motion boundary $\widehat{\rho}_{mb}^i(t)$ reliability is computed. By introducing motion reliability to Eq.(7.7) we make sure that strong dissimilarities $d_m \approx 1$ only occur when both flows are sufficiently reliable $\rho_t \approx 1$. If $\rho_t \approx 0$, motion difference becomes irrelevant $d_m \approx 0$, as the two estimated motions may have arbitrary, possibly non real, values.

Final trajectory region distance Although color and motion are two key characteristics, the region size can help to reduce noise effects. We use a size factor $d_v(v_1, v_2)$ that encourages the merging of regions of small size over regions of larger volume:

$$d_v(v_1, v_2) = \log(1 + \min(v_1, v_2)/\gamma_v) \quad (7.9)$$

where v_1 and v_2 are the volumes of the two trajectory regions in voxels. γ_v acts similarly as γ_c, γ_m and it is set to 5% of the video volume. This factor prevents smaller regions to be considered of equal importance as the bigger ones. The final region distance is:

$$d = (1 - (1 - d_c)(1 - d_m)) d_v \quad (7.10)$$

where notation has been simplified for clarity purposes. d is close to zero when both color and motion are very similar, while it is close to d_v if either d_c or d_m are close to one. There are other forms of combination for region distance (Vilaplana et al. 2008; Calderero and Marques 2010), but Eq. (7.10) allows to mitigate the effects of arbitrary and non-reliable flows. That is, when $\rho_t \approx 0$, motion distance $d_m \approx 0$ and region distance is mainly governed by color, not by non-reliable motions. This effect is specially important in the early steps of the TBPT, where possibly occluded regions containing arbitrary flows can be compared.

7.2.5 Results on Early Segmentations

There exists strong controversy on whether one should evaluate the performance of the segmentation itself or evaluate it as a part of an application (Unnikrishnan et al.

2007). This is mainly to the fact that segmentation is an ill-posed problem, and more than one segmentation can be suitable for an algorithm purposes. Even humans, when faced to the same input produce different results. This fact encouraged the creation of the BSDS Dataset (Arbeláez et al. 2011) on image segmentation, where performance of an algorithm is evaluated against different human segmentations on the same image. On the video field, human annotations are much more costly, as a single sequence may contain more frames than images in the whole BSDS dataset. In this direction, the work of (A. Y. C. Chen and Corso 2010) allows to propagate region labels throughout sequences, helping to easily extend human groundtruth to full sequences.

In this section we restrict the evaluation of segmentations produced by the merging sequence, see Sec. 7.2.4. We analyze the quality of partitions involving between 900 and 100 regions, which correspond to strong and moderate oversegmentation respectively. Note that, one of the advantages of the TBPT is that using a binary tree and following the merging sequence, we have an exact control on the desired number of regions unlike methods like GBH or Meanshift.

In this section we are evaluating the quality of the (over) segmentations produced by the algorithm. Since at this point no depth ordering is available yet, we use a common and public evaluation method proposed in (C. Xu and Corso 2012) with the dataset from *xiph.org* used in (A. Y. C. Chen and Corso 2010) composed of 8 sequences of approximately 80 frames each. Each frame has a semantic ground-truth segmentation leading to a total of 639 annotated frames. The evaluation metrics are the ones discussed in (C. Xu and Corso 2012): Undersegmentation Error (UE), Boundary Recall (BR), Segmentation Accuracy (SA) and Explained Variation (EV), although a more formal definition of these measures can be found in (C. Xu and Corso 2012), an explanation for each follows. The four measures can be applied either in the 3D volume or 2D partitions but here only the 3D version is formally defined. The extrapolation to the 2D domain can be done with ease. Let's assume that the given video volume has been partitioned into a set of K segments $\{S\}$, with S_i being a particular region. The groundtruth annotation of the same video is composed by a set $\{G\}$ with G_i being groundtruth region. Undersegmentation Error measures what fraction of voxels exceeds the volume boundary of the ground-truth region. Its formal expression for a given G_i is:

$$UE(G_i) = \frac{\left(\sum_{S_j, S_j \cap G_i \neq \emptyset} |S_j|\right) - |G_i|}{|G_i|} \quad (7.11)$$

where $|\cdot|$ denotes region volume. The final UE is the average of $UE(G_i)$ across all G_i . The Boundary Recall (BR) assesses the quality of the spatiotemporal boundary detec-

tion by measures the quantity of groundtruth boundaries generated by $\{G\}$ captured (either spatial or temporal) by the segments $\{S\}$. There is no easy way to express this concept with a closed formula, but the idea can be easily interpreted. The Segmentation Accuracy (SA) quantifies what fraction of ground-truth segments is correctly matched. Suppose that each S_i is matched to a groundtruth segment \tilde{G}_i such that it maximized the Jaccard index:

$$\tilde{G}_i = \arg \max_{G_j} \frac{S_i \cap G_j}{S_i \cup G_j} \quad (7.12)$$

then, the SA measure can be expressed as:

$$SA(G_i) = \frac{\sum_{j=1}^K |\tilde{G}_j \cap G_i|}{|G_i|} \quad (7.13)$$

as in the UE case, the final SA measure is the average accross all groundtruth segments. Finally, Explained Variation (EV) is a measure assessing spatio-temporal uniformity and it is proposed in (Moore et al. 2008) as a human independent metric. The idea behind it is to measure how well region variance correlate with video color variance:

$$EV = \frac{\sum_{\mathbf{p}} |\boldsymbol{\mu}_{\mathbf{p}} - \boldsymbol{\mu}|^2}{\sum_{\mathbf{p}} |I(\mathbf{p}) - \boldsymbol{\mu}|^2} \quad (7.14)$$

where the summations are done over all the voxels \mathbf{p} . $\boldsymbol{\mu}(\mathbf{p})$ is the mean color of the region where \mathbf{p} belongs and $\boldsymbol{\mu}$ is the overall video color mean. EV is not a perfect metric in the sense that it penalizes region containing high textures, but it provides a metric independent from the groundtruth annotations.

Results of segmentation measures are shown in Fig. 7.7. For UE, BR and SA both the corresponding 2D and 3D versions are shown. Although being similar, 3D boundaries are biased with the amount of motion in the scene. With big displacements, the amount of temporal boundaries created is high compared to spatial (2D) ones. The temporal stability of segmentation is shown through mean duration of trajectory regions. It can be observed that the Trajectory BPT approach, while maintaining a competitive UE and UA, clearly outperforms the other methods in BR and EV. This means that 1) boundaries are very well preserved, achieving recalls above 0.8 and 2) produced voxels are more uniform in color statistics according to EV. This is specially difficult in complex scenes involving a lot of details and small regions. We believe that this difference in BR is mainly due to the introduction of the flow reliability into the region similarity. The average duration of the resulting trajectory regions can also be seen

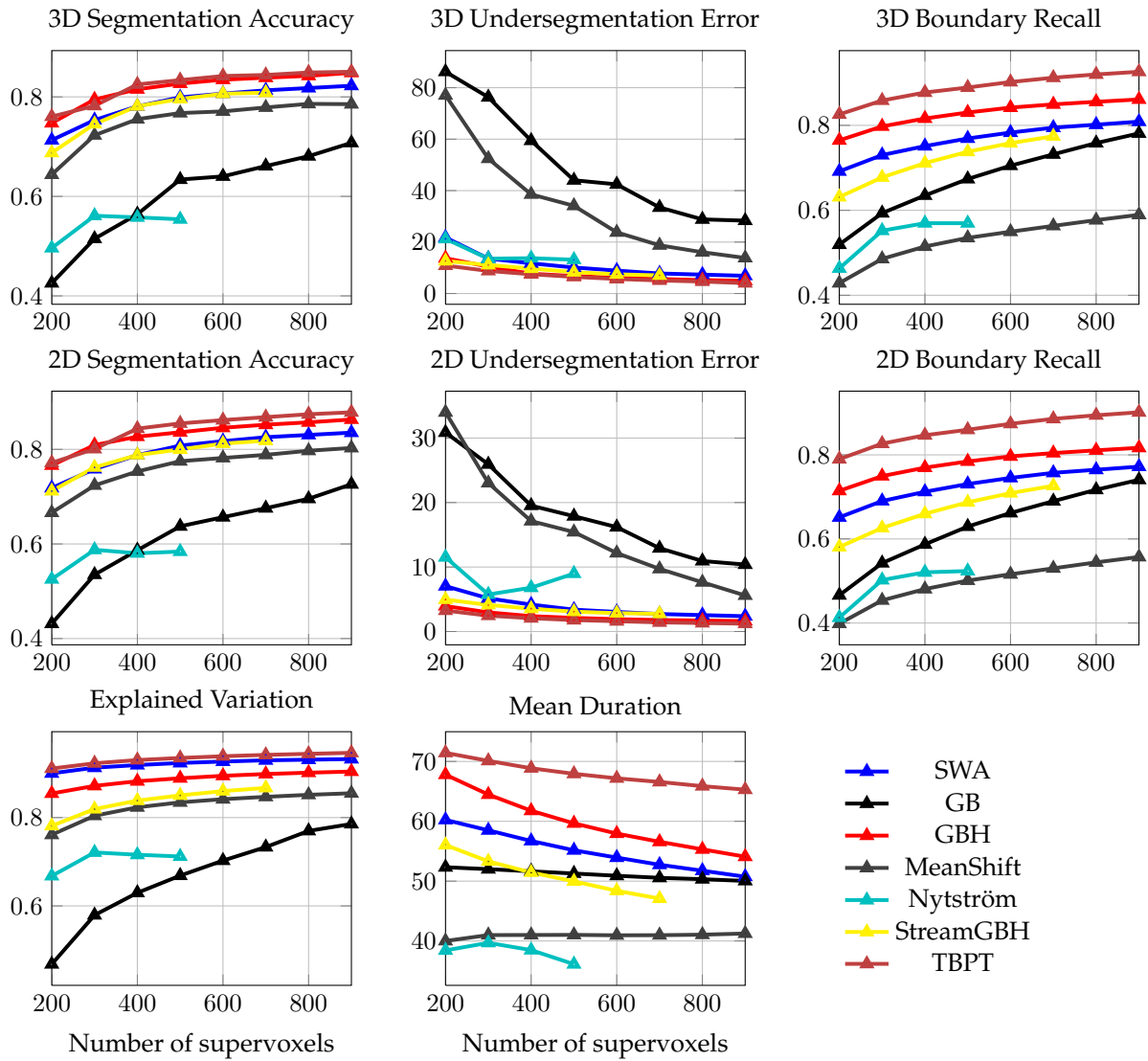


Figure 7.7: Results on the dataset of (A. Y. C. Chen and Corso 2010). From left to right and top to bottom: Segmentation Accuracy (SA), Undersegmentation Error (UE), Boundary Recall (BR) and Mean Duration versus the region number. The proposed system is among the best ones in terms of SA and UE and the best in BR. The Trajectory BPT creates regions spanning longer temporal intervals than other state of the art methods.



Figure 7.8: Video segmentation examples. Frames 1,11,21,31 from the bus sequence in the dataset (A. Y. C. Chen and Corso 2010). In row order, partitions obtained with the SWA method, the GBH algorithm and the proposed Trajectory BPT. Each region is colored with a unique color that is consistent over time. Each partition involves roughly 100 regions.

in Fig. 7.7 for different number of regions. The introduction of trajectories into the segmentation process has allowed the creation of temporally stable regions spanning throughout longer time intervals than other methods. As we shall see in the following sections, error figures in Fig. 7.7 represent only a lower bound on the proposed measures and, therefore, better partitions can be found.

For subjective evaluation, we show the partitions for three methods in Fig. 7.8. The sequence is particularly challenging as it involves small details and severe occlusions. State of the art algorithms have difficulties, but the Trajectory BPT algorithm is able to preserve boundaries such as the front fence. Although horizontal motion is dominant in the sequence, the TBPT is also able to track thin vertical structures.

To see how the algorithm behaves as the hierarchy progresses, Fig. 7.9 shows results on two sequences from the dataset used in (Sundberg et al. 2011). The airplane sequence is specially challenging because many areas have similar and homogeneous colors. As can be noticed, at finer levels of the hierarchy, boundaries are still well preserved. At coarser levels, regions with different semantic may be merged. In the bowling sequence, the color contrast is higher, but difficult challenges arise because of big displacements, appearing objects and specular reflections. The Trajectory BPT is able to track most of the objects of the scene and the produced regions have semantic homogeneity.

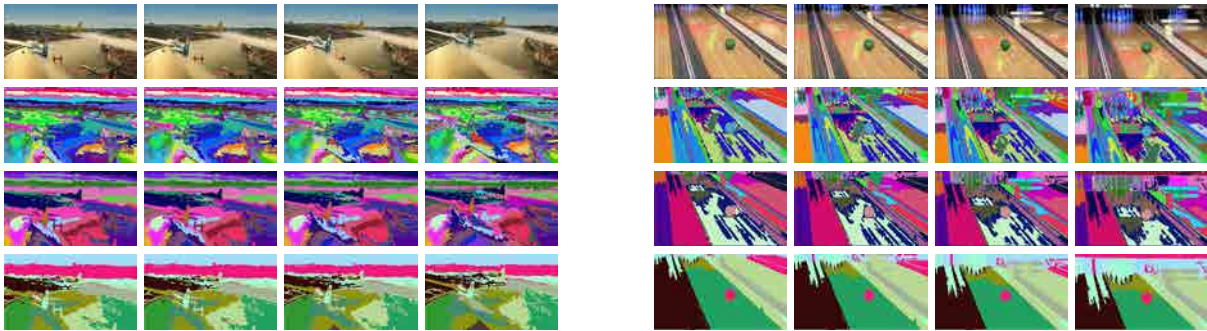


Figure 7.9: Two examples of the segmentation hierarchy. For each block, the first row contains frames 1, 5, 10 and 15. The rows 2, 3 and 4 show segmentations with 100, 40 and 10 segments respectively. A segment is uniquely colored across frames.

Computational cost The CPU time is governed by the complexity of the Trajectory BPT priority queue used to handle the distance values. Its complexity is $O(E \log E)$ where E is the number of edges between regions. Consumed memory is dominated by the storage of color and motion models for each region. Since region adjacency is sparse, the number of edges E can be considered proportional to the number of regions N . Therefore, the overall algorithm complexity is $O(N \log N)$ in time and $O(N)$ in memory. Overall, the algorithm is able to process sequences of 3 million voxels in around 1000 seconds using less than 20GB of memory in a CPU.

7.3 Relative Depth Ordering

Once a hierarchical representation for the whole sequence is constructed using the TBPT, the depth ordering process can proceed as in Sec. 6.3. The fact that regions in the TBPT have an additional dimension does not affect the algorithm. Nevertheless, slight changes are introduced in the occlusion estimation process and also in the costs used to cut the tree. Since a segmentation is available in every considered frame, unlike in the previous case in Sec 6.3, where the tree was only built for the central frame, it is possible to extend the occlusion estimation to several frames and it is also possible to evaluate the performance of the algorithm using different window lengths. In the following sections all particularizations for the video case with respect to the single frame case are exposed.

7.3.1 Tree Cut for Motion Segmentation

When pruning the tree constructed for frames, a parametric flow model was fitted to each region on a single frame and the tree cut minimizing a sort of distortion error was found. In the video case, it is straightforward to extend the cost to include the motion distortion to several frames. Since a tree cut attempts to find the region best fitting to a certain model, it is possible to see the cut as a motion segmentation step, where moving objects are supposed to be found. There are many approaches that tackle motion segmentation in the literature. Two of the most common techniques are either spectral clustering (Brox and Malik 2010) or low-rank factorization methods (S. R. Rao et al. 2008). In this context, we propose a different approach using tree cuts and parametric motion models. Defining the energy for a region R_i as e_i following the tree cuts energy (5.15):

$$e_i = \sum_t \sum_{q=t\pm 1} \sum_{\mathbf{p} \in R_i} |\mathbf{w}^{t,q}(\mathbf{p}) - \tilde{\mathbf{w}}_{R_i}^{t,q}(\mathbf{p})| + \lambda \quad (7.15)$$

where the summation is done over all time instants t where R_i is present. $\mathbf{w}^{t,q}(\mathbf{p}) = (u^{t,q}(\mathbf{p}), v^{t,q}(\mathbf{p}))$ is the motion field from frame t to frame q in a pixel $\mathbf{p} = (x, y)$ of frame t . λ is a constant penalty term in each region which controls oversegmentation. The 8-parameter flow model is defined as in Eq. (3.37) and is composed 8-parameters that define a planar surface under rigid motion.

The flow models correspond to motions subject to planar surfaces and projective cameras. Although real objects are not flat, if their distance to the camera is relatively high, they can be considered as such. Results of the tree cuts with respect to the merging sequence partitions and some state of the art on motion segmentation are shown in Fig. 7.10 for the MOSEG dataset (Brox and Malik 2010)² with 10 frames. The improvement of tree cut with respect to the merging sequence is clear on the pixel error where, for the same degree of oversegmentation, the tree-cut algorithm finds partitions with less average error than its corresponding merging sequence partition. On the region error the improvement is not so clear, due to the fact that missed objects contribute to high error penalties. Therefore, missing an object has a more severe impact on the performance than being more accurate on the segmentation. Since missed objects are normally small, they are normally missed in the tree creation process. Hence, both performances TBPT-ms and TBPT-tc are similar because even with the tree cut, these

²The matching process of (Brox and Malik 2010) assigned segments based on their common overlap. In case of subsegmentations, small objects could be missed so we changed the algorithm to use the Jaccard index (intersection over union) instead of the union.

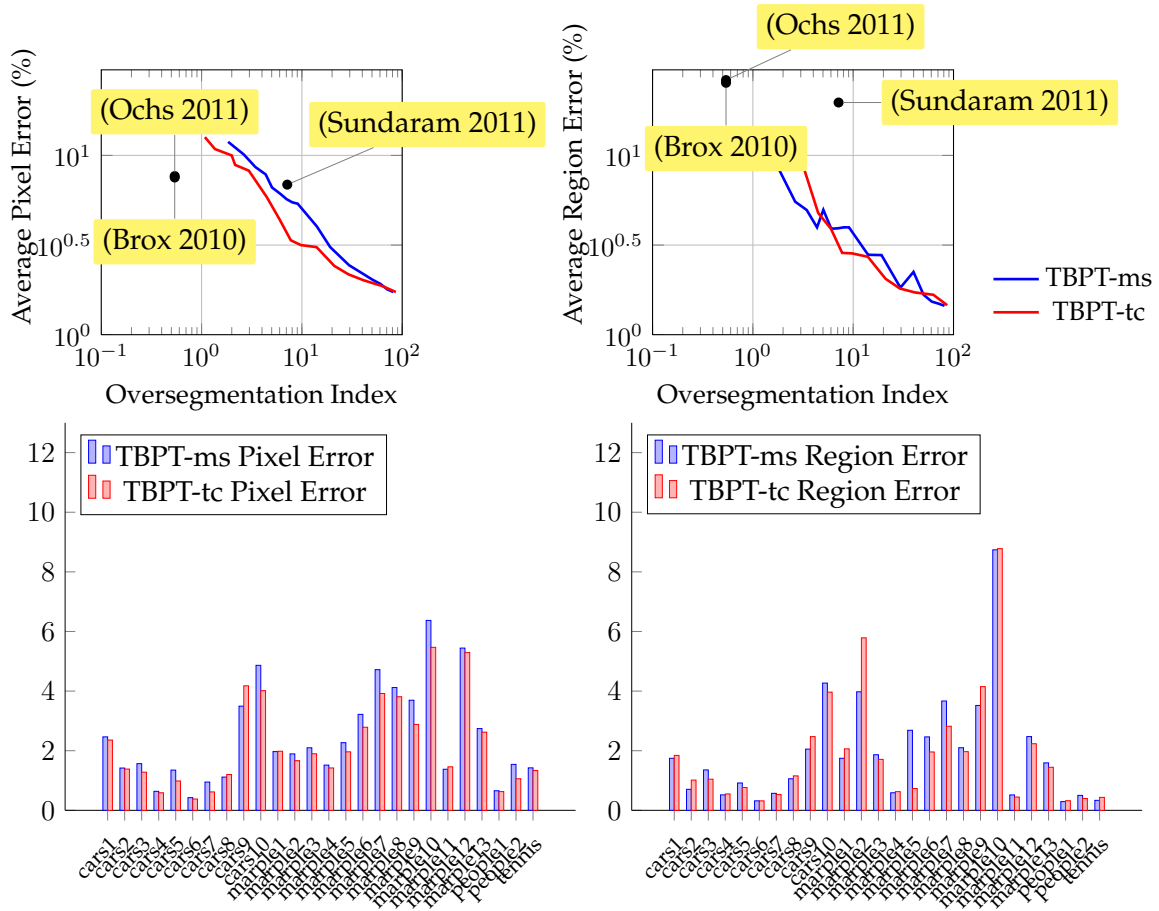


Figure 7.10: Motion segmentation performance evaluation. Top: Average pixel (left) and region (right) error for the TBPT on the MOSEG dataset with 10 frames. The suffix ‘ms’ refers to partitions of the merging sequence, while ‘tc’ refers partitions obtained with optical flow based tree cuts. Bottom: error in each of 26 sequences composing the MOSEG dataset.

objects cannot be recovered. Although the pixel error for the TBPT is sometimes worse (Brox and Malik 2011; Ochs and Brox 2011), it offers a clear improvement on region error. Comparison with (Sundaram and Keutzer 2011) deserves a special attention as, like the TBPT, the produced result is a hierarchy of regions using ultrametric contour maps. Error figures show that the TBPT improves the UCM in both measures. Tree cuts show a better performance, and in the majority of cases, it recovers the salient objects with more accuracy than the merging sequence partitions.

When a segmentation is available, the occlusion relations can be estimated. Since a flow model is available for each frame, occlusion estimation can incorporate region information as explained in Sec. 3.2.

7.3.2 Occlusion relations for video

The method for incorporating region information into occlusion detection is showed in Eq. (3.35) in Sec. 3.2. Nevertheless, a short summary is given for the reader's clarity. The process is similar as with the single frame case in Sec. 6.3. With flow models $\tilde{w}^{a,b}$, $a, b = t, t + 1$ available, a pixel becomes occluded if:

$$\Lambda(\mathbf{p}) \neq \Lambda(\mathbf{p} + \tilde{w}^{t,t+1}(\mathbf{p}) + \tilde{w}^{t+1,t}(\mathbf{p} + \tilde{w}^{t,t+1}(\mathbf{p}))) \quad (7.16)$$

Where Λ is an operator which maps each pixel to a region label of the partition P_o . That is, a pixel is occluded only if their compensating flows end up in a different region than the original pixel. Performance of the occlusion detector (3.35) is shown in the corresponding Sec. 3.2. When occluded points are available, occluding points and relations can be established as the single frame case in Eq. 3.43, by forward-backward flow compensation.

7.3.3 Depth ordering

Prior to discussing the depth ordering algorithm for video sequences, it should be stated that depth ordering assumes that the order of the objects is constant throughout the sequence. That is, objects do not change their relative depth in the examined frames. This may be somewhat limiting, but for short video sequences this assumption normally fulfills.

The only difference between the video and the single frame approach is that occlusion relations are present in all the frames of the sequence and not only in the reference one. That is, whether or not the central frame has occlusions, if occlusions appear in some other frame it is possible to relate spatio-temporal regions in all the other parts of the video. This is clearly an advantage if small motions are present, as one frame may not contain sufficient information. This can also be a drawback depending on the kind of motion in the scene and the length of the video sequence. If objects change their depth ordering during the sequence, including occluding relations of all frames may introduce conflicting depth relations between objects.

The process is as follows. When a window of length $L = 2 * W + 1$ is segmented, there are many possibilities to handle motion occlusions. To assess the performance and the redundancy of motion occlusions with different video lengths, a variable window of length $O = 2 * P + 1$ centered at frame W of the segmentation window is considered. Occlusion relations occurring outside the window O are discarded. Therefore, for each input sequence and a reference frame there are two parameters of the system: The

length of the segmentation window L and the length of the occlusion window O . We will denote the methods as $\text{TBPT}(L,O)$.

The depth ordering algorithm is essentially the same as in single frames. A detailed explanation can be found in Sec. 6.3, but here the main ideas are reproduced. Once the parametric flows are found for the video using a first tree cut with energy Eq. (7.15), occlusion relation within the window of length O are estimated. These occlusions are then used for a second tree cut to obtain a new partition, the region of which are ordered according to their relative depth. The ordering is performed by constructing a depth order graph (DOG) using motion occlusions and inferring a global consistency eliminating possible conflicts, just as in Sec. 6.3.

7.3.4 Results

Evaluation of the system is a little more complex in the video case than in the single frame case because there are many factors coming into play. For instance, the length of the video sequence that should be segmented may be of crucial importance. For short sequences, there will be less motion information than for longer sequences (imagine a stationary objects that remains still all the sequence but not in some frames) although longer sequences may be more difficult to segment since region models for the TBPT may not adjust to the real data. Additionally, even if the window length L is changed, the set of occlusion relations considered may also be varied by changing O . Again, considering occlusion relations in short time lapses may not be sufficient to relate some moving regions if movement is too small. However, considering all relations may introduce conflicts which may be solved erroneously during the depth ordering process using the DOG.

To deal with these issues, the following experiments are designed: the LDC and GDC from Sec. 4.2 are evaluated varying either the length of the segmented video and the length of occlusions considered. That is, $L = 3, 5, 7, 9$ and $O = 1, 3, 5, 7, 9$ and in all cases $O \leq L$. The considered occlusions are always centered in frame $W = 0.5(L - 1)$. As with single frames, the BMDS dataset is used. Results are shown in Fig. 7.11 for LDC and the GDC measures.

By looking at the figure, several conclusions can be drawn. First, for each window size, the length of the occlusion window O does not seem to vary the system performance. This can be interpreted in two ways:

- The system is stable to the introduction of more cues, handling correctly the in-

7. DEPTH ORDERING OF VIDEO SEQUENCES

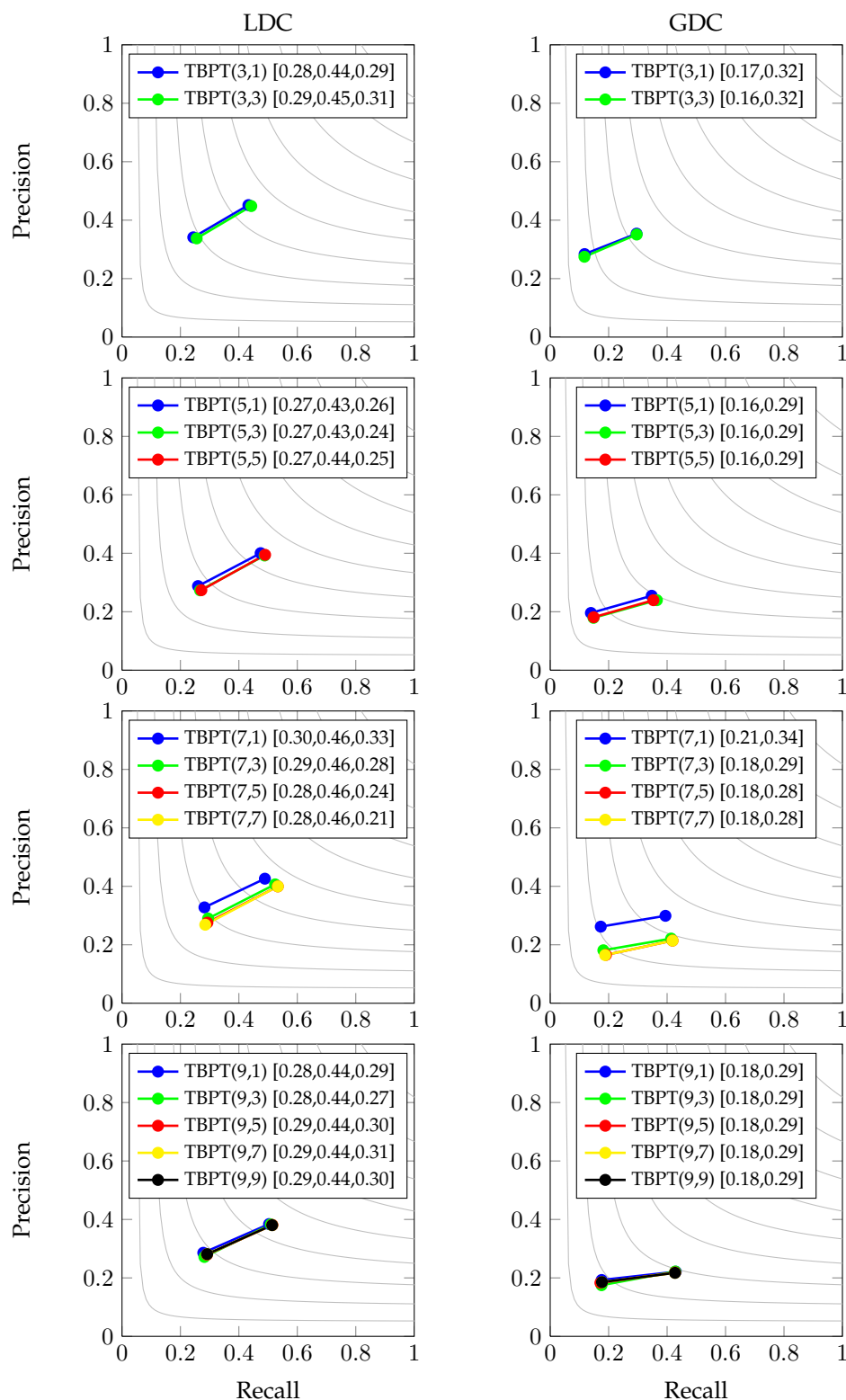


Figure 7.11: LDC and GDC results, left and right respectively, for a series of segmentation windows and different occlusion windows. Each approach is named TBPT(L,O) where L refers to the segmentation window and O refers to the occlusion window considered. Numbers between brackets indicate, in order, the classification score, the detection score and the ORI index

roduction of possible conflicting cues as O increases. When considering large O , it is unavoidable to introduce wrongly estimated occlusion relations. Anyway, the conflict resolution step in the DOG is able to maintain the same performance.

- Occlusions of one single frame are consistent with the occlusions of the rest of the sequence. That is, objects follow a coherent motion throughout the sequence, without changing depth in the analyzed sequence, confirming the initial assumptions on object movement and relative depth constancy. If objects changed depth, some occlusion relations would indicate opposite depth relations, making the task of depth ordering impossible. Of course, only short sequences are analyzed and these cases rarely occur. Nevertheless with longer sequences variable depth cases should have been considered.

Second, by looking across several windows length L , one can see that the detection score of all the system is more or less stable. In the LDC case, around $F = 0.45$, and in the GDC case around $F = 0.30$. Of course, following the trend of static images and single frames, the GDC score is lower than the LDC. Note that, considering the short video length, the segmentation performances are stable. This observation opens the door to design a streaming approach for the TBPT: as the quality of the tree stays approximately the same regardless of the segmented window length, a full video sequence could be divided and processed in small chunks without losing quality. Comparing the video case with the single frame case ($F = 0.44$ in the best case), but , the detection score is slightly better in the video case, showing that the extra information of the video signal helps to the segmentation process.

Moreover, the classification score stays also constant, around $F = 0.28$ for the LDC and $F = 0.18$ for the GDC measure, giving an ORI index of around $\text{ORI}=0.3$, much better than the best case in single frames ($\text{ORI}=0.25$). That is explained by the improvement of the occlusion detection process, where region information was introduced to make motion and color edges coincide spatially. Although these measures improve state of the art algorithms, performance is still far away from human perception.

To verify the behavior of the system, LDC and GDC results are also shown for the MOSEG dataset (Brox and Malik 2011) in Fig. 7.13. The chosen parameters are $L = 10$ and $O = 10$. This dataset is easier than BMDS in the sense that it is specially designed for motion segmentation, whether BMDS is designed for general video segmentation (using color, texture and other cues) and not only with motion cues. Therefore, in MOSEG there are always moving objects which can be clearly distinguished by their

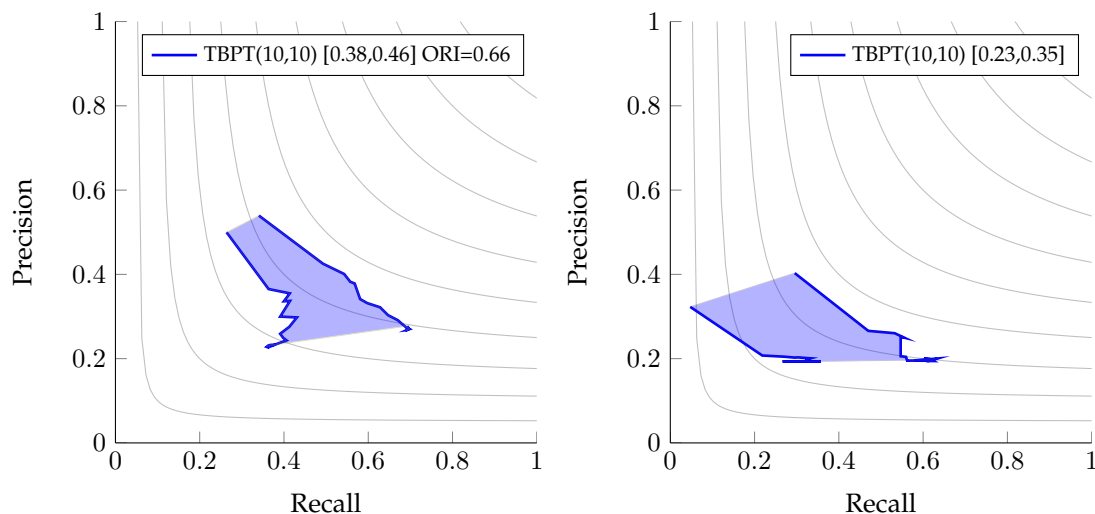


Figure 7.12: LDC and GDC results for the MOSEG dataset. The notation of the method TBPT(10,10) is the same as in Fig. 7.11

motion (regardless of their color), so it is expected that motion occlusions provide more reliable depth orderings. Indeed, the system presents an ORI index of 0.66, way over the ORI indexes for BMDS.

This high ORI index is explained by the kind of motions present in the dataset. Sequences are mainly comprised by moving cars in roads and people walking, so scenes present easier situations than in the BMDS sequences. In Fig. 7.13 some examples are shown, showing that motion occlusion indicate the correct ordering in most of the cases.

7.3.4.1 Qualitative results

As done in the previous two cases in images and frames, visual results are represented for several situations. This type of analysis helps to see in which situation the systems works or in which scenes the algorithm has difficulties. Fig. 7.14 shows a few examples of the BMDS sequences with varying illuminations, strong textures and different motion types. Results show that the system is able to cope with almost all situations, and the overall structure is seen in most of the cases. Of course, there are a few wrong estimated depth relations but in general the main depth relations are captured. Note that, different from single frames, the system orders by depth in the whole sequence rather than just in individual frames.



Figure 7.13: Examples of depth ordering in the MOSEG dataset. From left to right: reference frame, groundtruth depth ordering and results for first, sixth and tenth frame.

Although here only a few examples are shown, the general behavior for the system can be grasped for several situations. For example, in static scenes with a moving camera (rows 3 and 5) the algorithm still identifies the regions on the video which present a high change in depth, leaving smooth gradients untouched. In the third row, the buildings are extracted because they are occluding the sky, while the water is not identified with different depths, as its depth fades away and no occlusions are present. In other situations with moving objects, the algorithm is able to handle most cases, but there are some sequences in which the algorithm has difficulties:

Non-rigid motion Specially seen in sequences where there exist people walking, the

7. DEPTH ORDERING OF VIDEO SEQUENCES

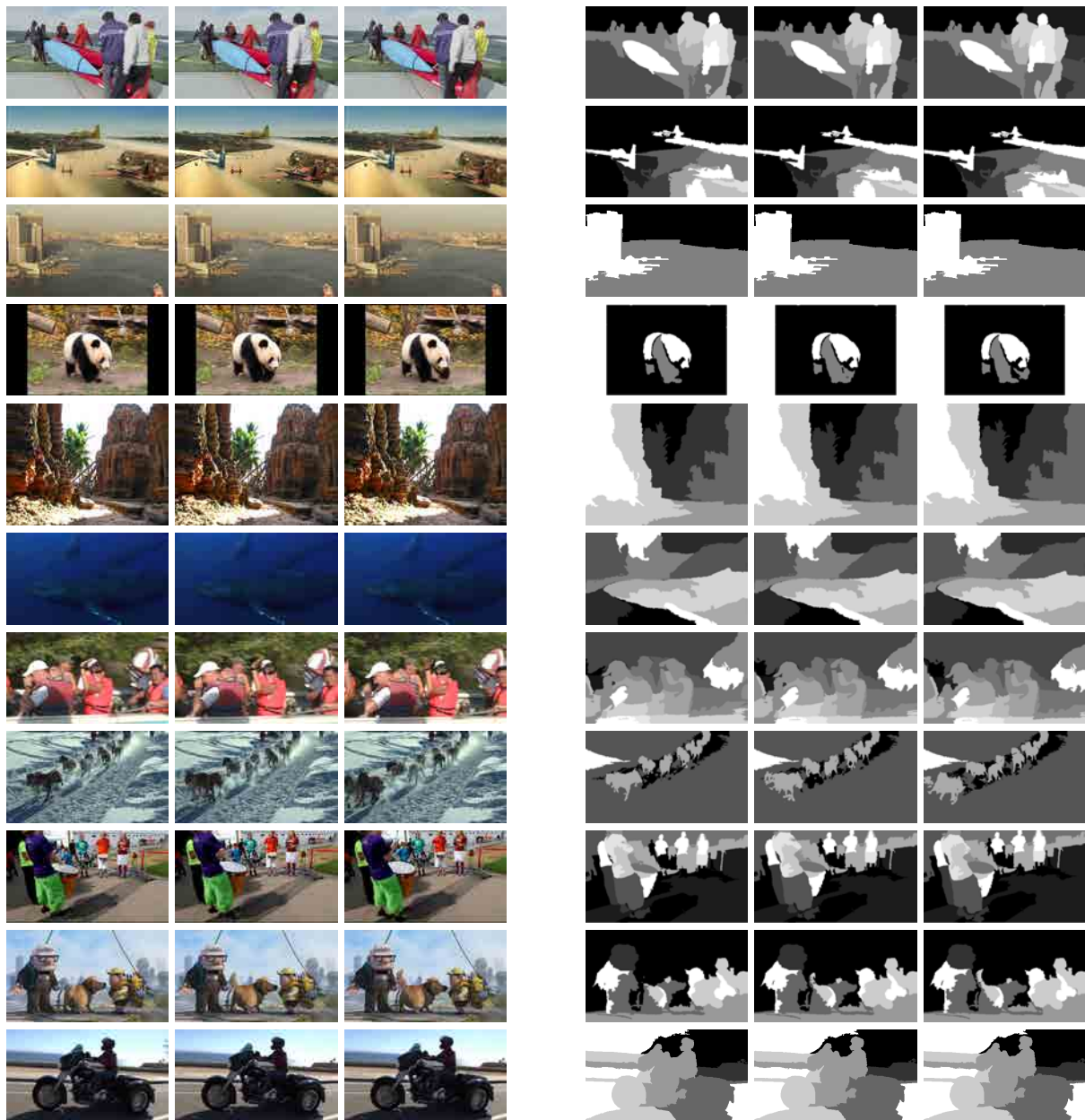


Figure 7.14: Series of results in the BMDS dataset. Results are shown using the TBPT(9,9) system, but only the three central frames are shown for clarity. Original images are on the left, and depth results are on the right. See how the system is able to handle many situation, but in sequences with high rotations (two last ones) some spurious and wrong orderings are obtained.

algorithm breaks objects into small parts (take for example rows 1,4,7 and 9) to capture self-occlusion. Specially arms and legs are broken from the body and assigned their own depth order. This may be undesirable in some systems, although it may also be seen as an advantage to have a more precise system.

Rotations Looking at the two last rows, when strong rotations are present (the dog tail in the next-to-last row and the wheels in the last row) spurious responses are created and, even, wrong depth relations are inferred.

Overall, the system can be considered to be well-behaved. Most of the situations are correctly handled and the main structure of the scene is correctly capture by the algorithm.

8 Structure from Motion

All along this thesis depth estimation was limited to infer the relative depth order of the objects in the scene. By using only low level cues it has been possible to determine local depth relations between regions in the image. By means of a globalization process local relations are extended to create global consistent relative depth map orders. Nevertheless, the system has been limited to obtain flat objects and relative depth, without absolute references. Precisely, in video sequences it is possible to estimate depth beyond relative depth if some suppositions are fulfilled. The first, is that there must exist some kind of apparent motion. Either the camera or the background should move with respect each other so that structure from motion can be inferred. The second is that this motion has to be rigid if, the algorithm complexity should be kept moderate. In Sec. 7.1 a review of algorithms retrieving depth maps from video is exposed. Several ways exist to tackle the depth estimation on sequences. Two of the most common approaches are 1) to describe the video using a depth ordered layered representation or 2) to apply structure from motion algorithms. As it would be seen in the open problems and future work, still much work needs to be done when estimating structure in videos with multiple moving objects. The purpose of this section is to give the first steps towards the design of a system that exploits optical flow to compute structure from motion of a scene, under the supposition that the scene is static.

The perspective of this chapter is somewhat different as the the approach followed in previous sections, where relative depth ordering between objects is estimated. Whereas estimating relative depth requires a segmentation step, structure from motion does not need region information to recover depth. Instead, using only pixel-based information, structure from motion algorithms are able to recover a dense depth map for a static scene. Segmentation techniques used throughout this thesis, namely binary partition trees, are not used in this part. As a future work, relative depth ordering using segmentation information and structure from motion will be combined to produce dense depth maps for arbitrary moving sequences. Nevertheless, since structure from motion can be used alone for static scenes, we present the algorithm and intermediate results in the following sections.

The proposed algorithm follows a similar strategy than many state of the art structure from motion algorithms, specially (Pollefeys et al. 2004). The difference of the proposed algorithm with respect to the others is that here only optical flow information is used. In this way, we obtain a much denser coverage of tracked points throughout the video sequence. This allows us to have more information to improve reliability

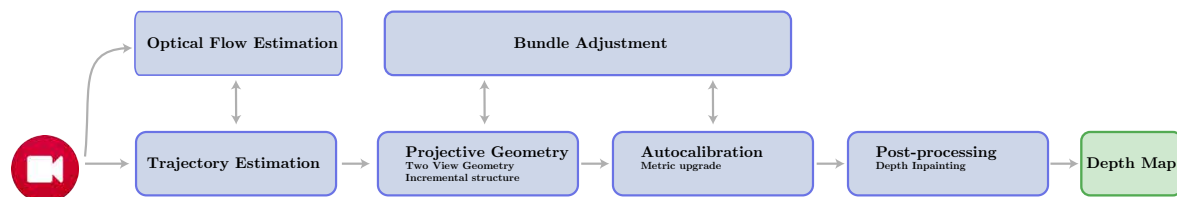


Figure 8.1: Block diagram of the system to retrieve structure from motion.

of the structure estimation and it also makes the transition from sparse depth map to dense coverage much easier. The following sections discuss the proposed approach to recover depth from videos recorded using only one point of view.

8.1 Structure from General Motion

In practical sequences recorded with hand held cameras, the point of view normally moves and rotate freely, producing some motion patterns which can be exploited to derive the depth structure. Additionally, if the scene is static and rigid, it is possible to recover directly the spatial structure by using only motion information.

The approach of the proposed system can be seen in Fig. 8.1 and a detailed explanation is provided in the following sections. The proposed system has many similarities with the work (R. Hartley and Zisserman 2004), although some improvements are made in many stages of the algorithm. This section is, to the moment of publication, ongoing work which would be used in combination with motion occlusions exposed in Chapter 7 so as to handle depth reconstructions for sequences with arbitrary movements and arbitrary object movement. In a nutshell, the system uses estimated trajectories from optical flow to find reliable point tracks. From the first two frames, the structure of the video is built incrementally, one frame at a time until all frames are processed. Since the camera calibrations not known to the system, an autocalibration step is needed to ensure a proper reconstruction. In the incremental structure computation and the autocalibration steps, non-linear refinement of the obtained solutions is performed by means of bundle adjustment. Since the estimated depth maps are sparse, a post processing step is needed to generate dense depth maps for each frame. Prior to the description of the system, a brief introduction to the pinhole camera model and homogeneous coordinates follows, as they are central for the reconstruction algorithm.

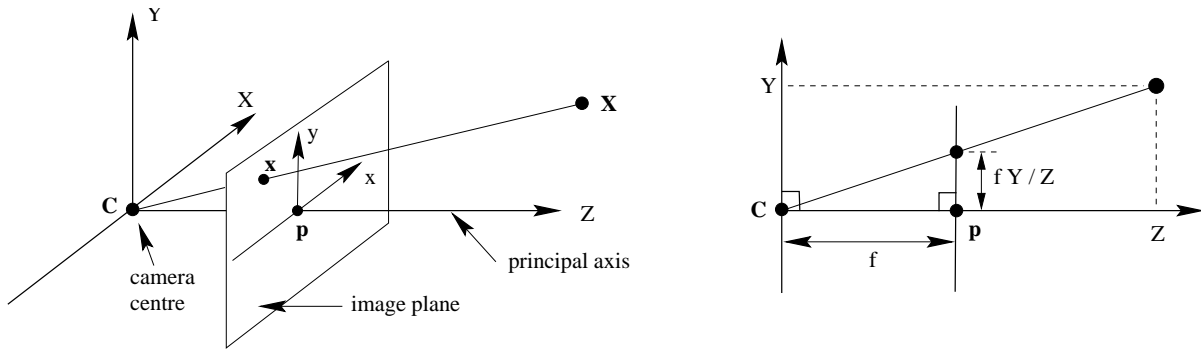


Figure 8.2: Pinhole camera geometry. Figure extracted from (R. Hartley and Zisserman 2004).

8.1.1 Pinhole Camera Model and Homogeneous coordinates

Depth reconstruction is considered to be the inverse problem of 3D to 2D projection. By having a series of images of the same object/scene, it is possible to retrieve the absolute position of each point in the object (up to a scale factor). Therefore, understanding how 3D points are projection onto the camera plane is a key aspect of the development of reconstruction algorithms. Suppose that the camera center is at the point $C = (0, 0, 0)$ and it is looking in the positive Z -direction. A point in space $X = (X, Y, Z)$ should be projected to a plane $Z = f$, where f is known as focal length. The expression of the projection is $x = (fX/Z, fY/Z)^T$, see Fig. 8.2.

The above expression can be expressed in matrix notation with the use of homogeneous coordinates. Homogeneous coordinates were introduced in (Möbius 1827) to allow a compact representation for infinite points. The point X in a three dimensional euclidean space corresponds to a point $\lambda(X, Y, Z, 1)$ in the projective space with $\lambda \neq 0$. Points at infinite are represented in the projective space with their last coordinate equal to zero, $\lambda(X, Y, Z, 0)$. This kind of representation is suitable to represent points at infinity with finite coordinates (unlike normal, or euclidean coordinates), and are very widely used in structure from motion algorithms. From now on, assume that point coordinates are expressed in homogeneous coordinates, whereas euclidean coordinates will be explicitly mentioned. The projection equation in matrix notation is:

$$\lambda x = \mathbf{P}X = \mathbf{K} [\mathbf{I}_3 | \mathbf{0}_3] X \quad (8.1)$$

$$\mathbf{K} = \begin{bmatrix} f & 0 & 0 \\ 0 & f & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8.2)$$

where \mathbf{I}_3 is the 3 by 3 identity matrix and $\mathbf{0}_3$ is a zero vector with three coordinates and λ is a non zero factor. In real situations, the camera is not at the world center $(0, 0, 0)$ but has some translation and rotation. Moreover, the camera calibration matrix \mathbf{K} may have varying focal lengths, projection centers and skew parameters. For arbitrary camera internal parameters, and arbitrary position \mathbf{C} and orientation \mathbf{R} , the projection of a point X becomes:

$$\lambda \mathbf{x} = \mathbf{P} \mathbf{X} = \mathbf{K} [\mathbf{R} | -\mathbf{R} \mathbf{C}] \mathbf{X} \quad (8.3)$$

$$\mathbf{K} = \begin{bmatrix} f_x & s & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad (8.4)$$

where \mathbf{R} is the rotation matrix of the camera, \mathbf{C} is the world camera center and (c_x, c_y) is the image projection center. The skew s is normally considered $s = 0$ and focal lengths f_x, f_y are considered equal in practical cases $f_x = f_y = f$. With Eq. (8.3) of the projection of a single point in one image it is possible to relate the projection of the same point to two (or more) different images.

8.1.1.1 Example Application: Motion Parallax

Motion parallax appears when the camera is moving relatively to an object, and this object/region presents large motion differences, generally when the observed objects occupy high ranges of depths. For example, if one travels by train and looks at the landscape, near objects appear to move faster than objects far away. To show the applications of the pinhole camera mode, the motion parallax case will be developed here, showing its equivalence to stereo/disparity estimation algorithms.

In this section it is assumed that the motion of the camera is restricted, with only a horizontal translation affecting the point of view. Assume that the camera is looking to a static scene and that it moves laterally, without rotating or zooming. With the pinhole camera model the projection of a 3D point \mathbf{X} to a point \mathbf{x} in the image plane is given by the projection matrix: $\lambda \mathbf{x} = \mathbf{P} \mathbf{X} = \mathbf{K} \mathbf{R} [\mathbf{I}_3 | -\mathbf{t}]$. The camera skew is $s = 0$, and the focal length f and the principal point in the image c_x, c_y are known. All these parameters are encoded as in Eq. (8.4).

The rotation matrix \mathbf{R} indicates the rotation with respect to the three axes x, y, z and for now let's assume that $\mathbf{R} = \mathbf{I}_3$. Additionally, if the camera center is at the origin of coordinates, $\mathbf{t} = (0, 0, 0)$, the projection of a 3D point to a 2D image point becomes

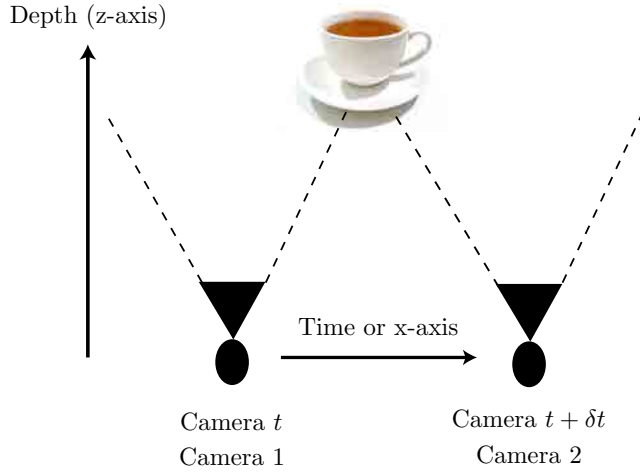


Figure 8.3: Motion parallax experiment setup and its equivalence to stereo problems. Two different cameras (either in different x-position or in different time instants) look at a static object in front of them.

$\lambda_1(x, y, 1) = (fX, fY, Z)$. Converting from homogeneous coordinates to euclidean, the projection becomes:

$$(x_1, y_1)^\top = \left(\frac{x}{\lambda}, \frac{y}{\lambda} \right)^\top = \left(\frac{X}{Z}, \frac{Y}{Z} \right)^\top \quad (8.5)$$

It is possible to see that the position of a point in the image is inversely proportional to its depth. Assume that the camera moves, or in the stereo case a second camera is placed with the same orientation than the first camera but with a displacement of d units only in the x-coordinates. The displacement vector becomes $\mathbf{t} = (d, 0, 0)$ and the projection of the same 3D point to the other image is:

$$\lambda \mathbf{x}_2 = \mathbf{P}_2 \mathbf{X} = \mathbf{K} [\mathbf{I}_3 | - (d, 0, 0)^\top] \mathbf{X} = \mathbf{K} (X - d, Y, Z)^\top \quad (8.6)$$

If the calibration matrix is known, it can be ignored, so the point can be expressed in euclidean coordinates as $x_2 = \left(\frac{X-d}{Z}, \frac{Y}{Z} \right)$. If $\delta x = -\frac{d}{Z}$, the point \mathbf{x}_2 can be expressed as $\mathbf{x}_2 = (x_1 + \delta x, y_1)$. That is, knowing that two images are taken by two cameras displaced by a pure horizontal translation, the horizontal displacement of a point in the two images is a direct indicator of the absolute depth.

In fact, the previous paragraph just proves mathematically the effects observed in motion parallax commented in Sec.2.2. Observing that the displacement $\delta x = \frac{d}{Z}$ is in-

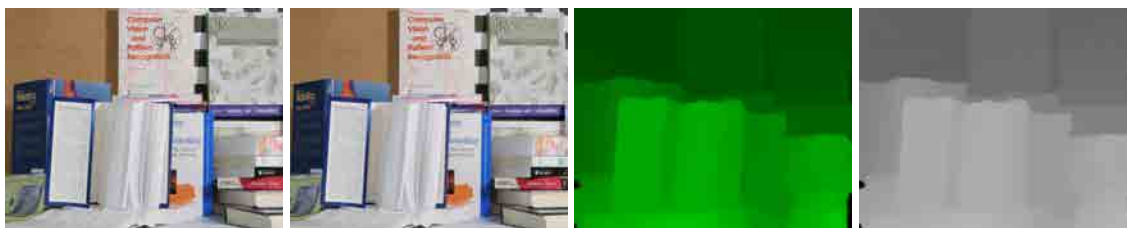


Figure 8.4: The two right images are the left and right original images respectively. The center right shows the estimated motion between images, with more saturation for larger motions. The right image shows the disparity image, where brighter areas correspond to closer regions.

versely proportional to the depth, near objects (small Z) will produce bigger displacement, while objects with big Z will seem rather stationary.

In Fig. 8.4 a clear example of this effect can be seen. In this case, disparity is directly estimated from optical flow. Nevertheless, efficient algorithms exist for the stereo/motion parallax case. Optical flow needs to estimate two variables per pixel, which are the displacements of the pixel from one image to another. Since in motion parallax/stereo only displacements in the x -direction appear, algorithms only need to estimate one variable per pixel. The general approach to estimate disparity is very similar to optical flow: using a variational approach relying on data and on smoothness constraints, see Eq. (3.22). Some works tried to compare both approaches, such as (Durgin et al. 1995) and (Miled et al. 2009) where disparity is closely related to motion. Literature on stereo disparity estimation is vast, and current state of the art techniques offer very accurate and fast approaches, see the surveys in (Barnard and Fischler 1982) and (Scharstein and Szeliski 2002) for a detailed literature enumeration.

8.1.2 Reliable trajectory tracking

After reviewing a particular application of the pinhole camera model, we are going to address the case of general motions. To relate the same point between images, the point should be tracked across frames from a video. Assume that optical flow information is available and trajectories are already estimated for each video sequence as in Chapter 7. Trajectories relate a point $\mathbf{x}(t)$ in frame t with L other consecutive frames. Since it is assumed that the scene is static, the same space point X is seen in image coordinates $\mathbf{x}(t), \dots, \mathbf{x}(t + L - 1)$. To avoid tracking outliers, trajectories with $L < 3$ are discarded. Since trajectories may not cover the whole video domain and may only be present in a subset of the frames, N-view factorization methods such as (Tomasi

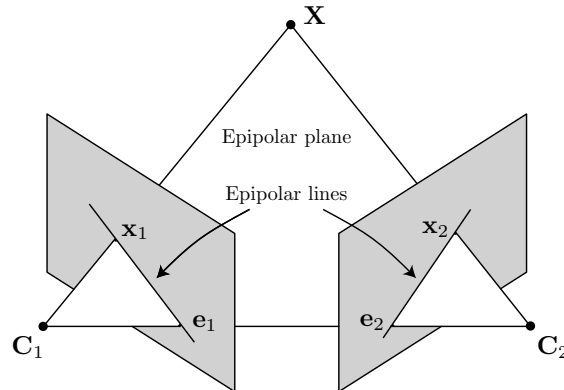


Figure 8.5: Epipolar geometry illustration. A point X projected onto two image planes create a plane with the camera centers. The points where the line joining the two centers intersect with the image plane are called epipoles, and correspond to the right and left null space of the fundamental matrix.

and Kanade 1992) and (Triggs 1996) cannot be applied. Instead, an algorithm which incrementally builds the video structure frame by frame is proposed.

The algorithm idea is to first extract projection matrices and spatial structure from relations of the first two frames of the video. After that, frame by frame, each camera projection matrix is obtained as well as the spatial position of appearing points. As the first step, two view structure computation is a key aspect of the system, so a detailed explanation of the two-view geometry and the algorithm strategy is given below.

8.1.3 Two View Geometry

When the same space point X is seen in two images with projections x_1 and x_2 , it is possible to relate the projection of each image in the following way. The two camera centers, namely C_1 and C_2 , and the point X form a plane in the three dimensional space as shown in Fig. 8.5. This geometric formation is known as epipolar geometry and can be algebraically expressed using the fundamental matrix. Multiple properties can be derived from the epipolar geometry, but here only the most important will be mentioned, see (Z. Zhang 1998) for a complete explanation. The most important property of the epipolar geometry is that the following relation holds:

$$x_2^\top \mathbf{F} x_1 = 0 \quad (8.7)$$

with \mathbf{F} being the fundamental matrix which has rank 2. The fundamental matrix is estimated for the first two frames of the video using the normalized eight point algorithm (R. I. Hartley 1997) which proved to be numerically very stable. Knowing only the matrix \mathbf{F} is the first step towards structure recovery. From \mathbf{F} the pair of projection cameras of the two images can be retrieved, (Sturm 1997). The two projection cameras are known as canonical cameras, and their formal expression is:

$$\tilde{\mathbf{P}}_1 = [\mathbf{I}_3 | \mathbf{0}_3] \quad (8.8)$$

$$\tilde{\mathbf{P}}_2 = [\mathbf{E}\mathbf{F} | \mathbf{e}_2] \quad (8.9)$$

where \mathbf{e}_2 is a vector such that $\mathbf{e}_2^\top \mathbf{F} = \mathbf{0}$ and it is called the epipole on the second image. The matrix \mathbf{E} is a skew-symmetric matrix formed with $\mathbf{e}_2 = [e_x, e_y, e_z]$:

$$\mathbf{E} = \begin{bmatrix} 0 & -e_z & e_y \\ e_z & 0 & -e_x \\ -e_y & e_x & 0 \end{bmatrix} \quad (8.10)$$

When the two canonical projection matrices are found, it is possible to triangulate the point position to find the 3D coordinates of all points $\tilde{\mathbf{X}}$ the projection of which is present in the two images. The used algorithm (Kanatani 2008) is an optimal triangulation method, which minimizes the projection error of the point correspondences. Other algorithms such as (R. I. Hartley and Sturm 1997) were tried, but the first produced more stable results, (Kanatani, Sugaya, et al. 2008). To understand the basics of optimal triangulation, a brief explanation follows. The previously cited works consider that the projection of a 3D point to each image is contaminated by additive Gaussian noise:

$$\tilde{\mathbf{x}}_1 = \mathbf{x}_1 + \boldsymbol{\sigma}_1 = \tilde{\mathbf{P}}_1 \tilde{\mathbf{X}} \quad (8.11)$$

$$\tilde{\mathbf{x}}_2 = \mathbf{x}_2 + \boldsymbol{\sigma}_2 = \tilde{\mathbf{P}}_2 \tilde{\mathbf{X}} \quad (8.12)$$

where $\tilde{\mathbf{x}}_{1,2}$ are the observed projections and $\mathbf{x}_{1,2}$ are the projections under the absence of noise, represented by vectors $\boldsymbol{\sigma}_{1,2}$. Optimal triangulation finds a space vector $\widehat{\mathbf{X}}$ and two 'optimal' projections $\widehat{\mathbf{x}}_1, \widehat{\mathbf{x}}_2$ such that it minimizes the following error:

$$|\tilde{\mathbf{x}}_1 - \widehat{\mathbf{x}}_1|^2 + |\tilde{\mathbf{x}}_2 - \widehat{\mathbf{x}}_2|^2 \quad (8.13)$$

$$|\tilde{\mathbf{x}}_1 - \tilde{\mathbf{P}}_1 \widehat{\mathbf{X}}|^2 + |\tilde{\mathbf{x}}_2 - \tilde{\mathbf{P}}_2 \widehat{\mathbf{X}}|^2 \quad (8.14)$$

Eq. (8.13) is the basic of bundle adjustment algorithms (Triggs et al. 2000) and can be easily generalized to several images instead of two. The common name in the

literature for Eq. (8.13) is ‘reprojection error’ as it measures how the true space point deviates from their ideal projection in both images due to noise. It follows from (R. I. Hartley and Sturm 1997) that for the case of two points in two images, there exists a special (and somewhat simple) solution to Eq. (8.13) which involves solving a low-degree polynomial. The same is not true when the reprojection error is considered in several images. Extensions to multiple image should be done in other ways, see Sec. 8.1.4 for more details. The general cases of reprojection error and bundle adjustment are essential to structure recovery due to the finite precision of algebraic methods in computers and they should be used in all the systems (Triggs et al. 2000).

Up to this point, we have recovered the camera matrices for the first two frames of a video, along with the structure of the point trajectories visible in those frames. Nevertheless, this reconstruction is not unique. Consider a 4 by 4 invertible matrix \mathbf{H} , it can be shown that the fundamental matrix \mathbf{F} representing a pair of camera matrices $(\mathbf{P}_1, \mathbf{P}_2)$ and $(\mathbf{P}_1\mathbf{H}^{-1}, \mathbf{P}_2\mathbf{H}^{-1})$ is the same. This can be seen from the fact that $\mathbf{x}_1 = \mathbf{P}_1\mathbf{X} = (\mathbf{P}_1\mathbf{H}^{-1})(\mathbf{H}\mathbf{X}) = \widehat{\mathbf{P}}\widehat{\mathbf{X}}$. Therefore, the same image projection \mathbf{x}_1 is performed by the point-camera pairs $(\mathbf{X}, \mathbf{P}_1)$ and $(\mathbf{H}\mathbf{X}, \mathbf{P}_1\mathbf{H}^{-1})$. This fact also holds for the second image.

Therefore, recovering the structure from two views can only be achieved up to a transformation of the 3D-space, represented by the rectifying homography matrix \mathbf{H} . For this reason, the initial structure computation is known as projective reconstruction and may be very different from the real (euclidean) reconstruction. The proposed algorithm finds the 3D structure of the scene as follows:

- Find a projective reconstruction for the whole video sequence, Sec. 8.1.5
- Apply autocalibration to find a rectifying homography \mathbf{H} , Sec. 8.1.6
- Transform the camera matrices and the points to obtain an euclidean reconstruction, Sec. 8.1.6

When the structure is recovered for the first two frames, camera poses and the spatial position of other points are obtained incrementally, one frame at a time. Since triangulating points and finding camera matrices involve complex algebraic operations, the solutions obtained are refined at each step with non-linear optimization. A special case of non-linear optimization in structure from motion algorithms is known as bundle adjustment and allows to improve initial solutions. Since improvements are noticeable (Triggs et al. 2000), bundle adjustment is included in the proposed system and a brief explanation follows.

8.1.4 Bundle Adjustment

Although bundle adjustment is not a particular step of the system, it is included in every step of the system where an algebraic decomposition is made. That is, each time the systems needs to perform a singular value decomposition, an inverse or any kind of matrix factorization (LU, Cholesky), a bundle adjustment step is added to prevent precision errors to propagate to subsequent steps. If the precision of actual computers was not an issue, bundle adjustment would not be necessary. But, since modern CPU have a finite precision, errors propagate and can possibly cause the system to fail.

The main idea of bundle adjustment is to refine an initial solution of a set of parameters θ to minimize a geometric magnitude, such as the projection error, which can be non-linear with θ . Bundle adjustment is an iterative minimization technique applied to geometric cost functions (R. Hartley and Zisserman 2004). Due to the vast literature on this subject, the reader is referred to the previous citation for a detailed survey on iterative methods. Here only the bundle adjustment case is explained. In structure recovery, the most common magnitude to minimize is the so-called reprojection error. Consider a set of 3D points \mathbf{X}_i and a set of observed projections of these points \mathbf{x}_{il} , where the subscript l refers to the view the projection belongs to. The bundle adjustment finds projection matrices and 3D points which minimize the distance of the projection with the observed values:

$$\min_{\mathbf{P}_l, \mathbf{X}_i} \sum_l \sum_i D(\mathbf{x}_{il}, \mathbf{P}_l \mathbf{X}_i) \quad (8.15)$$

where $D(\cdot)$ is a distance function (commonly the euclidean). Generally, projection matrices and points are parametrized with camera parameters such as position or rotation. In that case, the algorithm finds the optimal position and rotation angles such that Eq. (8.15) is minimized. The minimization needs a starting point close to the minimum to maximize the odds of convergence (although not guaranteed) and it is carried out using a Levenberg-Marquardt algorithm exploiting the sparse structure of the problem (Triggs et al. 2000). Many solvers are available but, for this thesis, the solver (*Ceres Solver*) is used.

8.1.5 Incremental Projective Structure Recovery

Assume that we observe a set of N point trajectories, with each trajectory $t_i = \{\mathbf{x}(t_{0i}) \dots \mathbf{x}(t_{0i} + L_i - 1)\}$ formed by a set of L_i points present in consecutive frames from t_{0i} to $t_{0i} + L_i - 1$. When the initial projective structure is recovered from the first two frames, each of the trajectories with $t_{0i} = 0$ has a corresponding point in space $\widetilde{\mathbf{X}}_i$.

When moving to the third frame, there will be trajectories with $t_{0i} < 2$ that have already a position $\widetilde{\mathbf{X}}_i$ triangulated because their projection was present in previous frames. If there is a sufficient number of trajectories, the projection matrix for the third frame can be linearly obtained. Consider that a point of a trajectory t_i is observed in the third frame. For notation simplicity we will call this point in homogeneous coordinates $\mathbf{x} = (x, y, 1)$, its corresponding space point \mathbf{X} and the projection matrix to be recovered as $\mathbf{P} = [\mathbf{P}_1^\top, \mathbf{P}_2^\top, \mathbf{P}_3^\top]$ with \mathbf{P}_1 being one row of the projection matrix. By knowing that $\lambda\mathbf{x} = \mathbf{P}\mathbf{X}$, it is possible to rewrite the previous equality as:

$$\mathbf{P}_3\mathbf{X}x - \mathbf{P}_1\mathbf{X} = 0 \quad (8.16)$$

$$\mathbf{P}_3\mathbf{X}y - \mathbf{P}_2\mathbf{X} = 0 \quad (8.17)$$

Therefore, each point contributes to two equations constraining the matrix \mathbf{P} , so a total of six known points suffice to find \mathbf{P} . In practice many more correspondences are known, so a least squares solution is found by singular value decomposition (SVD). As said before, after SVD, a bundle adjustment step to refine \mathbf{P} is performed so that

$$\mathbf{P}^* = \arg \min_{\mathbf{P}} \sum_i D(\mathbf{x}_i, \mathbf{P}\mathbf{X}_i) \quad (8.18)$$

with \mathbf{P} , a 3-by-4 matrix, parametrized using 11 variables, as the scale factor does not matter. After the optimum is found, the projective camera pose is obtained for the third frame, $\widetilde{\mathbf{P}}_3 = \mathbf{P}^*$.

Once the projection matrix is found for the third image, there may exist some trajectories starting in the second frame that may not have a corresponding space point. As the projection matrices $\widetilde{\mathbf{P}}_2$ and $\widetilde{\mathbf{P}}_3$ are known, it is possible to follow a similar strategy than Eqs. (8.16) and (8.17) to find a space point for each trajectory. Since a space point in homogeneous coordinates has four variables, two projections are sufficient to triangulate its spatial position. This process can be repeated for each new frame on the sequence, obtaining all matrices $\widetilde{\mathbf{P}}_l$ and all 3D projective positions \mathbf{X}_i for all trajectories and views of the sequence. The process of incremental structure recovery can be summarized into the following steps:

- Begin with projective cameras $\widetilde{\mathbf{P}}_1$ and $\widetilde{\mathbf{P}}_2$
- For each new view l :
 - Obtain a camera matrix $\widetilde{\mathbf{P}}_l$ for view l from triangulated points
 - Refine $\widetilde{\mathbf{P}}_l$ with Eq. (8.18)

- Obtain new point triangulations using the obtained camera matrices
- Process the next frame if available

Once all points and cameras are obtained (up to a projective transformation) it is then possible to find a rectifying homography \mathbf{H} to recover an euclidean reconstruction (autocalibration). Prior to that, a global projective bundle adjustment is performed, to refine the found solutions and to prevent drift errors to accumulate by minimizing:

$$\min_{\tilde{\mathbf{P}}_l, \tilde{\mathbf{X}}_i} \sum_l \sum_i D(\mathbf{x}_{il}, \tilde{\mathbf{P}}_l \tilde{\mathbf{X}}_i) \quad (8.19)$$

once the refinement is done, it is possible to proceed to the autocalibration step of the system. The minimization is done by keeping the first camera matrix fixed, as it used as a reference frame.

8.1.6 Autocalibration for Metric Reconstruction

Once the camera matrices are estimated in all views and the spatial structure of the points is also recovered, it is likely that the recovered structure does not respect the real world geometry. That is, parallelism, distances and other geometric magnitudes are not respected in projective transformations. The goal of autocalibration (or self calibration) is to find the transformation that maps the recovered projective structure and camera matrices to the real ones. In other words, the goal is to recover the camera internal \mathbf{K} and external parameters (position and rotation) for each projection view. There are many approaches to self calibration, but they can be divided into two kinds:

- Approaches that use the Kruppa equations ([O. D. Faugeras et al. 1992](#))
- Approaches that use the absolute quadric ([Triggs 1997](#))

Since the former kind of methods involve a large amount of non-linear programming, using the absolute quadric for self calibration proves to give more stable results ([Ponce, McHenry, et al. 2005](#)). The absolute quadric is the dual image of an imaginary point conic at infinity π_∞ , and the points belonging fulfill $X^2 + Y^2 + Z^2 = 0$, see ([Chandraker et al. 2007](#)) for a detailed explanation and [Fig. 8.6](#) for an illustration of the concept. The absolute quadric can be represented by a 4-by-4 matrix Ω_∞^* and when the obtained

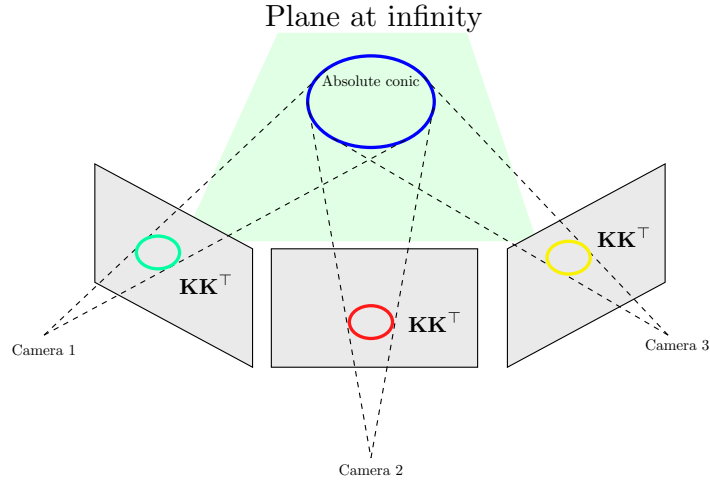


Figure 8.6: Illustration of the dual image of the absolute conic. Regardless of the camera pose, the dual conic project equally to each camera view plane. The conic formed in each plane can be expressed as the matrix $\mathbf{K}_l \mathbf{K}_l^\top$

camera matrices correspond to a metric space it can be expressed as:

$$\Omega_\infty^* = \text{diag}(1, 1, 1, 0) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (8.20)$$

if the recovered camera matrices do not correspond to a metric reconstruction (which is likely the case for the initial reconstruction), the absolute quadric matrix will have another form. The most important property of Ω_∞^* is that it fulfills:

$$\mathbf{K}_l \mathbf{K}_l^\top = \mathbf{P}_l \Omega_\infty^* \mathbf{P}_l^\top \quad (8.21)$$

where \mathbf{K}_l is the matrix encoding the internal camera parameters for each view l , see Fig 8.6. In a projective space, the matrix Ω_∞^* is a rank-3, symmetric positive definite matrix. If some constraints about the camera calibration parameters \mathbf{K} are known, a set of linear equations can be imposed on Ω_∞^* coefficients. Depending on the number of constraints on \mathbf{K}_l , the number of views needed to find Ω_∞^* may vary from 2 to 10 (R. Hartley and Zisserman 2004). In this thesis, we follow the work of (Pollefeys et al. 2004) to cope with critic cases.

We first assume that $\mathbf{K}_l = \mathbf{K} \forall l$, so internal parameters are constant. The idea is to transform each camera matrix by a 'normalizing' matrix $\tilde{\mathbf{K}} = \mathbf{K}_0^{-1} \mathbf{K}$ introducing a

priori knowledge of the parameters so the matrix $\tilde{\mathbf{K}}\tilde{\mathbf{K}}^\top$ can be expressed as:

$$\mathbf{K}_0 = \begin{bmatrix} w + h & 0 & \frac{w}{2} \\ 0 & w + h & \frac{h}{2} \\ 0 & 0 & 1 \end{bmatrix} \quad (8.22)$$

$$\tilde{\mathbf{K}}\tilde{\mathbf{K}}^\top = \begin{bmatrix} 1 \pm 9 & \pm 0.01 & \pm 0.1 \\ \pm 0.01 & 1 \pm 9 & \pm 0.1 \\ \pm 0.1 & \pm 0.1 & 1 \pm 9 \end{bmatrix} \quad (8.23)$$

where w and h are the video width and height in pixels respectively. As $\tilde{\mathbf{K}}\tilde{\mathbf{K}}^\top$ is symmetric, each view contributes to a total of six equations (the upper part of the matrix). Moreover, the absolute quadric is a 4 by 4 symmetric matrix which can be parametrized with 10 coefficients, so two views may be sufficient to find $\tilde{\mathbf{K}}$, and thus \mathbf{K} . Normally, however, many more views are needed as self calibration is very sensitive to noise. In the proposed algorithm we use all available frames to find a solution for Ω_∞^* . For a projection matrix $\mathbf{P} = [\mathbf{P}_1^\top, \mathbf{P}_2^\top, \mathbf{P}_3^\top]$, and combining Eq. (8.21) and Eq. (8.23), the six generated equations are:

$$\frac{1}{\beta}(\mathbf{P}_1\Omega_\infty^*\mathbf{P}_1^\top - \mathbf{P}_3\Omega_\infty^*\mathbf{P}_3^\top) = 0 \quad (8.24)$$

$$\frac{1}{\beta}(\mathbf{P}_2\Omega_\infty^*\mathbf{P}_2^\top - \mathbf{P}_3\Omega_\infty^*\mathbf{P}_3^\top) = 0 \quad (8.25)$$

$$\frac{1}{0.2\nu}(\mathbf{P}_1\Omega_\infty^*\mathbf{P}_1^\top - \mathbf{P}_2\Omega_\infty^*\mathbf{P}_2^\top) = 0 \quad (8.26)$$

$$\frac{1}{0.01\nu}(\mathbf{P}_1\Omega_\infty^*\mathbf{P}_2^\top) = 0 \quad (8.27)$$

$$\frac{1}{0.1\nu}(\mathbf{P}_1\Omega_\infty^*\mathbf{P}_3^\top) = 0 \quad (8.28)$$

$$\frac{1}{0.1\nu}(\mathbf{P}_2\Omega_\infty^*\mathbf{P}_3^\top) = 0 \quad (8.29)$$

where ν and β are weighting factors for numerical stability. Note that, theoretically ν, β should not have any impact of the solution, but practice shows that they are indeed crucial. The uncertainty of the the matrix $\tilde{\mathbf{K}}\tilde{\mathbf{K}}^\top$ is set by assuming reasonable values in (Pollefeys et al. 2004) and proved to give good results. The solution is found again by least squares, performing an SVD to the generated linear system. To prevent numerical errors, the algorithm (Thormählen et al. 2006) is followed: the value of β in (8.24),(8.25) is varied exponentially $\beta = 0.1 \exp(0.1n)$ with $n = 1 \dots 50$. ν is kept fixed at $\nu = 1$. A total of 50 possible solution are found, and the one that minimizes the criterion from (Nistér 2001) is considered to be the final solution.

Once Ω_∞^* is found for the projective reconstruction, it is possible to recover the rectifying homography \mathbf{H} by knowing Eq. (8.20). So, the decomposition $\text{diag}(1, 1, 1, 0) = \mathbf{H}\Omega_\infty^*\mathbf{H}^\top$ can be performed using SVD. Once \mathbf{H} is known, cameras and points are transformed according to:

$$\mathbf{P}_l = \tilde{\mathbf{P}}_l\mathbf{H}^{-1} \quad (8.30)$$

$$\mathbf{X}_i = \mathbf{H}\mathbf{X}_i \quad (8.31)$$

When the camera matrices are transformed by \mathbf{H}^{-1} internal parameters \mathbf{K} and the rotation matrix \mathbf{R} can be retrieved by QR-decomposition (Lawson and Hanson 1974), as well as the camera center \mathbf{C} . With this set of parameters, a final refinement on the found solution is performed using bundle adjustment.

8.1.6.1 Final bundle adjustment

In a metric space, a camera matrix can be written as $\mathbf{P}_l = \mathbf{K}_l [\mathbf{R}_l | \mathbf{T}_l]$ with \mathbf{R}_l being a rotation matrix (3 angles), $\mathbf{T}_l = -\mathbf{R}_l\mathbf{C}_l$ being the rotated center of the camera (3 coordinates) and \mathbf{K}_l having two focal lengths and two principal points (4 parameters). Additionally, the second order radial distortion δ_l (Weng et al. 1992) is taken into account, so a total of $3 + 3 + 4 + 1 = 11$ parameters are needed for each camera. The bundle optimization minimizes the following quantity:

$$\min_{\mathbf{K}_l, \mathbf{R}_l, \mathbf{T}_l, \delta} \sum_l \sum_i D(\mathbf{x}_{il}, \text{proj}(\mathbf{X}_i, \mathbf{K}_l, \mathbf{R}_l, \mathbf{T}_l, \delta_l)) \quad (8.32)$$

where $\text{proj}(\cdot)$ is the (non-linear) projection of the point \mathbf{X} taking all the parameters into account. The solution of this minimization gives a set of parameters for each camera and an absolute position for all points which can then be represented in depth maps or point clouds. As an example on the structure recovery represented on point clouds, Fig. 8.7 shows three examples of reconstructions represented in point clouds. Although trajectories do not cover the whole video, they are sufficiently dense to be visualized and to provide a good interpretation of the observed scene. Nevertheless, it is normally desirable to have dense depth maps for each frame so a depth is available for each point in the video. To this end, the following section discusses a possible way to post-process the reconstructed points so as to obtain such dense representations.

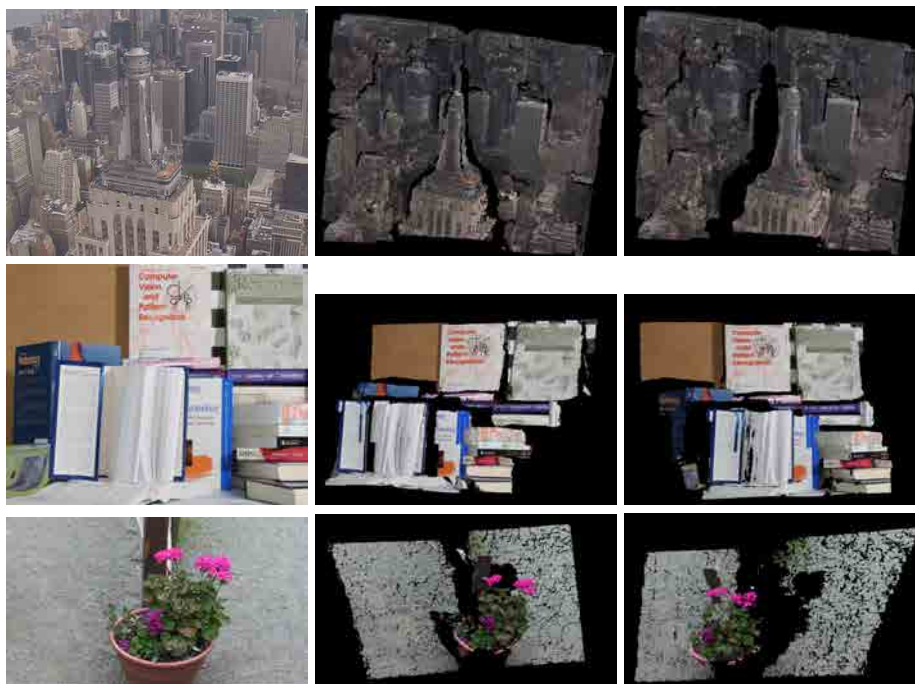


Figure 8.7: Examples of generated point clouds. The first column shows the first image of the sequence and the two other columns show the point cloud from two different orientations. Each point is colored by its mean average color over time.

8.1.7 Depth-map post-processing

The only last step to generate depth maps from video sequences is to take non reliable points from the video into account. Recall that the trajectory tracking algorithm in Chapter 7 only tracked points which had reliable flow. Thus, depending on the scene motion, there may be many areas on the image that do not have a trajectory associated to them, see Fig. 8.8.

After reconstruction, points in frames associated to a trajectory have also an associated depth. Assuming that the projection of a point $\mathbf{X} = (X, Y, Z, 1)$ corresponds to the image point $\lambda \mathbf{x} = \mathbf{P}\mathbf{X}$, the absolute depth of the point associated to a camera matrix \mathbf{P} is given by:

$$\text{depth}(\mathbf{X}, \mathbf{P}) = \frac{\lambda}{|\mathbf{m}|} \quad (8.33)$$

where $\mathbf{m} = (P_{31}, P_{32}, P_{33})^\top$ is a vector containing the first three coordinates of the third row of \mathbf{P} . \mathbf{m} corresponds to the vector marking the direction of the camera and it is pointing towards the front of the camera, perpendicularly to the image plane (R. Hartley and Zisserman 2004). Then, for every reconstructed point \mathbf{X}_i it is possible to

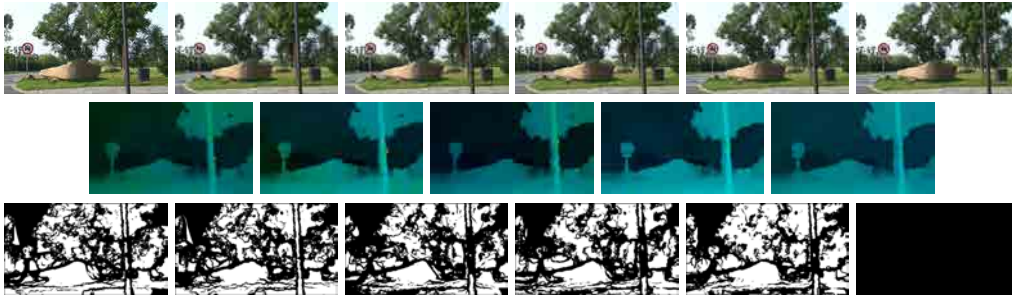


Figure 8.8: Example of flow reliability for a particular sequence. The top row shows the original frame, the middle row shows the forward flow field and the last row shows the flow reliability. White pixels are reliable pixels. Note that flow is not reliable in strong depth gradients or in homogeneous zones. The last image is black because there is no optical flow field.

assign its depth at each view where its projection is visible. Since the depth maps still contain points with unassigned depth, inpainting techniques are used to ‘fill the gaps’. The two most common approaches to inpaint images are (Bertalmio et al. 2000) and (Telea 2004). Both techniques work similarly, and the only difference is in the way they update missing information. From an initial image and an inpainting mask (telling which regions to inpaint) the algorithms proceed iteratively until no pixel is updated and the inpainting has ended. The former incrementally updates the holes using a level sets approach combined with anisotropic diffusion (Perona and Malik 1990), while the latter updates each pixel using a weighted neighborhood filter taking into account the direction of propagation. The main limitation of these works is that inpainting an image is done using the same image information. Since obtained depth maps may have large undefined region, inpainting the obtained depth map using only depth information may lead to an over smoothing of the depth edges. Nevertheless, if depth information is propagated using also color information, the *inpainted* depth map is much more sharp.

In the works of (Dimiccoli 2009) and (Calderero and Caselles 2013) a diffusion filter is used to propagate depth information using the color image. As the diffusion is performed using a linear filter, the process may average neighbor depth values, converging when all values are equal to a common depth value. In this thesis, we present an hybrid approach of inpainting and diffusion techniques. The main idea of the algorithm is to incrementally complete the depth image using color information. This hybrid approach is based on the iterative process of (Telea 2004), and the steps of the

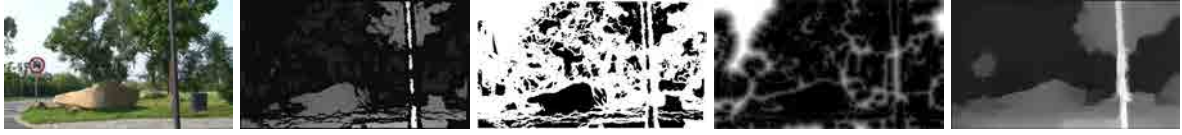


Figure 8.9: Depth inpainting example using color information. At the left image the first frame of a video is shown. The second shows the estimated depth, with brighter colors assigned to closer pixels. Unassigned depth pixels are black pixels. The third image corresponds to the mask (white pixels) where depth should be propagated. The fourth image is the distance transform of the mask. The last image is the inpainted depth.

algorithm are as follows:

- Perform the distance transform of the mask marking the ‘holes’ to be completed
- Order each pixel according to their distance to the boundary of the region to be inpainted
- Proceeding in increasing distance value for each pixel, find its depth value by a weighted average using the color image

Fig. 8.9 shows an example of depth inpainting using image color. Note that unreliable areas cover large regions on the generated depth image, so classical inpainting algorithms do not handle well these situations. More formally, the algorithm has as an input a color image I and an incomplete depth image D where a region Ω (possibly disconnected) needs to be filled with depth values. The region Ω is marked in the mask M of Fig. 8.9 with white pixels. The algorithm iteratively updates pixels from the boundary of the region, $\delta\Omega$, as follows:

$$D(\mathbf{p} \in \delta\Omega) = \frac{1}{Z(\mathbf{p})} \sum_{\mathbf{d} \in \Gamma} w(\mathbf{p}, \mathbf{d}) D(\mathbf{p} + \mathbf{d}) \quad (8.34)$$

where Γ is a circular neighborhood with radius $R = 10$ pixels. The factor $Z(\mathbf{p})$ is a normalization and its formal expression is $Z(\mathbf{p}) = \sum_{\mathbf{d} \in \Gamma} w(\mathbf{p}, \mathbf{d})$. The weights $w(\mathbf{p}, \mathbf{d})$ are defined with a similar technique as for diffusion filters, taking into account color and distance to the pixel to be updated:

$$w(\mathbf{p}, \mathbf{d}) = \exp\left(\frac{|I(\mathbf{p}) - I(\mathbf{p} + \mathbf{d})|}{\gamma}\right) \exp\left(\frac{|\mathbf{d}|}{R}\right) \quad (8.35)$$

where $\gamma = 14$ and controls how color difference influence. The averaging. Eq. (8.35) assigns more weight to close pixels with similar color, while pixels being far away or

with different color have a low influence in the current update. Each frame is completed independently, filling all the missing depths on the video. Next section shows results on the whole process, showing that trajectories obtained from the optical flow can be a good initialization to a complete depth recovery system.

8.2 Results

In contrast to single images, there is no standard benchmark to compare sequences structure from motion algorithms. There are many datasets to evaluate either stereo/disparity or multiview algorithms such as (Seitz et al. 2006). Many of the public benchmarks are either scenes on controlled environments or synthetic generated data. Since the approach presented here is a general way to retrieve depth from natural sequences using only optical flow, it is not specialized for stereo or multiview. Nevertheless, we present results qualitatively, comparing the estimated depth with state of the art approaches and with groundtruth disparity over a variety of datasets.

8.2.1 State of the Art Comparison

The proposed algorithm is compared to state of the art algorithm on depth estimation in video sequences (G. Zhang, Jia, T.-T. Wong, et al. 2009). As there is no standard dataset to evaluate depth recovery from videos, example results from (G. Zhang, Jia, T.-T. Wong, et al. 2009) are used for comparison. For the proposed depth estimation, each sequence is cropped to 7 frames, and optical flow is computed between frame pairs. The depth recovery uses trajectories estimated within these 7 frames. The experiment setup for (G. Zhang, Jia, T.-T. Wong, et al. 2009) is somewhat different, as all frames for each sequence were used (about 200) and the CPU time was 3 min/frame. The running time of the proposed algorithm is under 3 minutes for 7 frames, considerably faster.

Results show that the performance of the system is somewhat similar to the one in the cited work. Nevertheless, large areas involve unreliable flow information, the inpainting algorithm has problems filling large regions. Nevertheless, the presented approach achieves comparable depth map quality while improving by far the running time.

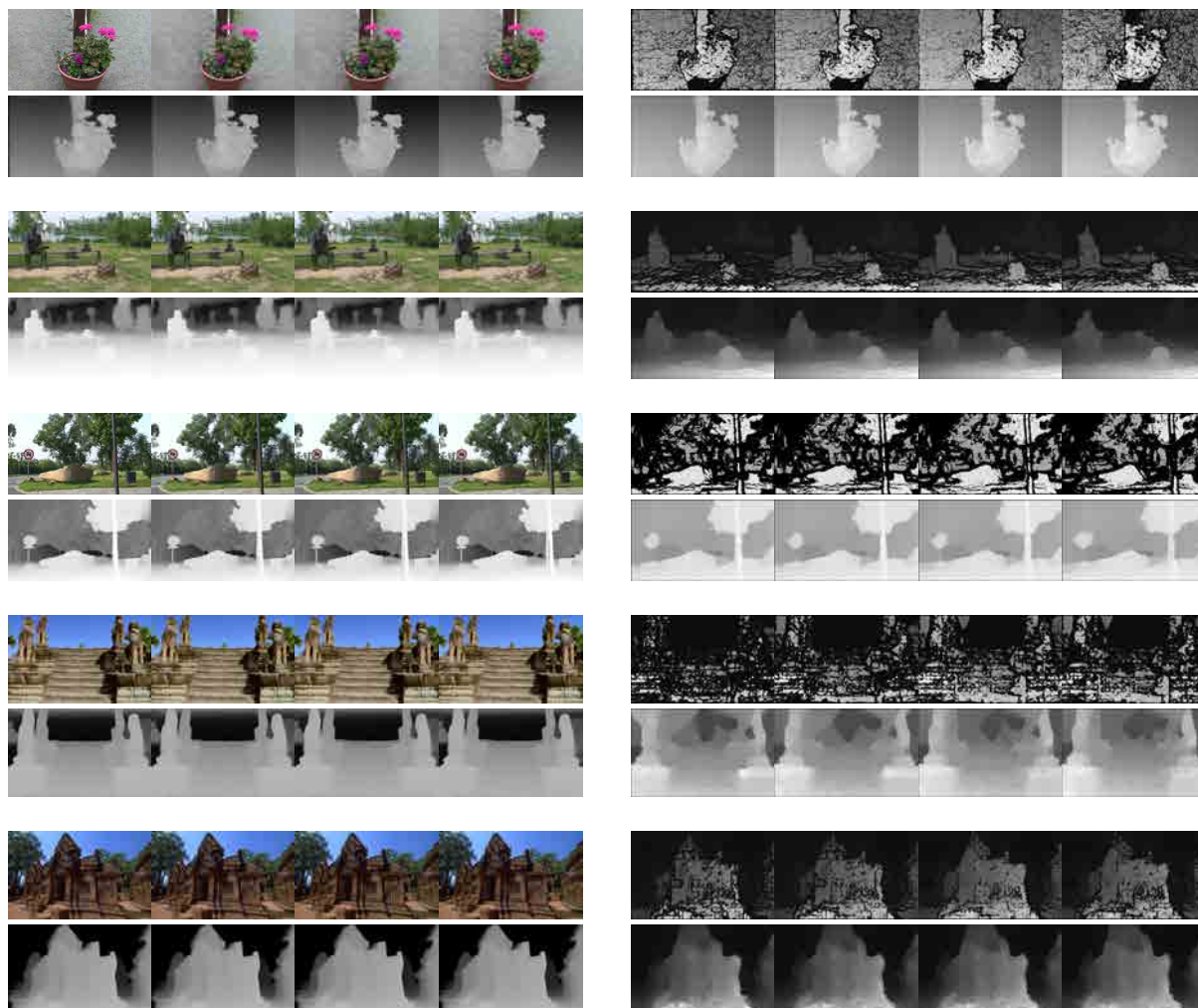


Figure 8.10: Results compared to (G. Zhang, Jia, T.-T. Wong, et al. 2009) on some sequences. Each group of two rows correspond to a different sequence. The first four images of the group correspond to frames of the sequence. Images directly below correspond to results of (G. Zhang, Jia, T.-T. Wong, et al. 2009). The right column shows the proposed system results. The first row of each group correspond to results without inpainting, and the row below it to inpainted results.

8.2.2 Groundtruth Comparison on Stereo

A second comparison is done by estimating depth on stereo sequences (Seitz et al. 2006). Although the stereo problem can be done much more precisely with the constraints of the setup (two cameras, aligned horizontally and calibrated), here we show that using optical flow for stereo problems can also produce good results. Quantitative evaluation was impossible on the proposed dataset, as the number of views required for camera calibration were not fulfilled by some sequences on the dataset.

Instead, qualitative results can be seen in Fig. 8.11. The proposed algorithm performs quite well in most of the cases, although effects of the inpainting algorithm can be seen at depth edges. On the generated results it can be seen that most structures and depth gradients are retrieved. If objects are dissimilar in color, inpainting missing depth values corrects well edges, while if two objects have close colors, some depth values leak from one object to another (see the second row in Fig. 8.11). Moreover, the over-smoothness of the optical flow algorithm is observed when many fine detail is present where these small structures are missed, for example in the sixth and seventh rows. In overall, the proposed algorithm is able to recover the depth with reasonable precision.

8.2.3 Limitations

The proposed algorithm is, compared to the state of the art, a low-complexity system able to retrieve absolute depth from video sequences of static backgrounds. The principal factor that limits the system is the quality of the estimated optical flow. There exist many high quality optical flow estimation algorithms, but many of them last minutes, even hours to compute the flow between pairs of frames, as (Sun et al. 2010) for example. If the video sequence consists of a few frames, this computation time may add to several days of optical flow estimation. We choose not to use these approaches but aim for a faster algorithm (Brox and Malik 2010), while maintaining a good quality on public benchmarks. Although here it is shown that depth can be recovered only from flow, the fact that many regions of the image contain not-reliable flow is a caveat for the algorithm, as inpainting may not work properly with large missing areas.

There are two other factors that may be limiting the system. First, there must exist some relative movement between the camera and the scene. And second, this movement must be rigid, with no independently moving objects. Future work will be devoted to the definition of an algorithm able to recover structure even with the presence of moving objects.

8. STRUCTURE FROM MOTION

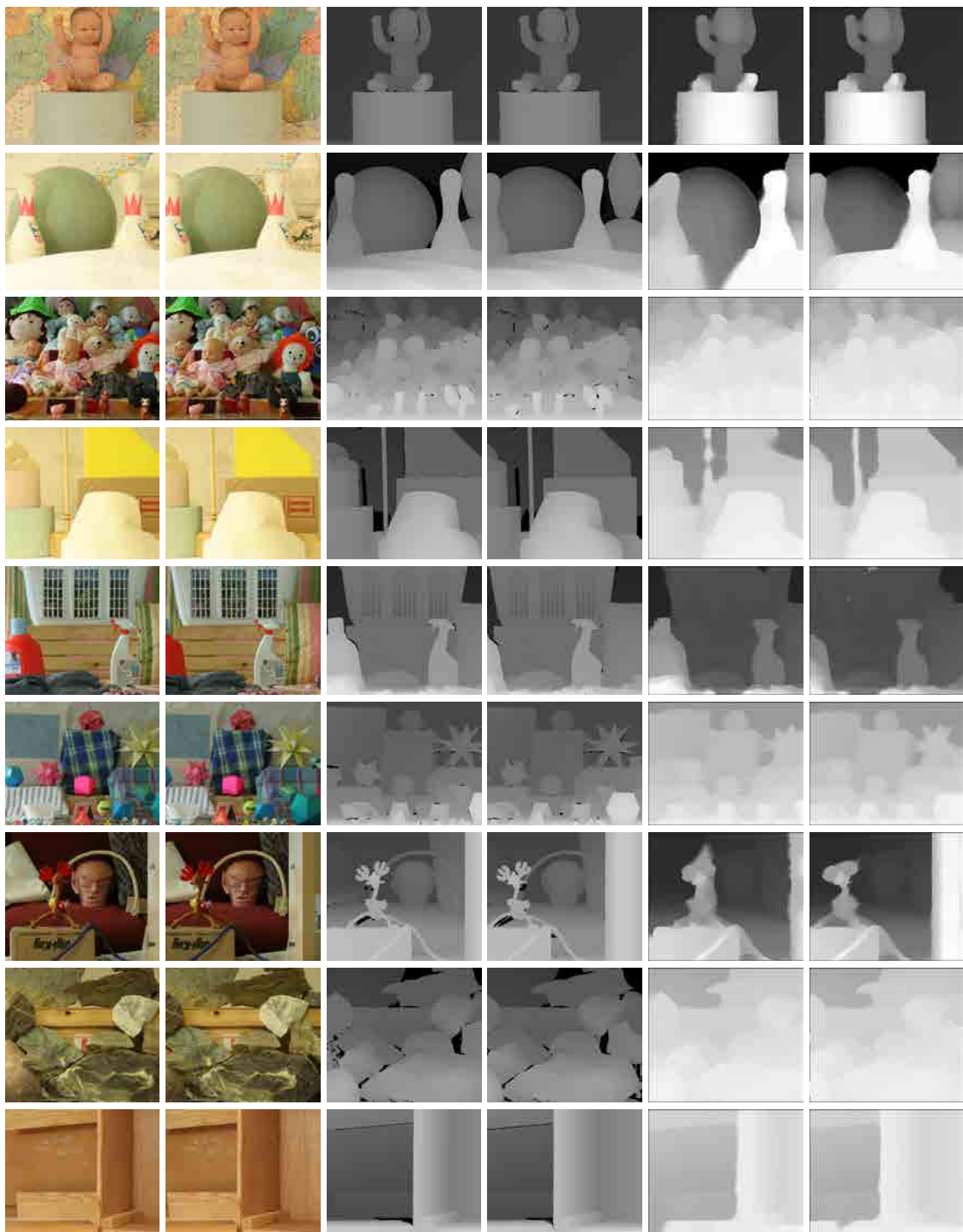


Figure 8.11: Examples on multiview stereo sequences. The two right images are the two reference frames. The central columns show the depth groundtruth, and the left two columns show the estimated depth. Note that the proposed system performs reasonably well on most sequences.

Part III

Conclusions

9 Conclusions

The work presented in this manuscript exposed a system which relied only on low level image cues for image and video segmentation and depth ordering. Regardless of the simplicity of the cues used to obtain the results, the different proposed algorithms offered results comparable with other approaches which based their reasoning on higher level information by means of learning. The proposed system shows several approaches that were not found in the literature. Several key aspects on BPT construction were addressed throughout the thesis, such as the introduction of T-junction estimation and optical flow to the BPT construction process.

The proposed BPT merging criteria, either in images and video showed results comparable with state of the art in several public datasets. Formally, the specific contributions of the segmentation process are:

- The segmentation is carried out using perceptual measures rather than using only statistical information like many state of the art algorithms use.
- The adaptive 3D-histogram region model was not used before, at least in the BPT structure, making us able to exploit the channel correlations on the distance measures.
- The Earth Mover's Distance, was not found to be exploited for a whole segmentation process though it was used in small image regions and/or image query applications. Its computational cost is the main factor that discourages its implementation in a whole segmentation system. The proposed histogram region model allows to implement the EMD as comparison measure between regions within a reasonable computation time.
- Specifically for video, the concept of 'flow reliability' is used to create spatio-temporal consistent segmentations, by exploiting color and motion information differently than the state of the art approaches.

A quantitative evaluation of the hierarchies and ways to process them using tree cuts are exposed, showing that optimization methods over the trees do indeed provide with better quality partition than the usual techniques performed until the date.

Additional contributions are also made in the evaluation process, where two new measures are proposed to evaluate depth order maps estimates. The proposed framework is generalizable to problems including detection and classification, and provides an

intuitive scheme to grasp system results using a single plot, rather than with multiple independent measures for detection and classification respectively. As for depth ordering, the detailed contributions are

- Push the limits of low-level cue performance for depth ordering.
- Propose a new probabilistic framework to propagate local depth information to a global depth consistency.
- Two new methods for T-junctions and motion occlusions estimation are proposed in the corresponding sections, showing results comparable to the state of art.

Although the most important innovations of the system have been commented, the system may suffer from several limitations, which are commented next.

9.1 Limitations

Projecting a 3D world to a 2D plane has an inherent loss of information which cannot be recovered completely using a single point of view. Therefore, it is impossible to recover the exact depth map using only image information. Rough depth interpretations can be computed if additional information is available, such as the type of observed scene or the type of objects present. This kind of information cannot be retrieved from within the input image/video, but a high-level learning process should take place using an external dataset and learning from examples. Therefore, low level cues are limited to produce an approximation of the current scene structure.

Further limitations of the system are found in the types of monocular depth cues used. As seen in the corresponding results sections, cues do not always indicate the correct depth order locally, but a global inference is needed. Many times, this global inference cannot correct error on low level measures so, incorrect depth orders may be produced.

Although these are the most characteristic limitations of a monocular depth ordering system, other weak points may be found in each particular stage of the system. Nevertheless, throughout the thesis we've shown that steps taken perform similarly or even

better than their respective state of the art approaches. By examining the weaknesses of the system several lines of work open for possible future work.

10 Open Problems and Future Research

The presented results for depth ordering show that with few suppositions, performance was similar (often better) to the current state of the art solutions. There are, however, some improvements that may be presented. Since the system performance is improved when perfect human segmentation is available, one could think that improving the segmentation process, the overall system performance should increase. Therefore, maybe some changes in the BPT construction part should be introduced to achieve such a goal. Other improvements of the system can be:

- The depth region model could allow a flexible surface orientation, having smooth depth gradients and not only depth discontinuities. Surface orientation may be estimated with state of the art techniques and further integrated with the occlusion detection.
- Adopting a low-level learning approach to avoid the current system limitations. Results show that a scene structure learning is a too-ambitious model for the infinite number of possible situations, but focusing the efforts on a low-level scheme (such as contours, junctions) could lead to better cue classification.
- Introducing more depth cues. For example, the use of haze, texture gradient, perspective cues to recover original depth maps could be a good complement to occlusions cues.

Furthermore, the proposed system could be extended to handle different types of situations, specially when depth cues are either absent or indicate contradictory information. Below two examples of such extensions are presented.

Joining Static depth cues and motion occlusions The most immediate work is to combine the two types of occlusion cues: static (T-junctions and convexity) with the dynamic ones (motion occlusions). It is not clear in the literature how humans combine these kind of cues. Moreover, in this thesis it is shown that motion occlusion work far better than static cues. Nevertheless, in many real world cases when the camera and world are static, motion cues are absent and other approaches to estimate depth should

be proposed. A system which combines the static and dynamic cues used in this thesis could be a starting point to deal with such cases.

Full depth maps recovery for video sequences If the scene is assumed to be composed of several rigid bodies moving freely, it is possible to recover partially its structure. When the scene is composed by a single static object, we have shown that it is possible to recover its structure up to a scale factor. If two or more bodies are present, it is also possible to recover each individual structure. With the introduction of motion occlusions, the different objects could be related and a global depth map could be estimated. A possible scheme for such a system could be:

1. Identify the number of rigid objects N with independent motions in a scene
2. Recover structure parameters for each object
3. Estimate the structure of these objects
4. Arrive at a global depth understanding using occlusions

In this way, it would be possible to combine structure from motion cues and motion occlusions to provide dense depth maps in most types of image sequences.

11 List of Publications

- G. Palou and G. Salembier. "Precision-Recall-Classification Evaluation Framework: Application to Depth Estimation on Single Images". In: *Submitted to CVPR*. 2014
- G. Palou and P. Salembier. "Occlusion-based depth ordering on monocular images with Binary Partition Tree". In: *IEEE ICASSP*. Prague, Czech Republic, 2011
- G. Palou and P. Salembier. "From local occlusion cues to global depth estimation". In: *IEEE ICASSP*. Kyoto, Japan, 2012
- G. Palou and P. Salembier. "Monocular Depth Ordering Using T-junctions and Convexity Occlusion Cues." In: *IEEE Trans. on Image Proc.* 2013
- G. Palou and P. Salembier. "2.1 Depth Estimation of Frames in Image Sequences Using Motion Occlusions." In: *ECCV Workshops*. Firenze, Italy, 2012

-
- G. Palou and P. Salembier. “Depth ordering on image sequences using motion occlusions”. In: *IEEE ICIP*. Orlando, FL, USA, 2012
 - G. Palou and P. Salembier. “Depth order estimation for video frames using motion occlusions”. In: *IET Computer Vision* 2013
 - G. Palou and P. Salembier. “Hierarchical Video Representation with Trajectory Binary Partition Tree”. In: *IEEE CVPR*. Portland, OR, USA, 2013
 - G. Palou and P. Salembier. “Hierarchical Video Representation with Trajectory Binary Partition Tree and its Applications”. In: *IEEE TPAMI*, *in peer review* 2013

Part IV

References

References

- Agarwal, S., K. Mierle, et al. *Ceres Solver*. <https://code.google.com/p/ceres-solver/> (see p. 210).
- Aloimonos, J. "Shape from texture". In: *Biological cybernetics* 58.5 1988, pp. 345–360 (see p. 88).
- Alpert, S. et al. "Image Segmentation by Probabilistic Bottom-Up Aggregation and Cue Integration". In: *IEEE CVPR*. 2007, pp. 1–8 (see p. 95).
- Andersen, R. *Modern methods for robust regression*. n. 152. 2008 (see pp. 68, 69).
- Anderson, B. L. et al. "A theory of illusory lightness and transparency in monocular and binocular images: The role of contour junctions". In: *Perception London* 26 1997, pp. 419–454 (see p. 27).
- Anderson, B. L. "The role of occlusion in the perception of depth, lightness, and opacity." In: *Psychological review* 110.4 2003, p. 785 (see p. 27).
- Antonides, J. and T. Kubota. "Binocular disparity as an explanation for the moon illusion". In: *CoRR abs/1301.2715* 2013 (see p. 25).
- Arbelaez, P. "Boundary Extraction in Natural Images Using Ultrametric Contour Maps". In: *IEEE CVPR Workshop*. 2006, p. 182 (see pp. 106, 108).
- Arbeláez, P. et al. "Contour detection and hierarchical image segmentation." In: *IEEE TPAMI* 33.5 2011, pp. 898–916 (see pp. 79, 95, 106–108, 124, 136, 138, 151–153, 155, 156, 162, 174, 179, 185).
- Arbelaez, P. et al. "From contours to regions: An empirical evaluation". In: *IEEE CVPR*. 2009, pp. 2294–2301 (see p. 91).
- Asada, H. and M. Brady. "The Curvature Primal Sketch". In: *IEEE TPAMI PAMI-8.1* 1986, pp. 2–14 (see p. 49).
- Atick, J. J., P. A. Griffin, and A. N. Redlich. "Statistical Approach to Shape from Shading: Reconstruction of 3D Face Surfaces from Single 2D Images". In: *Neural Computation* 8 1997, pp. 1321–1340 (see p. 30).
- Ayer, S. and H. S. Sawhney. "Layered representation of motion video using robust maximum-likelihood estimation of mixture models and MDL encoding". In: *IEEE ICCV*. 1995, pp. 777–784 (see p. 170).
- Ayvaci, A., M. Raptis, and S. Soatto. "Occlusion detection and motion estimation with convex optimization". In: *Advances in Neural Information Processing Systems*. 2010, pp. 100–108 (see p. 65).

- Bajcsy, R. and L. Lieberman. "Texture gradient as a depth cue". In: *Computer Graphics and Image Processing* 5.1 1976, pp. 52–67 (see p. 31).
- Baker, S., R. Szeliski, and P. Anandan. "A layered approach to stereo reconstruction". In: *IEEE CVPR*. 1998, pp. 434–441 (see p. 37).
- Baker, S. and S. K. Nayar. "A theory of single-viewpoint catadioptric image formation". In: *IJCV* 35.2 1999, pp. 175–196 (see p. 45).
- Barnard, S. T. and M. A. Fischler. "Computational stereo". In: *ACM Computing Surveys* 14.4 1982, pp. 553–572 (see p. 206).
- Barron, J. T. and J. Malik. *Shape, Illumination, and Reflectance from Shading*. Tech. rep. UCB/EECS-2013-117. EECS, UC Berkeley, May 2013 (see p. 87).
- Basha, T., Y. Moses, and S. Avidan. "Photo Sequencing". In: *IEEE ECCV*. Ed. by A. Fitzgibbon et al. Vol. 7577. 2012, pp. 654–667 (see p. 129).
- Bay, H. et al. "Speeded-Up Robust Features (SURF)". In: *CVIU* 110.3 2008, pp. 346–359 (see p. 64).
- Bayes, M. and M. Price. "An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S." In: *Philosophical Transactions* 53 1763, pp. 370–418 (see p. 20).
- Beck, J., K. Prazdny, and R. Ivry. "The perception of transparency with achromatic colors". In: *Perception & Psychophysics* 35.5 1984, pp. 407–422 (see p. 42).
- Berg, A. C. and J. Malik. "Geometric blur for template matching". In: *IEEE CVPR*. Vol. 1. 2001, (see p. 90).
- Bergen, L. and F. Meyer. "A novel approach to depth ordering in monocular image sequences". In: *IEEE CVPR*. Vol. 2. 2000, 536–541 vol.2 (see pp. 42, 150, 170).
- Bergevin, R. and A. Bubel. "Detection and characterization of junctions in a 2D image". In: *Computer Vision and Image Understanding* 93.3 2004, pp. 288–309 (see p. 43).
- Bertalmio, M. et al. "Image inpainting". In: *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co. 2000, pp. 417–424 (see p. 217).
- Bertero, M., T. Poggio, and V. Torre. "Ill-posed problems in early vision". In: *Proceedings of the IEEE* 76.8 1988, pp. 869–889 (see pp. 24, 37).
- Beucher, S. and C. Lantuejoul. "Use of watersheds in contour detection". In: *International Workshop on Image Processing: Real-time Edge and Motion detection/estimation* 1979 (see p. 108).
- Black, M. J. and P. Anandan. "A framework for the robust estimation of optical flow". In: *IEEE ICCV*. 1993, pp. 231–236 (see pp. 64, 174).

- Black, M. J. and P. Anandan. "The robust estimation of multiple motions: parametric and piecewise-smooth flow fields". In: *CVIU* 63.1 1996, pp. 75–104 (see pp. 60, 63).
- Bond, C. "System and process for transforming two-dimensional images into three-dimensional images". Patent 20110050864. 2011 (see p. 87).
- Braunstein, M. L., D. D. Hoffman, and A. Saidpour. "Parts of visual objects: an experimental test of the minima rule." In: *Perception* 18.6 1989, pp. 817–826 (see pp. 13, 29).
- Brox, T., A. Bruhn, et al. "High accuracy optical flow estimation based on a theory for warping". In: *IEEE ECCV*. Vol. 3024. Prague, Czech Republic, 2004, pp. 25–36 (see pp. 60, 61, 63, 64).
- Brox, T. and J. Malik. "Large Displacement Optical Flow: Descriptor Matching in Variational Motion Estimation". In: *IEEE TPAMI* 33.3 2011, pp. 500–513 (see pp. 60–62, 64, 65, 154, 175, 176, 178, 191, 195).
- Brox, T. and J. Malik. "Object segmentation by long term analysis of point trajectories". In: *IEEE ECCV*. 2010, pp. 282–295 (see pp. 173, 174, 180, 182, 183, 190, 221).
- Buades, A., B. Coll, and J.-M. Morel. "A Review of Image Denoising Algorithms, with a New One". In: *Multiscale Modeling & Simulation* 4.2 2005, pp. 490–530 (see pp. 56, 98).
- Buades, A., B. Coll, and J.-M. Morel. "Neighborhood filters and PDEs". In: *Numerische Mathematik* 105.1 2006, pp. 1–34 (see p. 93).
- Bulthoff, H. H. "Bayesian decision theory and psychophysics". In: *Perception as Bayesian inference* 1996, p. 123 (see p. 36).
- Burge, J., C. C. Fowlkes, and M. S. Banks. "Natural-Scene Statistics Predict How the Figure-Ground Cue of Convexity Affects Human Depth Perception". In: *Journal of Neuroscience* 30.21 2010, pp. 7269–7280 (see p. 28).
- Burr, D. C. and J. Ross. "How does binocular delay give information about depth?" In: *Vision research* 19.5 1979, pp. 523–532 (see p. 25).
- Calderero, F. and V. Caselles. "Recovering relative depth from low-level features without explicit T-junction detection and interpretation". In: *IEEE IJCV* In Press 2013 (see pp. 42, 56–59, 92, 93, 124–126, 138, 217).
- Calderero, F. and F. Marques. "Region Merging Techniques Using Information Theory Statistical Measures". In: *IEEE Trans. on Image Proc.* 19.6 2010, pp. 1567–1586 (see pp. 98, 119, 184).
- Carreira, J. and C. Sminchisescu. "CPMC: Automatic Object Segmentation Using Constrained Parametric Min-Cuts". In: *IEEE TPAMI* 34.7 2012, pp. 1312–1328 (see pp. 20, 123).

- Chandraker, M. et al. "Autocalibration via rank-constrained estimation of the absolute quadric". In: *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. IEEE. 2007, pp. 1–8 (see p. 212).
- Chang, J.-Y. et al. "Relative Depth Layer Extraction for Monoscopic Video by Use of Multidimensional Filter". In: *Proc. IEEE Int Multimedia and Expo*. 2006, pp. 221–224 (see pp. 150, 170).
- Charbonnier, P. et al. "Two deterministic half-quadratic regularization algorithms for computed imaging". In: *IEEE ICIP*. Vol. 2. 1994, 168–172 vol.2 (see p. 63).
- Chen, A. Y. C. and J. J. Corso. "Propagating multi-class pixel labels throughout video frames". In: *Image Processing Workshop (WNYIPW)*. 2010, pp. 14–17 (see pp. 185, 187, 188).
- Cipolla, R., Y. Okamoto, and Y. Kuno. "Robust structure from motion using motion parallax". In: *IEEE ICCV*. 1993, pp. 374–382 (see p. 168).
- Clausi, D. A. and M. E. Jernigan. "Designing Gabor filters for optimal texture separability". In: *Pattern Recognition* 33.11 2000, pp. 1835–1849 (see p. 31).
- Clerc, M. and S. Mallat. "The texture gradient equation for recovering shape from texture". In: *IEEE TPAMI* 24.4 2002, pp. 536–549 (see p. 31).
- Cohen, I. "Nonlinear Variational Method for Optical Flow Computation". In: *8th Society for Industrial and Applied Mathematics*. 1993 (see p. 63).
- Colantoni, P. and B. Laget. "Color image segmentation using region adjacency graphs". In: *Image Processing and Its Applications*. Vol. 2. 1997, 698–702 vol.2 (see p. 95).
- Comaniciu, D. and P. Meer. "Mean shift: a robust approach toward feature space analysis". In: *IEEE TPAMI* 24.5 2002, pp. 603–619 (see pp. 95, 121).
- Cormen, T. H. et al. *Introduction to Algorithms*. 2nd Revise. 2001 (see pp. 114, 133).
- Corso, J. J. et al. "Efficient Multilevel Brain Tumor Segmentation With Integrated Bayesian Model Classification". In: *IEEE Trans. on Medical Imaging* 27.5 2008, pp. 629–640 (see pp. 172, 173, 180).
- Costeira, J. P. and T. Kanade. "A Multibody Factorization Method for Independently Moving Objects". In: *IEEE IJCV* 29.3 1998, pp. 159–179 (see pp. 171, 173).
- Cour, T. and F. Benezit. "Spectral Segmentation with Multiscale Graph Decomposition". In: *IEEE CVPR*. Vol. 2. 2005, pp. 1124–1131 (see pp. 95, 121, 123).
- Criminisi, A., I. Reid, and A. Zisserman. "Single view metrology". In: *IEEE ICCV*. Vol. 1. 1999, pp. 434–441 (see p. 89).
- Dalal, N. and B. Triggs. "Histograms of oriented gradients for human detection". In: *IEEE CVPR*. Vol. 1. IEEE. 2005, pp. 886–893 (see p. 64).

- Darrell, T. and K. Wohn. "Pyramid based depth from focus". In: *IEEE CVPR*. IEEE. 1988, pp. 504–509 (see p. 88).
- Davison, A. J. and D. W. Murray. *Mobile robot localisation using active vision*. Springer, 1998 (see p. 168).
- Davison, A. J., I. D. Reid, et al. "MonoSLAM: Real-Time Single Camera SLAM". In: *IEEE TPAMI* 29.6 2007, pp. 1052–1067 (see pp. 37, 169, 176).
- Day, G. S. and P. Schoemaker. "Peripheral vision: sensing and acting on weak signals". In: *Long Range Planning* 37.2 2004, pp. 117–121 (see p. 26).
- Delage, E., H. Lee, and A. Y. Ng. "A Dynamic Bayesian Network Model for Autonomous 3D Reconstruction from a Single Indoor Image". In: *IEEE CVPR*. Vol. 2. 2006, pp. 2418–2428 (see p. 89).
- Dimiccoli, M. "Monocular Depth Estimation for Image Segmentation and Filtering". PhD thesis. Universitat Politècnica de Catalunya, 2009 (see pp. 13, 27, 42, 43, 50, 93, 98, 217).
- Dorea, C. C., M. Pardàs, and F. Marques. "Trajectory tree as an object-oriented hierarchical representation for video". In: *IEEE Trans. on Circuits and Systems for Video Technology* 19.4 2009, pp. 547–560 (see p. 175).
- Dougherty, E. *Mathematical morphology in image processing*. CRC press, 1992 (see p. 108).
- Durgin, F. H. et al. "Comparing depth from motion with depth from binocular disparity". In: *Journal of Experimental Psychology: Human Perception and Performance* 21 1995, pp. 679–699 (see p. 206).
- Dwork, C. et al. "Rank aggregation methods for the Web". In: *Int. Conf. on World Wide Web*. New York, NY, USA, 2001, pp. 613–622 (see p. 129).
- Epstein, W. and S. Rogers. *Perception of space and motion*. Access Online via Elsevier, 1995 (see p. 17).
- Ernst, M. O. and M. S. Banks. "Humans integrate visual and haptic information in a statistically optimal fashion". In: *Nature* 415.6870 2002, pp. 429–433 (see p. 36).
- Escher, M. C. and J. E. Brigham. *The graphic work of MC Escher*. Vol. 960. 1967 (see p. 24).
- Faugeras, O. D., Q.-T. Luong, and S. J. Maybank. "Camera self-calibration: Theory and experiments". In: *IEEE ECCV*. Springer. 1992, pp. 321–334 (see pp. 169, 212).
- Felzenszwalb, P. F. and D. P. Huttenlocher. "Efficient Graph-Based Image Segmentation". In: *Int. J. Comput. Vision* 59.2 2004, pp. 167–181 (see pp. 95, 172, 179).
- Fernald, R. D. "The evolution of eyes." In: *Brain, behavior and evolution* 50.4 1997, pp. 253–9 (see p. 18).

- Fitzgibbon, A. W. and A. Zisserman. "Multibody structure and motion: 3-d reconstruction of independently moving objects". In: *IEEE ECCV*. Springer, 2000, pp. 891–906 (see p. 171).
- Fowlkes, C. C., D. R. Martin, and J. Malik. "Local figure-ground cues are valid for natural images." In: *Journal of Vision* 7.8 2007, p. 2 (see pp. 28, 54).
- Fowlkes, C. et al. "Spectral grouping using the Nystrom method". In: *IEEE TPAMI* 26.2 2004, pp. 214–225 (see p. 172).
- Fox, C. *An introduction to the calculus of variations*. 1950 (see p. 63).
- Galtier, J., A. Laugier, and P. Pons. "Algorithms to evaluate the reliability of a network". In: *Int. Workshop on Design of Reliable Communication Networks*. 2005, p. 8 (see p. 131).
- Ghita, O., P. F. Whelan, and J. Mallon. "Computational approach for depth from defocus". In: *Journal of Electronic Imaging* 14.2 2005, p. 23021 (see p. 88).
- Gibson, E. J. et al. "Motion parallax as a determinant of perceived depth." In: *Journal of experimental psychology* 58.1 1959, p. 40 (see p. 59).
- Gibson, J. J. *The Ecological approach to visual perception*. 1, 1986 (see p. 21).
- Gould, S., R. Fulton, and D. Koller. "Decomposing a scene into geometric and semantically consistent regions". In: *IEEE ICCV*. IEEE. 2009, pp. 1–8 (see p. 93).
- Gregory, R. L. "The Medawar lecture 2001 knowledge for vision: vision for knowledge". In: *Philosophical Transactions of the Royal Society B: Biological Sciences* 360.1458 2005, pp. 1231–1251 (see p. 34).
- Gregory, R. L. *Even odder perceptions*. 1994 (see p. 33).
- Grossman, E. et al. "Brain areas involved in perception of biological motion". In: *Journal of cognitive neuroscience* 12.5 2000, pp. 711–720 (see p. 60).
- Grundmann, M. et al. "Efficient hierarchical graph-based video segmentation." In: *IEEE CVPR*. 2010, pp. 2141–2148 (see pp. 95, 172, 173, 179, 180).
- Guichard, F. and J. M. Morel. "Image Analysis and {PDEs}". In: *Institute for Pure and Applied Mathematics GBM Tutorial* 2001 (see p. 49).
- Guo, C.-e., S.-C. Zhu, and Y. N. Wu. "Towards a mathematical theory of primal sketch and sketchability". In: *IEEE ICCV*. IEEE. 2003, pp. 1228–1235 (see p. 23).
- Guo, C.-e., S.-C. Zhu, and Y. N. Wu. "Primal sketch: Integrating structure and texture". In: *CVIU* 106.1 2007, pp. 5–19 (see p. 23).
- Guzmán, A. "Decomposition of a visual scene into three-dimensional bodies". In: *AFIPS*. New York, NY, USA, 1968, pp. 291–304 (see p. 27).
- Harris, C. and M. Stephens. "A Combined Corner and Edge Detection". In: *Alvey Vision Conf*. 1988, pp. 147–151 (see pp. 42, 46, 176, 177).

- Hartley, R. I. "In defense of the eight-point algorithm". In: *IEEE TPAMI* 19.6 1997, pp. 580–593 (see p. 208).
- Hartley, R. I. and P. Sturm. "Triangulation". In: *CVIU* 68.2 1997, pp. 146–157 (see pp. 208, 209).
- Hartley, R. and A. Zisserman. *Multiple View Geometry in Computer Vision*. Vol. 2. 2. 2004. Chap. 189, p. 672 (see pp. 168, 169, 202, 203, 210, 213, 216).
- He, X. and A. Yuille. "Occlusion Boundary Detection Using Pseudo-depth". In: *IEEE ECCV*. Vol. 6314. 2010, pp. 539–552 (see p. 151).
- Henson, D. *Visual Fields*. 1998 (see p. 26).
- Hildreth, E. C. *Measurement of Visual Motion*. Cambridge, MA, USA, 1984 (see p. 61).
- Hillier, F. S. and G. J. Lieberman. *Introduction to Mathematical Programming*. 1990 (see pp. 101, 103).
- Hillis, J. M. et al. "Combining sensory information: mandatory fusion within, but not between, senses". In: *Science* 298.5598 2002, pp. 1627–1630 (see p. 36).
- Hirschmuller, H. "Stereo processing by semiglobal matching and mutual information". In: *IEEE TPAMI* 30.2 2008, pp. 328–341 (see p. 37).
- Hochberg, J. E. and E. McAlister. "Relative size vs. familiar size in the perception of represented depth". In: *The American journal of psychology* 68.2 1955, pp. 294–296 (see p. 30).
- Hoffman, D. D. and M. Singh. "Salience of visual parts." In: *Cognition* 63.1 1997, pp. 29–78 (see p. 28).
- Hoiem, D., A. A. Efros, and M. Hebert. "Recovering Occlusion Boundaries from an Image". In: *IEEE IJCV* 91.3 2011, pp. 328–346 (see pp. 33, 37, 91, 92, 140, 143).
- Hoiem et al. "Recovering Surface Layout from an Image". In: *IEEE IJCV* 75.1 2007, pp. 151–172 (see p. 91).
- Holender, D. "Semantic activation without conscious identification in dichotic listening, parafoveal vision, and visual masking: A survey and appraisal". In: *Behavioral and brain Sciences* 9.1 1986, pp. 1–66 (see p. 37).
- Horn, B. K. P. and B. G. Schunk. "Determining Optical Flow". In: *Artificial Intelligence* 17 1981, pp. 185–203 (see pp. 60, 63, 168, 174).
- Horn, B. K. "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view". PhD thesis. MIT, 1970 (see p. 87).
- Hubona, G. S. et al. "The relative contributions of stereo, lighting, and background scenes in promoting 3D depth visualization". In: *ACM Trans. on Computer-Human Interaction* 6.3 1999, pp. 214–242 (see pp. 41, 87).

- Humayun, A., O. Mac Aodha, and G. J. Brostow. "Learning to find occlusion regions". In: *IEEE CVPR*. 2011, pp. 2161–2168 (see pp. 66, 70–72).
- ILOG, Inc. *ILOG CPLEX: High-performance software for mathematical programming and optimization*. See <http://www.ilog.com/products/cplex/>. 2006 (see p. 113).
- Ishikawa, H. and D. Geiger. "Segmentation by Grouping Junctions". In: *IEEE CVPR*. Washington, DC, USA, 1998, p. 125 (see p. 42).
- CIELAB standard colour image data* (2010). Norm. 2010 (see p. 98).
- Jacobs, R. A. "What determines visual cue reliability?" In: *Trends in cognitive sciences* 6.8 2002, pp. 345–350 (see p. 33).
- Johansson, G. "Visual perception of biological motion and a model for its analysis". In: *Perception & psychophysics* 14.2 1973, pp. 201–211 (see p. 60).
- Jojic, N. and B. J. Frey. "Learning flexible sprites in video layers". In: *IEEE CVPR*. Vol. 1. 2001, (see p. 170).
- Jones, R. K. and D. N. Lee. "Why two eyes are better than one: the two views of binocular vision." In: *Journal of Experimental Psychology: Human Perception and Performance* 7.1 1981, p. 30 (see p. 25).
- Kanatani, K. "Transformation of optical flow by camera rotation". In: *IEEE TPAMI* 10.2 1988, pp. 131–143 (see pp. 67, 68, 157, 168).
- Kanatani, K. "Statistical optimization for geometric fitting: Theoretical accuracy bound and high order error analysis". In: *International Journal of Computer Vision* 80.2 2008, pp. 167–188 (see p. 208).
- Kanatani, K., Y. Shimizu, et al. "Fundamental matrix from optical flow: optimal computation and reliability evaluation". In: *Journal of Electronic Imaging* 9.2 2000, pp. 194–202 (see p. 168).
- Kanatani, K., Y. Sugaya, and H. Niitsuma. "Triangulation from two views revisited: Hartley-Sturm vs. optimal correction". In: vol. 4. 2008, p. 5 (see p. 208).
- Karsch, K., C. Liu, and S. B. Kang. "Depth Extraction from Video Using Non-parametric Sampling". In: *IEEE ECCV*. 2012 (see pp. 151, 171).
- Kersten, D. and A. Yuille. "Bayesian models of object perception". In: *Current opinion in neurobiology* 13.2 2003, pp. 150–158 (see p. 33).
- King, M. T. et al. "Using gestalt information to identify locations in printed information". US20110167075 A1. 2011 (see p. 20).
- Kleffner, D. A. and V. S. Ramachandran. "On the perception of shape from shading". In: *Perception & Psychophysics* 52 1992, pp. 18–36 (see p. 29).
- Knill, D. C. and W. Richards. *Perception as Bayesian inference*. 1996 (see p. 22).

- Koenderink, J. J., A. J. Van Doorn, et al. "Affine structure from motion". In: *JOSA A* 8.2 1991, pp. 377–385 (see p. 168).
- Koffka, K. *Principles of Gestalt psychology*. Harcourt, Brace New York, 1935 (see p. 20).
- Kogo, N. et al. "Surface construction by a 2-D differentiation–integration process: A neurocomputational model for perceived border ownership, depth, and lightness in Kanizsa figures." In: *Psychological review* 117.2 2010, p. 406 (see p. 43).
- Köhler, W. *Gestalt psychology*. Liveright, 1929 (see p. 20).
- Kolmogorov, V. and R. Zabih. "Computing visual correspondence with occlusions using graph cuts". In: *IEEE ICCV*. 2001, pp. 508–515 (see p. 170).
- Kolmogorov, V. and R. Zabih. "What energy functions can be minimized via graph cuts?" In: *IEEE TPAMI* 26.2 2004, pp. 147–59 (see pp. 114–117).
- Konrad, J. and M. Ristivojevic. "Video segmentation and occlusion detection over multiple frames". In: *Image and Video Communications and Processing*. Ed. by B. Vasudev et al. Vol. 5022. 1. 2003, pp. 377–388 (see p. 170).
- Krishnamurthy, R., P. Moulin, and J. Woods. "Optical flow techniques applied to video coding". In: *IEEE ICIP*. Vol. 1. IEEE. 1995, pp. 570–573 (see p. 60).
- Kumar, M. P., B. Packer, and D. Koller. "Self-paced learning for latent variable models". In: *Advances in Neural Information Processing Systems*. 2010, pp. 1189–1197 (see p. 73).
- Lafferty, J. D., A. McCallum, and F. C. N. Pereira. "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data". In: *Int. Conf. on Machine Learning*. San Francisco, CA, USA, 2001, pp. 282–289 (see p. 90).
- Landy, M. S. et al. "Measurement and modeling of depth cue combination: In defense of weak fusion". In: *Vision research* 35.3 1995, pp. 389–412 (see pp. 33, 36).
- Lawson, C. L. and R. J. Hanson. *Solving least squares problems*. Vol. 161. SIAM, 1974 (see p. 215).
- Lee, A. B., D. Mumford, and J. Huang. "Occlusion models for natural images: A statistical study of a scale-invariant dead leaves model". In: *IEEE IJCV* 41.1-2 2001, pp. 35–59 (see pp. 43, 56, 57).
- Lee, Y. et al. "Mesh scissoring with minima rule and part salience". In: *Comput. Aided Geom. Des.* 22.5 2005, pp. 444–465 (see p. 29).
- Leichter, I. and M. Lindenbaum. "Boundary ownership by lifting to 2.1D". In: *IEEE ICCV*. 2009, pp. 9–16 (see p. 91).
- Leordeanu, M., A. Zanfir, and C. Sminchisescu. "Locally affine sparse to dense matching for motion and occlusion estimation". In: *IEEE ICCV*. 2013 (see p. 65).
- Levina, E. and P. Bickel. "The Earth Mover's distance is the Mallows distance: some insights from statistics". In: *IEEE ICCV*. Vol. 2. 2001, pp. 251–256 (see p. 45).

- Lezama, J. et al. "Track to the Future: Spatio-temporal Video Segmentation with Long-range Motion Cues". In: *IEEE CVPR*. 2011 (see pp. 170, 174, 175, 180).
- Li, P. et al. "On Creating Depth Maps from Monoscopic Video using Structure from Motion". In: *27th Symposium on Information Theory in the Benelux*. 2006, pp. 508–515 (see pp. 151, 169).
- Lindeberg, T. "Principles for automatic scale selection". In: *ISRN KTH*. 1999 (see p. 44).
- Lindeberg, T. "Junction Detection With Automatic Selection Of Detection Scales And Localization Scales". In: *IEEE ICIP*. 1994, pp. 924–928 (see pp. 27, 43, 44).
- Ling, H. and K. Okada. "Diffusion Distance for Histogram Comparison". In: *IEEE CVPR*. 2006, pp. 246–253 (see p. 101).
- Ling, H. and K. Okada. "EMD-L1: An Efficient and Robust Algorithm for Comparing Histogram-Based Descriptors". In: *IEEE ECCV*. 2006, pp. 330–343 (see p. 103).
- Liu, B., S. Gould, and D. Koller. "Single image depth estimation from predicted semantic labels". In: *IEEE CVPR*. 2010, pp. 1253–1260 (see pp. 73, 91).
- Liu, C. et al. "SIFT Flow: Dense Correspondence across Different Scenes". In: *IEEE ECCV*. ECCV '08. Marseille, France: Springer-Verlag, 2008, pp. 28–42 (see p. 60).
- Lowe, D. G. "Distinctive Image Features from Scale-Invariant Keypoints". In: *IEEE IJCV* 60.2 2004, pp. 91–110 (see p. 64).
- Lucas, B. and T. Kanade. "An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA)". In: *DARPA Image Understanding Workshop*. 1981, pp. 121–130 (see p. 60).
- Mac Aodha, O. et al. "Learning a confidence measure for optical flow." In: *IEEE TPAMI* 35.5 2013, pp. 1107–20 (see pp. 66, 176).
- Mainberger, M., A. Bruhn, and J. Weickert. "Is Dense Optic Flow Useful to Compute the Fundamental Matrix?" In: *Int. Conf. on Image Analysis and Recognition*. Berlin, Heidelberg, 2008, pp. 630–639 (see p. 168).
- Maire, M. "Simultaneous Segmentation and Figure/Ground Organization Using Angular Embedding". In: *IEEE ECCV*. 2010, pp. 450–464 (see pp. 91, 92, 124, 138).
- Maire, M. R. "Contour Detection and Image Segmentation". PhD thesis. University of California, Berkeley, 2009 (see pp. 79, 140).
- Maire, M. et al. "Using contours to detect and localize junctions in natural images". In: *IEEE CVPR*. 2008, pp. 1–8 (see pp. 42, 107).
- Malik, J. "Interpreting line drawings of curved objects". In: *IEEE IJCV* 1.1 1987, pp. 73–103 (see p. 27).
- Malik, J. and R. Rosenholtz. "Computing Local Surface Orientation and Shape from Texture for Curved Surfaces". In: *IEEE IJCV* 23.2 1997, pp. 149–168 (see p. 31).

- Mamassian, P. and M. S. Landy. "Interaction of visual prior constraints." In: *Vision research* 41.20 2001, pp. 2653–68 (see p. 20).
- Mamassian, P. "Bayesian inference of form and shape." In: *Progress in brain research* 154 2006, pp. 265–70 (see p. 20).
- Manjunath, B. S. et al. "Color and Texture Descriptors". In: *IEEE Trans. on Circuits and Systems for Video Technology* 11 1998, pp. 703–715 (see p. 98).
- Marr, D. *Vision : a computational investigation into the human representation and processing of visual information*. 1982, pp. xvii, 397 (see p. 21).
- Martin, D. R., C. C. Fowlkes, and J. Malik. "Learning to detect natural image boundaries using local brightness, color, and texture cues". In: *IEEE TPAMI* 26.5 2004, pp. 530–549 (see pp. 42, 78–80, 107, 120).
- Martin, D. et al. "A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics". In: *IEEE ICCV*. Vol. 2. 2001, pp. 416–423 (see pp. 51, 54, 89, 106).
- McDermott, J. "Psychophysics with junctions in real images". In: *Perception* 33.9 2004, pp. 1101–1127 (see pp. 13, 27, 44, 48, 52, 145).
- Meinhardt-Llopis, E. et al. "Relative depth from monocular optical flow". In: *IEEE ICIP*. IEEE. 2011, pp. 2085–2088 (see p. 163).
- Miled, W., B. Pesquet-Popescu, and W. Cherif. "A variational framework for simultaneous motion and disparity estimation in a sequence of stereo images". In: *IEEE ICASSP*. 2009, pp. 741–744 (see p. 206).
- Möbius, A. F. *Der Barycentrische Calcul : ein neues Hülfsmittel zur analytischen Behandlung der Geometrie*. MacTutor History of Mathematics, 1827 (see p. 203).
- Moore, A. P. et al. "Superpixel lattices". In: *IEEE CVPR*. IEEE. 2008, pp. 1–8 (see p. 186).
- Moreno-Bote, R., D. C. Knill, and A. Pouget. "Bayesian sampling in visual perception". In: *Proceedings of the National Academy of Sciences* 108.30 2011, pp. 12491–12496 (see p. 22).
- Mumford, D. and J. Shah. "Optimal approximations by piecewise smooth functions and associated variational problems". In: *Comm. on Pure and Applied Mathematics* 42.5 1989, pp. 577–685 (see pp. 63, 103, 119, 125).
- Nagata, S. "Pictorial communication in virtual and real environments". In: ed. by S. R. Ellis. Bristol, PA, USA: Taylor & Francis, Inc., 1991. Chap. How to reinforce perception of depth in single two-dimensional pictures, pp. 527–545 (see pp. 18, 41).
- Najman, L. and M. Schmitt. "Watershed of a continuous function". In: *Signal Processing* 38.1 1994, pp. 99–112 (see p. 108).

- Nakayama, K. and G. H. Silverman. "The aperture problem I. Perception of nonrigidity and motion direction in translating sinusoidal lines". In: *Vision Research* 28.6 1988, pp. 739–746 (see pp. 61, 174, 176).
- Nawrot, M. and K. Stroyan. "The motion/pursuit law for visual depth perception from motion parallax". In: *Vision Research* 49.15 2009, pp. 1969–1978 (see p. 35).
- Nistér, D. "Calibration with robust use of cheirality by quasi-affine reconstruction of the set of camera projection centres". In: *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*. Vol. 2. IEEE. 2001, pp. 116–123 (see p. 214).
- Ochs, P. and T. Brox. "Object segmentation in video: A hierarchical variational approach for turning point trajectories into dense regions". In: *IEEE ICCV*. 2011, pp. 1583–1590 (see pp. 174, 191).
- Ogale, A. S., C. Fermuller, and Y. Aloimonos. "Motion segmentation using occlusions". In: *IEEE TPAMI* 27.6 2005, pp. 988–992 (see p. 174).
- Okoshi, T. *Three-dimensional imaging techniques*. Academic Press, 1976 (see p. 32).
- Ono, M. E., J. Rivest, and H. Ono. "Depth perception as a function of motion parallax and absolute-distance information." In: *Journal of experimental psychology. Human perception and performance* 12.3 1986, pp. 331–7 (see pp. 13, 35, 149).
- Orchard, M. T. and C. A. Bouman. "Color quantization of images". In: *IEEE Trans. on Signal Processing* 39.12 1991, pp. 2677–2690 (see p. 56).
- Osher, S. and N. Paragios. *Geometric Level Set Methods in Imaging, Vision, and Graphics*. Secaucus, NJ, USA, 2003 (see p. 49).
- Pardas, M. et al. "Partition tree for a segmentation-based video coding system". In: *IEEE ICASSP*. Washington, DC, USA, 1996, pp. 1982–1985 (see p. 95).
- Paris, S. and F. Durand. "A Topological Approach to Hierarchical Segmentation using Mean Shift". In: *IEEE CVPR*. 2007, pp. 1–8 (see pp. 95, 172, 180).
- Paris, S. "Edge-Preserving Smoothing and Mean-Shift Segmentation of Video Streams". In: *IEEE ECCV*. 2008, pp. 460–473 (see p. 172).
- Paschos, G. "Perceptually uniform color spaces for color texture analysis: an empirical evaluation". In: *IEEE Trans. on Image Processing* 10.6 2001, pp. 932–937 (see p. 97).
- Pentland, A. P. "A New Sense for Depth of Field". In: *IEEE TPAMI PAMI-9.4* 1987, pp. 523–531 (see p. 88).
- Perona, P. and J. Malik. "Scale-space and edge detection using anisotropic diffusion". In: *IEEE TPAMI* 12.7 1990, pp. 629–639 (see pp. 49, 217).

- Phan, R., R. Rzeszutek, and D. Androutsos. "Semi-automatic 2D to 3D image conversion using a hybrid Random Walks and graph cuts based approach". In: *IEEE ICASSP*. IEEE. 2011, pp. 897–900 (see pp. 41, 87, 146).
- Pock, T. et al. "An algorithm for minimizing the Mumford-Shah functional". In: *IEEE ICCV*. 2009, pp. 1133–1140 (see p. 119).
- Pollefeys, M. "Self-calibration and metric 3D reconstruction from uncalibrated image sequences". PhD thesis. ESAT-PSI K.U.Leuven, 1999 (see p. 169).
- Pollefeys, M. et al. "Visual modeling with a hand-held camera". In: *IEEE IJCV* 59.3 2004, pp. 207–232 (see pp. 169, 201, 213, 214).
- Ponce, J., D. Forsyth, et al. "Computer vision: a modern approach". In: *Computer* 16 2011, p. 11 (see p. 169).
- Ponce, J., K. McHenry, et al. "On the absolute quadratic complex and its application to autocalibration". In: *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*. Vol. 1. IEEE. 2005, pp. 780–787 (see p. 212).
- Pont-Tuset, J. and F. Marqués. "Supervised Assessment of Segmentation Hierarchies". In: *IEEE ECCV*. 2012 (see p. 114).
- Pont-Tuset, J. and F. Marqués. "Measures and Meta-Measures for the Supervised Evaluation of Image Segmentation". In: *IEEE CVPR*. 2013 (see pp. 121, 123, 138).
- Ponzo, M. "Intorno ad alcune illusioni nel campo delle sensazioni tattili, sull'illusione di Aristotele e fenomeni analoghi". In: *Archiv für die gesamte Psychologie* 16 1910, pp. 307–345 (see p. 24).
- Prados, E. and O. Faugeras. "Shape From Shading". In: *Handbook of Mathematical Models in Computer Vision*. 2006. Chap. 23, pp. 375–388 (see p. 30).
- Pratt, V. R. "Semantical consideration on floyo-hoare logic". In: *Foundations of Computer Science*. IEEE. 1976, pp. 109–121 (see p. 134).
- Puzicha, J., T. Hofmann, and J. M. Buhmann. "Non-parametric similarity measures for unsupervised texture segmentation and image retrieval". In: *IEEE CVPR*. 1997, pp. 267–272 (see p. 100).
- Qiu, F. T. and R. Von Der Heydt. "Figure and ground in the visual cortex: V2 combines stereoscopic cues with Gestalt rules". In: *Neuron* 47.1 2005, pp. 155–166 (see p. 33).
- Rao, S. R., R. Tron, and R. Vidal. "Motion segmentation via robust subspace separation in the presence of outlying, incomplete, or corrupted trajectories". In: *IEEE CVPR*. 2008, pp. 1–8 (see p. 190).
- Rao, S. et al. "Motion Segmentation in the Presence of Outlying, Incomplete, or Corrupted Trajectories". In: *IEEE TPAMI* 32.10 2010, pp. 1832–1845 (see pp. 173, 174, 180).

- Ren, X., C. C. Fowlkes, and J. Malik. "Figure/Ground Assignment in Natural Images". In: *IEEE ECCV*. Ed. by A. Leonardis, H. Bischof, and A. Pinz. Vol. 3952. Berlin, Heidelberg, 2006, pp. 614–627 (see pp. 33, 73, 78, 90, 91).
- Robertson, A. R. "Historical development of CIE recommended color difference equations". In: *Color Research & Application* 15.3 1990, pp. 167–170 (see pp. 44, 98).
- Rogers, B. and M. Graham. "Motion parallax as an independent cue for depth perception". In: *Perception* 8.2 1979, pp. 125–134 (see p. 59).
- Rogers, B. and M. Graham. "Similarities between motion parallax and stereopsis in human depth perception". In: *Vision research* 22.2 1982, pp. 261–270 (see p. 35).
- Ross, M. G. and A. Oliva. "Estimating perception of scene layout properties from global image features". In: *Journal of Vision* 10.1 2010 (see p. 89).
- Rubin, N. e. a. "The role of junctions in surface completion and contour matching". In: *Perception London* 30.3 2001, pp. 339–366 (see pp. 27, 28).
- Rubner, Y., C. Tomasi, and L. J. Guibas. "A metric for distributions with applications to image databases". In: *IEEE ICCV*. 1998, pp. 59–66 (see p. 101).
- Rudin, L. I., S. Osher, and E. Fatemi. "Nonlinear total variation based noise removal algorithms". In: *Physica D: Nonlinear Phenomena* 60.1-4 1992, pp. 259–268 (see p. 63).
- Ruzon, M. A. and C. Tomasi. "Edge, Junction, and Corner Detection Using Color Distributions". In: *IEEE TPAMI* 23.11 2001, pp. 1281–1295 (see pp. 43, 45, 46, 98, 99, 101, 156).
- Salembier, P. and L. Garrido. "Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval". In: *IEEE Trans. on Image Processing* 9.4 2000, pp. 561–576 (see pp. 95, 97, 117, 118, 175).
- Saxena, A., A. Ng, and S. Chung. "Learning Depth from Single Monocular Images". In: *IEEE NIPS* 18 2005 (see pp. 20, 37, 91, 92, 138, 143).
- Scharstein, D. and R. Szeliski. "A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms". In: *IEEE IJCV* 47.1-3 2002, pp. 7–42 (see p. 206).
- Schwartz, S. H. *Visual Perception: A Clinical Orientation*. 3rd ed. 2004 (see p. 33).
- Seitz, S. M. et al. "A comparison and evaluation of multi-view stereo reconstruction algorithms". In: *Computer vision and pattern recognition, 2006 IEEE Computer Society Conference on*. Vol. 1. IEEE. 2006, pp. 519–528 (see pp. 219, 221).
- Serra, J., B. Kiran, and J. Cousty. "Hierarchies and Climbing Energies". In: *Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications*. Vol. 7441. 2012, pp. 821–828 (see pp. 97, 115, 118).
- Sharma, G. *Digital Color Imaging Handbook*. Boca Raton, FL, USA, 2002 (see pp. 97, 98).

- Shepard, R. N. "Toward a Universal Law of Generalization for Psychological Science". In: *Science* 237.4820 1987, pp. 1317–1323 (see pp. 99, 156).
- Shi, J. and C. Tomasi. "Good features to track". In: *IEEE CVPR*. 1994, pp. 593–600 (see pp. 167, 174).
- Shi, J. and J. Malik. "Normalized Cuts and Image Segmentation". In: *IEEE TPAMI* 22.8 2000, pp. 888–905 (see pp. 91, 107, 174).
- Siddiqi, K., K. J. Tresness, and B. B. Kimia. "Parts of visual form: psychophysical aspects." In: *Perception* 25.4 1996, pp. 399–424 (see p. 29).
- Slugocki, M. et al. "Convexity as a Cue to Figure-Ground Segmentation in Children". In: *Journal of Vision* 13.9 2013, pp. 718–718 (see p. 28).
- Smith, P., T. Drummond, and R. Cipolla. "Layered Motion Segmentation and Depth Ordering by Tracking Edges". In: *IEEE TPAMI* 26.4 2004, pp. 479–494 (see p. 170).
- Smith, S. M. and J. M. Brady. "SUSAN-A New Approach to Low Level Image Processing". In: *IEEE IJCV* 23.1 1997, pp. 45–78 (see p. 42).
- Sperling, G. and B. A. Doshier. "Depth from motion". In: *Early vision and beyond* 1994, pp. 133–142 (see p. 36).
- Stein, A. "Occlusion Boundaries: Low-Level Detection to High-Level Reasoning". PhD thesis. Pittsburgh, PA: Robotics Institute, Carnegie Mellon University, 2008 (see p. 163).
- Stone, J. V. "Footprints sticking out of the sand. Part 2: children's Bayesian priors for shape and lighting direction." In: *Perception* 40.2 2011, pp. 175–90 (see p. 20).
- Stone, J. V. and O. Pascalis. "Footprints sticking out of the sand. Part 1: Children's perception of naturalistic and embossed symbol stimuli." In: *Perception* 39.9 2010, pp. 1254–60 (see p. 20).
- Stühmer, J., S. Gumhold, and D. Cremers. "Real-time dense geometry from a handheld camera". In: *DAGM Conf. on Pattern recognition*. Berlin, Heidelberg, 2010, pp. 11–20 (see p. 169).
- Sturm, P. "Critical motion sequences for monocular self-calibration and uncalibrated Euclidean reconstruction". In: *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*. IEEE. 1997, pp. 1100–1105 (see p. 208).
- Sun, D., S. Roth, and M. J. Black. "Secrets of optical flow estimation and their principles". In: *IEEE CVPR*. 2010, pp. 2432–2439 (see pp. 65, 174, 221).
- Sundaram, N., T. Brox, and K. Keutzer. "Dense point trajectories by GPU-accelerated large displacement optical flow". In: *IEEE ECCV*. 2010 (see pp. 60, 174–178).

- Sundaram, N. and K. Keutzer. "Long term video segmentation through pixel level spectral clustering on GPUs." In: *IEEE ICCV Workshops*. 2011, pp. 475–482 (see pp. 174, 191).
- Sundberg, P. et al. "Occlusion boundary detection and figure/ground assignment from optical flow". In: *IEEE CVPR*. 2011 (see pp. 151, 153, 154, 156, 160, 163, 179, 188).
- Super, B. J. and A. C. Bovik. "Planar surface orientation from texture spatial frequencies". In: *Pattern Recognition* 28.5 1995, pp. 729–743 (see p. 88).
- Szeliski, R. "Video mosaics for virtual environments". In: *Computer Graphics and Applications, IEEE* 16.2 1996, pp. 22–30 (see p. 168).
- Telea, A. "An image inpainting technique based on the fast marching method". In: *Journal of graphics tools* 9.1 2004, pp. 23–34 (see p. 217).
- Terruggia, R. "Reliability Analysis of Probabilistic Networks". PhD thesis. Universita degli Studi di Torino, 2010 (see pp. 129–131, 160).
- Thormählen, T., H. Broszio, and P. Mikulastik. "Robust linear auto-calibration of a moving camera from image sequences". In: *Computer Vision–ACCV 2006*. Springer, 2006, pp. 71–80 (see p. 214).
- Tomasi, C. and T. Kanade. "Shape and motion from image streams under orthography: a factorization method". In: *Int. J. Comput. Vision* 9.2 1992, pp. 137–154 (see p. 206).
- Torr, P. H. S., R. Szeliski, and P. Anandan. "An integrated Bayesian approach to layer extraction from image sequences". In: *IEEE ICCV*. Vol. 2. 1999, 983–990 vol.2 (see p. 170).
- Torralba, A. and A. Oliva. "Depth estimation from image structure". In: *IEEE TPAMI* 24 2002, p. 2002 (see p. 89).
- Tremeau, A. and P. Colantoni. "Regions adjacency graph applied to color image segmentation". In: *IEEE Trans. on Image Processing* 9.4 2000, pp. 735–744 (see p. 95).
- Triggs, B. "Factorization methods for projective structure and motion". In: *IEEE CVPR*. IEEE. 1996, pp. 845–851 (see pp. 168, 207).
- Triggs, B. "Autocalibration and the absolute quadric". In: *IEEE CVPR*. IEEE. 1997, pp. 609–614 (see p. 212).
- Triggs, B. et al. "Bundle Adjustment – A Modern Synthesis". In: *Vision Algorithms: Theory and Practice*. Ed. by B. Triggs, A. Zisserman, and R. Szeliski. Vol. 1883. 2000, pp. 298–372 (see pp. 208–210).
- Turetken, E. and A. A. Alatan. "Temporally consistent layer depth ordering via pixel voting for pseudo 3D representation". In: *3DTV Conference*. 2009, pp. 1–4 (see pp. 150, 170).

- Unnikrishnan, R., C. Pantofaru, and M. Hebert. "Toward objective evaluation of image segmentation algorithms." In: *IEEE TPAMI* 29.6 2007, pp. 929–44 (see p. 184).
- Van den Berg, A. and E. Brenner. "Humans combine the optic flow with static depth cues for robust perception of heading". In: *Vision research* 34.16 1994, pp. 2153–2167 (see p. 33).
- Van Sijll, J. *Cinematic Storytelling: The 100 Most Powerful Film Conventions Every Filmmaker Must Know*. Michael Wiese Productions, Aug. 25, 2005 (see p. 41).
- Vecera, S. P., E. K. Vogel, and G. F. Woodman. "Lower region: a new cue for figure-ground assignment." In: *Journal of Experimental Psychology: General* 131.2 2002, p. 194 (see p. 91).
- Vidal, R. et al. "Two-view multibody structure from motion". In: *IEEE IJCV* 68.1 2006, pp. 7–25 (see p. 171).
- Vilaplana, V., F. Marques, and P. Salembier. "Binary Partition Trees for Object Detection". In: *IEEE Trans. on Image Processing* 17.11 2008, pp. 2201–2216 (see pp. 98, 103, 119, 179, 184).
- Volz, S. et al. "Modeling temporal coherence for optical flow". In: *IEEE ICCV*. IEEE. 2011, pp. 1116–1123 (see p. 63).
- Von Helmholtz, H. *Handbuch der physiologischen Optik*. Vol. 9. 1866 (see pp. 13, 19, 30).
- Walker, L. L. and J. Malik. "Can convexity explain how humans segment objects into parts?" In: *Journal of Vision* 3.9 2003, p. 503 (see p. 29).
- Wang, O. et al. "StereoBrush: interactive 2D to 3D conversion using discontinuous warps". In: *Proceedings of the Eighth Eurographics Symposium on Sketch-Based Interfaces and Modeling*. New York, NY, USA, 2011, pp. 47–54 (see pp. 87, 150).
- Ward, B., S. Bing Kang, and E. P. Bennett. "Depth Director: A System for Adding Depth to Movies". In: *Computer Graphics and Applications, IEEE* 31.1 2011, pp. 36–48 (see pp. 87, 150).
- Wedel, A., T. Pock, et al. "Duality TV-L1 flow with fundamental matrix prior". In: *Int. Conf. on Image and Vision Computing*. IEEE. 2008, pp. 1–6 (see p. 168).
- Wedel, A., A. Meißner, et al. "Detection and Segmentation of Independently Moving Objects from Dense Scene Flow". In: *EMMCVPR*. Vol. 5681. 2009, pp. 14–27 (see p. 174).
- Weng, J., P. Cohen, and M. Herniou. "Camera calibration with distortion models and accuracy evaluation". In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 14.10 1992, pp. 965–980 (see p. 215).
- Werlberger, M. et al. "Anisotropic Huber-L1 Optical Flow". In: *BMVC*. 2009, pp. 1–11 (see p. 63).

- Wertheimer, M. *Gestalt theory*. Hayes Barton Press, 1938 (see p. 20).
- Wirjadi, O. *Survey of 3D image segmentation methods*. Tech. rep. 123. Fraunhofer Institut fur Techno und Wirtschaftsmathematik, 2007 (see p. 173).
- Wismeijer, D., R. van Ee, and C. J. Erkelens. "Depth cues, rather than perceived depth, govern vergence". In: *Experimental Brain Research* 184.1 2008, pp. 61–70 (see p. 25).
- Wurtz, R., K. H, and E. R. "Central visual pathways". In: *Principles of Neural Science* 4 2000, pp. 523–545 (see p. 19).
- Xiao, J., H. Cheng, et al. "Bilateral filtering-based optical flow estimation with occlusion detection". In: *IEEE ECCV*. 2006, pp. 211–224 (see p. 66).
- Xiao, J. and M. Shah. "Motion layer extraction in the presence of occlusion using graph cuts". In: *IEEE TPAMI* 27.10 2005, pp. 1644–1659 (see p. 174).
- Xiao, J., J. Hays, et al. "Sun database: Large-scale scene recognition from abbey to zoo". In: *IEEE CVPR*. IEEE. 2010, pp. 3485–3492 (see p. 93).
- Xu, C., S. Whitt, and J. J. Corso. "Flattening Supervoxel Hierarchies by the Uniform Entropy Slice". In: *IEEE ICCV*. 2013 (see p. 114).
- Xu, C. and J. J. Corso. "Evaluation of super-voxel methods for early video processing". In: *IEEE CVPR*. 2012, pp. 1202–1209 (see pp. 180, 185).
- Xu, L., J. Chen, and J. Jia. "A segmentation based variational model for accurate optical flow estimation". In: *IEEE ECCV*. Vol. 1. Citeseer. 2008, pp. 671–684 (see p. 60).
- Xu, L., J. Jia, and Y. Matsushita. "Motion detail preserving optical flow estimation". In: *IEEE CVPR*. 2010, pp. 1293–1300 (see p. 63).
- Xu, Y., T. Géraud, and L. Najman. "Morphological filtering in shape spaces: Applications using tree-based image representations". In: *Int. Conf. on Pattern Recognition*. IEEE. 2012, pp. 485–488 (see p. 97).
- Yan, J. and M. Pollefeys. "A General Framework for Motion Segmentation: Independent, Articulated, Rigid, Non-rigid, Degenerate and Non-degenerate". In: *IEEE ECCV*. 2006, pp. 94–106 (see pp. 169, 174).
- Yu, S. X. "Angular embedding: From jarring intensity differences to perceived luminance". In: *IEEE CVPR*. 2009, pp. 2302–2309 (see p. 91).
- Zhang, G., J. Jia, T. T. Wong, et al. "Recovering consistent video depth maps via bundle optimization". In: *IEEE TPAMI* 2008 (see p. 168).
- Zhang, G., J. Jia, W. Hua, et al. "Robust bilayer segmentation and motion/depth estimation with a handheld camera." In: *IEEE TPAMI* 33.3 2011, pp. 603–17 (see pp. 37, 171).
- Zhang, G., J. Jia, T.-T. Wong, et al. "Consistent Depth Maps Recovery from a Video Sequence". In: *IEEE TPAMI* 31.6 2009, pp. 974–988 (see pp. 151, 168, 219, 220).

- Zhang, R. et al. "Shape-from-shading: a survey". In: *IEEE TPAMI* 21.8 1999, pp. 690–706 (see pp. 29, 87).
- Zhang, Z. "Determining the epipolar geometry and its uncertainty: A review". In: *IEEE IJCV* 27.2 1998, pp. 161–195 (see p. 207).
- Zhao, L. and L. S. Davis. "Iterative figure-ground discrimination". In: *Int. Conf. on Pattern Recognition*. Vol. 1. IEEE. 2004, pp. 67–70 (see p. 90).
- Zhuo, S. and T. Sim. "On the recovery of depth from a single defocused image". In: *Computer Analysis of Images and Patterns*. Springer. 2009, pp. 889–897 (see p. 88).