UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Universitat Politècnica de Catalunya
Teoria del Senyal i Comunicacions

This thesis is submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy (PhD)

Visual Object Analysis using Regions and Local Features

by Carles Ventura Royo

Tutor: Prof. Ferran Marques Acosta
Co-advisor: Prof. Xavier Giró Nieto
Co-advisor: Prof. Verónica Vilaplana Besler

Barcelona, April 2016

Abstract

The first part of this dissertation focuses on an analysis of the spatial context in semantic image segmentation. First, we review how spatial context has been tackled in the literature by local features and spatial aggregation techniques. From a discussion about whether the context is beneficial or not for object recognition, we extend a Figure-Border-Ground segmentation for local feature aggregation with ground truth annotations to a more realistic scenario where object proposals techniques are used instead. Whereas the Figure and Ground regions represent the object and the surround respectively, the Border is a region around the object contour, which is found to be the region with the richest contextual information for object recognition. Furthermore, we propose a new contour-based spatial aggregation technique of the local features within the object region by a division of the region into four subregions. Both contributions have been tested on a semantic segmentation benchmark with a combination of free and non-free context local features that allows the models automatically learn whether the context is beneficial or not for each semantic category.

The second part of this dissertation addresses the semantic segmentation for a set of closely-related images from an uncalibrated multiview scenario. State-of-the-art semantic segmentation algorithms fail on correctly segmenting the objects from some viewpoints when the techniques are independently applied to each viewpoint image. The lack of large annotations available for multiview segmentation do not allow to obtain a proper model that is robust to viewpoint changes. In this second part, we exploit the spatial correlation that exists between the different viewpoints images to obtain a more robust semantic segmentation. First, we review the state-of-the-art co-clustering, co-segmentation and video segmentation techniques that aim to segment the set of images in a generic way, i.e. without considering semantics. Then, a new architecture that considers motion information and provides a multiresolution segmentation is proposed for the co-clustering framework and outperforms state-of-the-art techniques for generic multiview segmentation. Finally, the proposed multiview segmentation is combined with the semantic segmentation results giving a method for automatic resolution selection and a coherent semantic multiview segmentation.

Acknowledgments

I would like to express my gratitude to my tutor, Prof. Ferran Marques, and my co-advisors, Prof. Xavier Giró and Prof. Verónica Vilaplana, for our enriching discussions, for their support in the weak moments, for rowing in the same direction despite their personal interests and for the working environment they contribute to create. I would also give special thanks to Dr. Jordi Pont, who was co-advisor of my undergraduate thesis project and has been as a mentor during all the PhD.

A very special thanks goes to my colleagues, and friends, at D5-120, for our philosophical debates, baking competitions, exchanges of knowledge and skills, chocolates and cookies when deadlines were approaching, and for making the office a pleasant and comfortable place to work.

I must also acknowledge Albert and Josep, from whom I have learnt many things and discovered many tools; for setting up an awesome technical environment for princesses so we had to think only about research.

My sincerest gratitude goes also to Prof. Noel O'Connor and Dr. Kevin McGuinness, for the opportunity they granted me to collaborate with their group at the Dublin City University.

I would also like to thank my parents and the rest of my family for the support, love, guidance, and happiness they provided me not only through my thesis, but my entire life.

Last, but not least, I would like to say thanks to Cristina, my wife, for putting up with me, my deadlines, my thesis,... for making life so easy and so enjoyable, for being there, no matter what.

I also acknowledge that this thesis would not have been possible without the financial assistance of the Image Processing Group (GPI) at UPC, the Catalan Governement through the FI-AGAUR grant, the Spanish Government through the FPU grant and the projects CENIT-2009-1026 BuscaMedia and BIGGRAPH- TEC2013-43935-R. I would also thank the Universitat Oberta de Catalunya for giving me the opportunity to continue my academic career and the facilities given to write this dissertation in my research time.

Contents

Introduction

1

Humans have no difficulty in identifying the different elements that form the scene that they are visualizing. Not only they are able to separate one object from the other ones and recognize each of them, but also to understand the whole picture. In computer vision, the task of detecting the different objects belonging to such a scene is known as object detection, whereas the task of labeling each of them, e.g. as an aeroplane or a car, is referred to as object recognition. Semantic segmentation combines both previous tasks since aims at segmenting an image into regions and recognizing each of them as semantic classes. Whereas semantic segmentation is generally unaware of individual object instances, instance-aware semantic segmentation tries to identify the different object instances and predict a category label for each of them. One step further is the scene understanding task, which aims at giving an interpretation of what is happening in the scene, e.g. a football player who has scored a goal.

The scope of this dissertation is the semantic segmentation problem, where despite the huge progress the field has experienced with deep learning techniques in the last few years, we consider there is still room for improvement. This dissertation has been performed in a period of fast changes in the state-of-the-art techniques. As a consequence, while the first part of this dissertation is based on a more classic approach, where semantic segmentation problem is addressed with handcrafted local features, e.g. the popular SIFT descriptor, the second part abandons this classic approach and takes advantage of learned local features from state-of-the-art techniques based on convolutional neural networks.

The first part of this dissertation analyzes the impact of the context and the spatial codification in object recognition for a semantic segmentation problem. A variation of SIFT, called Masked SIFT, which only describes the object itself without being affected by the context, is used in the experiments to analyze the influence of the context. Furthermore, a richer spatial codification of the image for visual descriptors aggregation is also analyzed. Beyond the classic Figure-Ground segmentation for visual descriptors aggregation, we propose to also consider a region around the object called Border, which represents the closest context of the object. In addition to that, a richer spatial codification for the object is also proposed by dividing it into four regions over which the visual descriptors are aggregated.

Whereas the first part of this dissertation focuses on the semantic segmentation problem for independent images, e.g. images do not have any spatial or temporal correlation, the second part extends the semantic segmentation problem for uncalibrated multiview datasets. The lack of uniformity in the viewpoint distribution of the annotated datasets results in models for object recognition with a performance that changes depending on the viewpoint of the object being recognized. However, in such a scenario, semantic

segmentation techniques can take advantage of the spatial correlation existing between the different viewpoints and of the co-clustering and co-segmentation techniques that exploit such correlation. At this point, due to the outperformance of the off-the-shelf deep learning features with respect to hand-crafted descriptors, these recent learned representations are used in the second part of this dissertation to obtain an independent semantic segmentation for each viewpoint image. However, up to the author's knowledge, at the time of writing this dissertation there was no dataset with multiview annotations large enough to train an end-to-end solution. In this case, this problem requires a more hand crafted approach as proposed in this dissertation.

Basic elements in image representation

2

In this chapter, we give an overview of different techniques for image representation which will be used for the semantic segmentation problem along this dissertation. First, in Section 2.1, we introduce the concept of hierarchical segmentation and one of the techniques that we will use in both parts of this dissertation: the Ultrametric Contour Map (UCM). Then, in Section 2.2, two techniques of object proposals are overviewed: the Constrained Parametric Min-Cuts (CPMC) and the Multiscale Combinatorial Grouping (MCG). Both techniques will be compared in the first part of the dissertation about context analysis. Finally, a set of local features (SIFT, LBP and HOG) are reviewed in Section 2.3 and two different techniques of feature aggregation over a region (Bag-of-Features and Second Order Pooling) in Section 2.4. The local features will be used in both parts, whereas the feature aggregation techniques will be only considered in Part I.

2.1 Hierarchical segmentation algorithms

Hierarchical segmentation algorithms provide segmentation of images into regions at multiple resolutions. Given an initial oversegmentation $P^{(0)}$, hierarchical segmentation algorithms provide an order of mergings of these regions resulting into increasingly coarser partitions $P^{(1)}$, $P^{(2)}$, ..., $P^{(i)}$, ..., $P^{(N-1)}$, where N is the number of regions in $P^{(0)}$. The order of the mergings depends on a similarity criteria that measures how similar two regions are. Regions are merged following this criteria so the most similar regions from $P^{(i-1)}$ are those ones to be merged to create the partition $P^{(i)}$. Although the previous definition assumes binary mergings, i.e. mergings of region pairs, can be generalized to any d-ary mergings since any hierarchy can be described as a binary one.

The increasingly coarser partitions $\{P^{(i)}\}_{i=0}^{N-1}$ resulting from binary mergings can be represented as a tree which is referred to as Binary Partition Tree (BPT) [SG00]. This tree consists of a set of nodes such that each node represents one region from hierarchical partition. There are two kind of nodes: the internal or parent nodes and the leaf nodes. On the one hand, leaf nodes represent the regions from the initial partition $P^{(0)}$. On the other hand, internal or parent nodes represents the region that results from the merging of the two regions represented by their two sibling nodes.

The motivation for the use of hierarchical segmentation algorithms in this dissertation is twofold. First, in Part I, MCG object candidates (see Section 2.2.2) result from a combination of at most 4 nodes from the BPT representing the hierarchical partition. The use of partitions at multiple resolutions allows that the nodes being combined may come from different resolutions. These object location hypotheses will be used in a realistic scenario for the experiments as well as CPMC object candidates (see Section 2.2.1), which are not related with hierarchical segmentation algorithms. Second, in Part II, where the

segmentation of a set of closely-related images is addressed in a uncalibrated multiview scenario, hierarchical segmentation algorithms are used to force that hierarchies obtained for each image are preserved when a coherent multiview segmentation is considered.

Next, we give an overview of one of the state-of-art algorithms that will be used in this dissertation: the gPb-owt-ucm [AMFM11], which will be referred to as UCM along the dissertation for brevity. The first stage of this hierarchical segmentation algorithm is the gPb, a contour detector that will be also used in Part II.

2.1.1 The gPb-owt-ucm segmentation algorithm

The gPb-owt-ucm [AMFM11] is among the state-of-art segmentation algorithms. It is the algorithm that shows the best performance when evaluated on the Berkeley Segmentation Dataset (BSDS) [MFTM01] benchmark. The gPb-owt-ucm consists of 3 different blocks: (*i*) the gPb contour detector, (*ii*) the Oriented Watershed Transform (OWT), and (*iii*) the Ultrametric Contour Map (UCM).

First, the gPb contour detector aims at obtaining a set of images where the pixels of each image represent the boundary strength at a given orientation. So, each image in the collection is associated to a certain orientation. With this goal, it couples multiscale local brightness, color and texture cues to a powerful globalization framework using spectral clustering.

Then, the Oriented Watershed Transform (OWT) constructs a set of initial regions from the oriented contour signal $gPb(x, y, \theta)$. The Watershed Transform [BM92] is equivalent to placing a water source in each regional minimum, flooding the relief from sources, and building barriers when different sources are meeting. However, applying standard watershed transform [BM92] over a non-oriented contour signal $gPb(x, y)$, which is computed as $\max_\theta gPb(x, y, \theta)$, can produce artifacts since, for instance, horizontal watershed arcs near strong vertical contours are erronously upweighted due to a high magnitude of $gPb(x, y)$. To correct this problem, Oriented Watershed Transform enforces consistency between the strength of the boundaries and the underlying oriented contour signal. This is done by estimating the orientation $o(x, y)$ at each pixel on an arc from the local geometry of the arc itself and assigning each arc pixel a boundary strength of $gPb(x, y, o(x, y))$ instead of $gPb(x, y)$.

Finally, the Ultrametric Contour Map is a hierarchical region tree which results from an agglomerative clustering by iteratively merging the most similar regions, i.e. the two adjacent regions which are separated by the minimum weight contour. As a result, the base level of this hierarchy respects the weak contours and tends to correspond to a semantic oversegmentation of the image, whereas the upper levels respect only strong contours, resulting in a semantic undersegmentation. Figure 2.1 shows an example of the whole gPb-owt-ucm segmentation algorithm.

2.2 Object proposals techniques

Object proposals techniques are class-independent methods that generate object hypotheses or candidates in image areas where it is likely to represent an object. These techniques can be divided into those whose output is an image window and those that generate segmented candidates. In this dissertation we focus on segmented candidates since this kind

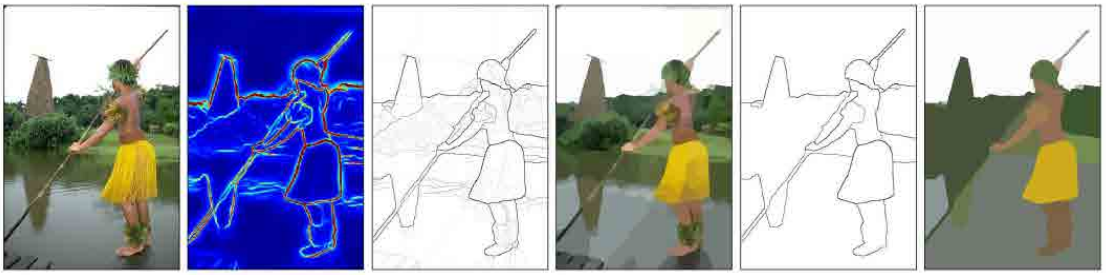


Figure 2.1: Hierarchical segmentation from contours. Figure taken from [AMFM11]. From left to right: image, maximal response of gPb over orientations, weighted contours resulting from OWT using gPb as input, initial oversegmentation resulting from OWT and corresponding to the finest level of UCM, contours obtained by thresholding UCM at level 0.5, and segmentation obtained by thresholding UCM at level 0.5.

of object candidates allows us to perform a more accurate analysis of the context in Part I, not including part of the context in the object candidate as it would happen with image windows.

The use of object candidates will be essential in Part I to extend the method of aggregating local features at three different areas of the image: the object, the object contour and the surround [USS12]. Whereas such a method was applied to ground truth object annotations, we propose to use object candidates to extend the method to a more realistic scenario where object locations are not provided. Next, we give an overview of the two object proposal techniques that will be used in the first part of this dissertation.

2.2.1 Constrained Parametric Min-Cuts (CPMC)

In [CS12], a rank list of plausible objects hypotheses is generated by solving a sequence of constrained parametric min-cut problems (CPMC) on a regular image grid. These object hypotheses are obtained using bottom-up processes and mid-level cues, without prior knowledge about properties of individual object classes. Each object hypothesis is represented as a figure-ground segmentation. The objective is to minimize over the pixel labels (foreground or background) an energy function that depends on a unary term and a pairwise term. The unary term measures the probability of the pixel to be part of the foreground (without considering the neighbor pixels) whereas the pairwise term promotes that similar neighbor pixels have the same pixel labels. Figure 2.2 shows an example of a rank list of object candidates resulting from applying CPMC to an image.

2.2.2 Multiscale Combinatorial Grouping (MCG)

The Multiscale Combinatorial Grouping (MCG) [APTB+14] is a unified framework for hierarchical image segmentation and object proposal generation. Regarding the hierarchical segmentation, MCG is a more efficient algorithm to compute the gPb-owt-ucm overviewed in Section 2.1. Regarding the object candidates, the authors consider the singletons, pairs, triplets and 4-tuples of regions from the hierarchical partition. Since the full set of candidates results in millions of candidates, they are reduced to thousand of candidates in a learning problem known as *Pareto front optimization* while keeping

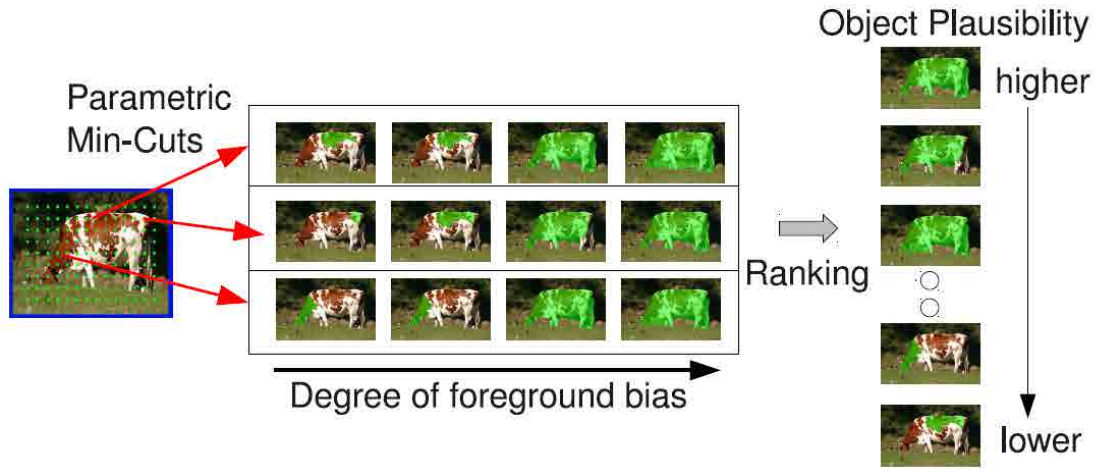


Figure 2.2: Figure taken from [CS12]. Segments are extracted around regularly placed foreground seeds and ranked according to their plausibility of being good object hypotheses, based on mid-level properties. Ranking also involves removing duplicates and diversifying the segments.

Figure 2.3: Figure taken from [APTB+14]. Top: original image, instance-level groundtruth and MCG multiscale hierarchical segmentation. Bottom: best MCG object candidates among 400.

the achievable quality as high as possible. To further reduce the number of candidates, they are ranked using a regressor from low-level features, such as the size and location, the shape and the contour strength. Finally, the candidates are also diversified based on Maximum Marginal Relevance measures. Figure 2.3 shows an example of object candidates resulting from applying MCG to an image.

2.3 Local features

A large variety of feature descriptors has been proposed for visual analysis, such as Gaussian derivatives, moment invariants, steerable filters, phase-based local features, and descriptors representing the distribution of smaller-scale features within the interest point neighbourhood. Based on the latter, one of the most known and used local features is the so-called Scale Invariant Feature Transform (SIFT). Other popular local features such as Local Binary Pattern (LBP) and Histogram of Oriented Gradients (HOG), which will be used in this dissertation, are also overviewed in this section. In Part I, these local features are computed over the different areas defined by the object candidates (object, contour and surround). This local features are aggregated (see aggregation techniques in Section 2.4) and used to build models for object recognition. Then, in Part II, these local features are used to compare contour elements from different images based on the texture over a patch around them. These similarities are injected into an optimization process to build a coherent multiview segmentation.

2.3.1 Scale Invariant Feature Transform (SIFT)

The Scale Invariant Feature Transform (SIFT) [Low04] proposes a representation from local image gradients in the regions around each interest point that is invariant to local shape distortion and change in illumination. First, the image gradient magnitudes and orientations are sampled around a location and weighted by a Gaussian function that gives less emphasis to gradients that are far from the center of the descriptor. Then, 8-bin orientation histograms are created over 4×4 sample regions (see Figure 2.4). Therefore, each location is represented by a 128 ($4 \times 4 \times 8$) element feature vector. The contribution of each gradient to its corresponding orientation bin depends on the gradient magnitude. SIFT descriptors are computed either in locations given by interest points detectors ([Low04] also proposes a detector based on a Difference of Gaussians) or in locations of a uniform grid over the image.

The work in [CCBS12] proposes a variation of the SIFT called masked SIFT (MSIFT), which is applied to CPMC object candidates [CLS12]. Before computing the descriptor, the background is set to zero intensity value and a color compression is also performed over the region such that the foreground colors belong to [50,255] intensity range. This way, the shape information is not lost independently of the color of the foreground. Furthermore, also in [CCBS12], the authors propose to enrich both SIFT and MSIFT with raw image information (RGB, HSV and LAB color values) and the relative location and the scale of the local features, which will be referred to as eSIFT and eMSIFT.

2.3.2 Local Binary Pattern (LBP)

In [OPM02], a texture operator that allows for detecting uniform local binary patterns at circular neighborhoods of any quantization of the angular space and at any spatial resolution is proposed. There is a parameter P that controls the number of pixels distributed uniformly in a circle of radius R, which is the parameter that determines the spatial resolution of the operator. Considering the sign of the differences in gray intensity between each pixel and the pixel at center results in a binary vector with P bits and 2^P possible values, which is invariant to illumination changes. In order to also achieve rotation invariance, the binary vector is shift-rotated so many times that a maximal number

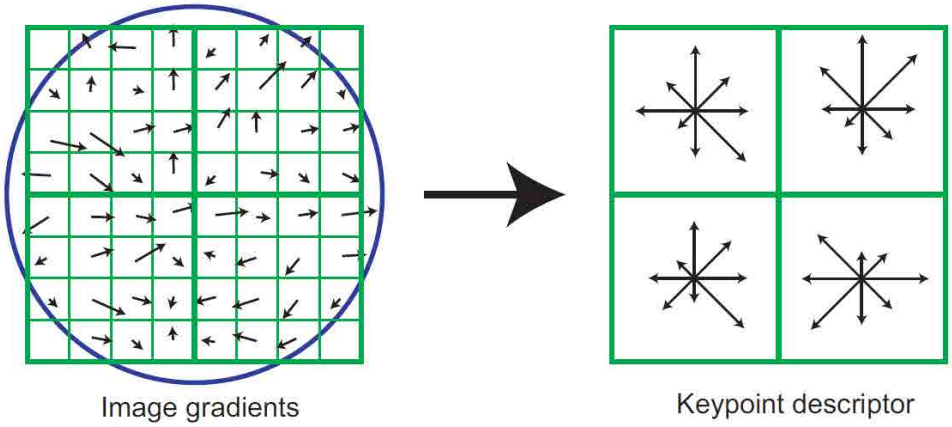


Figure 2.4: Figure taken from [Low04]. A SIFT descriptor is created by first computing the gradient magnitude and orientation at an image location in a region around it, as shown on the left. These are weighted by a Gaussian window, indicated by the overlaid circle. These samples are then accumulated into orientation histograms summarizing the contents over 4x4 subregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. This figure shows a 2×2 descriptor array computed from an 8×8 set of samples, whereas SIFT uses 4x4 descriptors computed from a 16x16 sample array.

of the most significant bits are 0. Figure 2.5 shows the 36 different binary patterns that can occur when a 45° angular space is considered, where white circles represent pixels with a gray intensity greater than the gray intensity of the pixel at center and black circles represent pixels with a gray intensity smaller than the gray intensity of the pixel at center.

As done with SIFT and MSIFT, in [CCBS12] it is also proposed to enrich LBP with raw image information (RGB, HSV and LAB color values) and the relative location and the scale of the local features, which will be referred to as eLBP.

2.3.3 Histogram of Oriented Gradients (HOG)

In [DT05], a texture descriptor known as Histogram of Oriented Gradients (HOG) is proposed. Similar to SIFT descriptor, HOG descriptor is also based on the accumulation of local intensity gradients in local orientation histograms over small spatial regions called cells. In object detection and retrieval applications, the classic exhaustive window sliding approach is usually combined with this representation. In practice, this is implemented by dividing the image window into cells and, for each cell, a local histogram of gradient directions or edge orientations is computed over the pixels of the cell. Each pixel calculates a weighted vote for an edge orientation histogram channel based on the orientation of the gradient element centered on it, and the votes are accumulated into orientation bins over the cells. This descriptor has been proved to be specially useful for human detection [DT05].

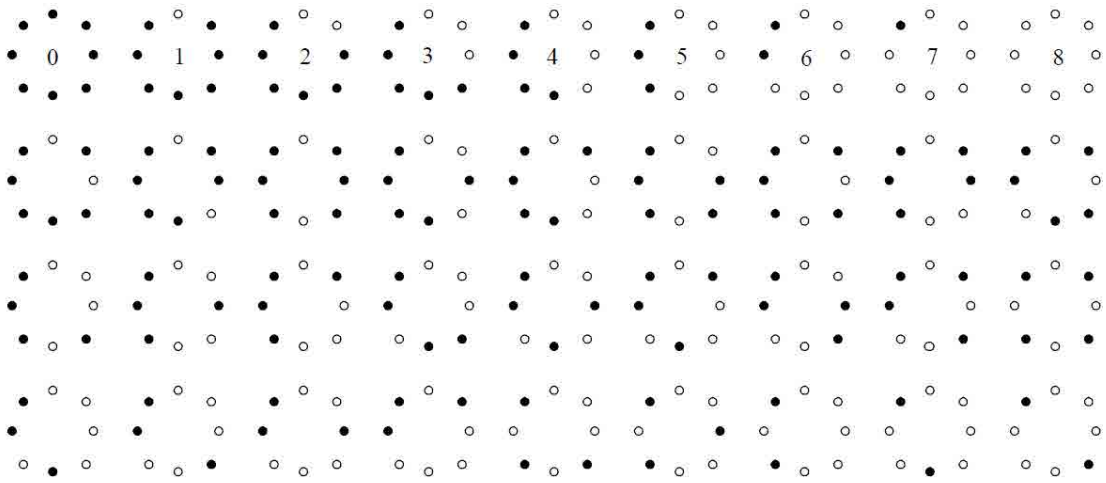


Figure 2.5: Figure taken from [OPM02]. These are the 36 unique rotation invariant binary patterns that can occur in the circularly symmetric neighbor set of $P = 8$. Black and white circles correspond to bit values of 0 and 1 in the 8-bit output of the operator.

2.4 Local features aggregation

Given an image region, which can be result of a segmentation algorithm or an object proposal technique, local features are usually aggregated in order to represent the entire region by a single descriptor instead of a set of them. The most popular way of aggregating the local features is the so-called Bag-of-Features (BoF) or Bag-of-Visual-Words (BoVW). However, other techniques more focused on second order moments such as Second Order Pooling (O2P) have also shown great performance in object recognition challenges. Next, we give an overview of both techniques.

2.4.1 Bag-of-Features (BoF)

Traditionally, local features such as SIFT are quantized by using a visual vocabulary or codebook [SZ03], allowing the use of analogous text retrieval techniques. In text, the basic units are the stems, which group different words that have the same root. For instance, the stem *run* can represent several words such as *run, runner, running, runnners*, etc. Analogously, in the visual field, visual words are equivalent to stems. Therefore, regions around locations that generate similar local features are assigned the same visual word. Figure 2.6 shows examples of regions whose local features are assigned the same visual word.

The visual vocabulary or codebook is generated applying a clustering algorithm, e.g. k-means, over the local features computed from a training set of images big enough to represent as accurately as possible the data to be analyzed. With the visual vocabulary, each image local feature is assigned its nearest visual word and, therefore, the image can be represented as a vector $\mathbf{v} = (v_1, ..., v_N)$ of visual word occurences, where N is the size of the codebook $C = \{w_1, ..., w_N\}$, and v_i is the number of times a local descriptor has been assigned the visual word w_i. This image representation is known as Bag-of-VisualWords

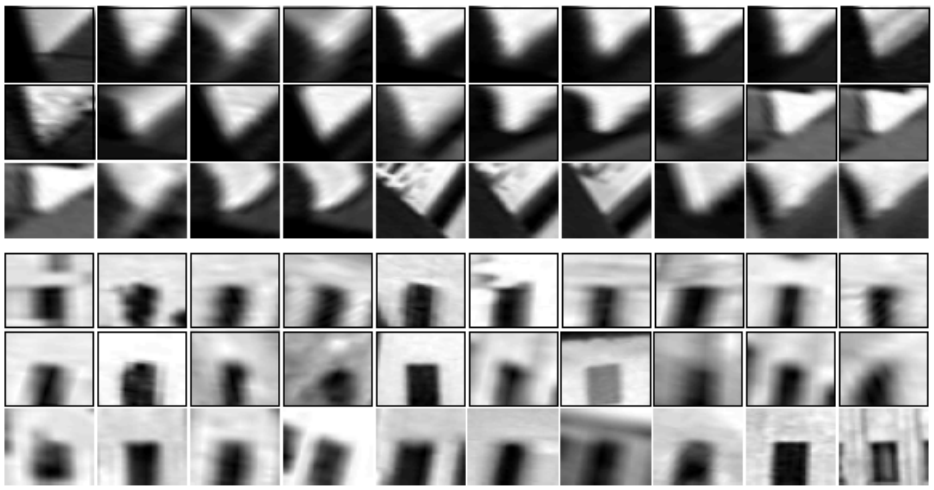


Figure 2.6: Figure taken from [SZ03]. It shows two sets of regions from two different clusters. Each region set is assigned the same visual word.

(BoVW) or Bag-of-Features (BoF), which is equivalent to the extended Bag-of-Words (BoW) used in text retrieval, where each document is represented as a vector of word frequencies.

Despite the lack of spatial information, the BoF image representation has been very popular in computer vision and visual content analysis in recent years. It has shown remarkable results for a wide variety of applications such as object recognition, image and video annotation or video event recognition. However, in order not to completely lose the spatial information, [LSP06] proposes to include a spatial codification called Spatial Pyramid, which is next described.

Spatial Pyramid

Bag-of-Features methods, which represent an image as an orderless collection of local features, disregard all information about the spatial layout of the features and are incapable of capturing shape or of segmenting an object from its background. The work in [LSP06] proposes to partition the image into increasingly fine sub-regions by repeatedly doubling the number of divisions in each axis direction and to compute histograms of local features found inside each sub-region. The final descriptor results from the concatenation of the histograms obtained at the different sub-regions, also including the histogram from the entire image. Figure 2.7 shows an example of a three-level Spatial Pyramid.

2.4.2 Second Order Pooling (O2P)

Another way of aggregating the local features is focusing in multiplicative second-order interaction (e.g. outer products) as done in [CCBS12]. In such work, the second-order average-pooling is defined as the average of the outer products for all local features in the region. Then, since the resulting matrices form a Riemannian manifold, it is possible to map them to an Euclidean tangent space by computing the matrix logarithm operation. Finally, power normalization is applied and the final global region descriptor vector is formed by concatenating the elements of the upper triangle of the resulting matrix.

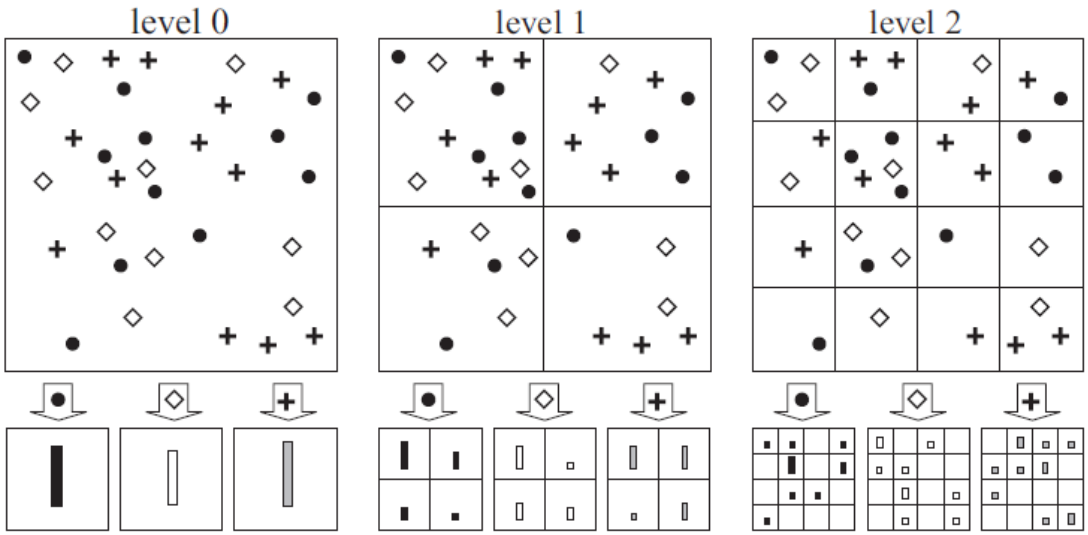


Figure 2.7: Figure taken from [LSP06]. Example of a three-level Spatial Pyramid. The image has local features that have been assigned to three different visual words, indicated by circles, diamonds and crosses. At the top, we subdivide the image at three different levels of resolution. Next, for each level of resolution and each visual word, we count the local features assigned to such visual word.

The main advantage of O2P with respect to BoF is that O2P does not require building any vocabulary. Therefore, it is a parameter-free aggregation technique because it does not depend either on the size of the codebook, i.e. the number of visual words, or the training set of images over which the local features have been computed to build such a codebook. Although state-of-the-art results have been obtained in [CCBS12], to our best knowledge, there are no works comparing both aggregation techniques. Due to its lack of parameterization, O2P will be used in Part I to aggregate the local features over the different regions considered there.

Part I

Context analysis in semantic segmentation

Introduction

3

The term "context" lacks a clear definition in computer vision and there exist different sources of context that have been discussed in the literature [DHH+09]. In this dissertation, we are going to focus on the *local pixel context*, which is defined as the image pixels/patches around the region of interest. More specifically, we are interested in performing an analysis on how the context influences on local feature descriptors such as SIFT descriptors, as well as on local feature aggregation techniques such as Bag of Features or Second Order Pooling.

This part analyzes the influence of context in both local features description and aggregation through the semantic segmentation problem, which is defined as the labeling of each pixel in an image to one of a set of category labels. The Pascal VOC Segmentation challenge [EVGW+10] provides a benchmark for semantic segmentation assessment. This problem can be solved by addressing two different challenges: (a) determining the precise regions that represent the objects, and (b) labeling each of these regions with the appropriate object class. Whereas the first challenge is left to state-of-the-art object proposal techniques, such as CPMC and MCG, this work addresses the second challenge. Furthermore, the division of the problem in these two challenges allows identifying the different instances in an image and, therefore, addressing the instance-aware semantic segmentation problem.

The classic approach to solve the second challenge, i.e. labeling the regions with the appropriate object class, has been based on SIFT-like and HOG-like features, which are commonly aggregated within each region using Bag-of-Features (BoF) [AHG+12, CLS12, RLYFF12] or, more recently, Second Order Pooling (O2P) techniques [CCBS12, YBS13]. In addition, approaches based on convolutional neural networks (CNN) have gained popularity among the scientific community thanks to the results achieved by works such as [GDDM14] or [HAGM14]. However, training a CNN end to end in a supervised manner requires a large amount of pixel-wise annotation. This has been done for a part of Microsoft Common Objects in Context (CoCo) [LMB+14] with crowdsourced annotators, but these classes do not represent the whole range of possible domains. Domains with costly and scarce pixel-wise annotation, e.g. medical domain, that require high expertise to be annotated can still benefit from off-the-shelf local features, whether hand-crafted (SIFT or HOG) or learned from other domains. These local features can be better exploited with an appropriate analysis of the spatial context, as explored in this part of the dissertation, extending our work presented in [VGiNV+15].

Specifically, we propose to improve the visual description by partitioning the image into three regions (Figure, Border and Ground) inspired by the work reported by Uijlings et al in [USS12]. Multiple authors have highlighted the importance of the spatial context around an object during its recognition [DT05, HJS09, FGMR10]. In our work,

we show the potential of the Figure-Border-Ground spatial feature aggregation by using object candidates instead of ground truth masks (as done in [USS12]). Our proposal is tested over two state-of-the-art object candidate algorithms: CPMC [CS12] and MCG [APTB+14]. Introducing the Border region pool for object candidates represents a novelty with respect to the previous works [CH07, LCS10, CCBS12, RLYFF12] which only considered Figure-Ground spatial feature aggregation for such a scenario. This intermediate area aims at minimizing the influence of the context in the object description and vice versa, as well as at capturing the rich contextual information located in the very neighbourhood of the object itself. Furthermore, our work explores a novel approach for enriching the visual description of the object. We propose to apply a contour-based Spatial Pyramid over the Figure region based on two different configurations: (*i*) a crown-based Spatial Pyramid, where the object is divided into different crowns for aggregation, and (*ii*) a cartesian-based Spatial Pyramid, where the object is divided into four geometric quadrants for aggregation. These approaches for a richer spatial codification are combined with the O2P pooling [CCBS12].

Our richer spatial codification proposals have improved the Figure-Ground spatial feature aggregation from [CCBS12], which have been assessed on the Pascal VOC Segmentation challenge under two different configurations.

In the first configuration, only Pascal VOC data is used for training (known as *comp5*). In this case, our approach improves the results from [CCBS12] with a gain of 12.9%. Our results for *comp5* are still far from the state-of-the-art results [LCLS13, XSF+12], with a drop of 10.2%, but the results from [LCLS13, XSF+12] have been obtained by using the bounding box annotations from the Pascal VOC detection challenge, which are not used in our experiments. Notice that these bounding box annotations represent an increase of about 440% images for training. The goal of this work is to show that a better spatial codification improves the system performance, no matter how much data is used for training the system.

In the second configuration, Pascal VOC training data is extended with a large additional dataset (known as *comp6*). In this case, the presented solution shows an increase of performance of 3.0% with respect to [CCBS12]. Including a richer spatial codification leads to the second best non-CNN-based technique in the VOC2012 Segmentation challenge *comp6*, presenting a drop of 3.8% with respect to the best non-CNN-based technique [DCYY14].

Figure 3.1 shows two examples where the proposed richer spatial feature aggregation based on a Figure-Border-Ground partition improves both the object segmentation and recognition with respect to a Figure-Ground spatial feature aggregation [CCBS12]. The first row shows an example where the achieved segmentation is more accurate, whereas the result in the second row corrects an erroneous object recognition made by the original O2P approach.

This part is structured as follows. Chapter 4 gives an overview of the related work about how the context is tackled in local feature description as well as in local feature aggregation to analyze if the context is beneficial or not for object recognition and semantic segmentation. In this chapter, we also present our contributions for this part. Then, Chapter 5 gives the experimental results in a semantic segmentation benchmark, where the proposed contributions are assessed in two different scenarios: an ideal scenario,

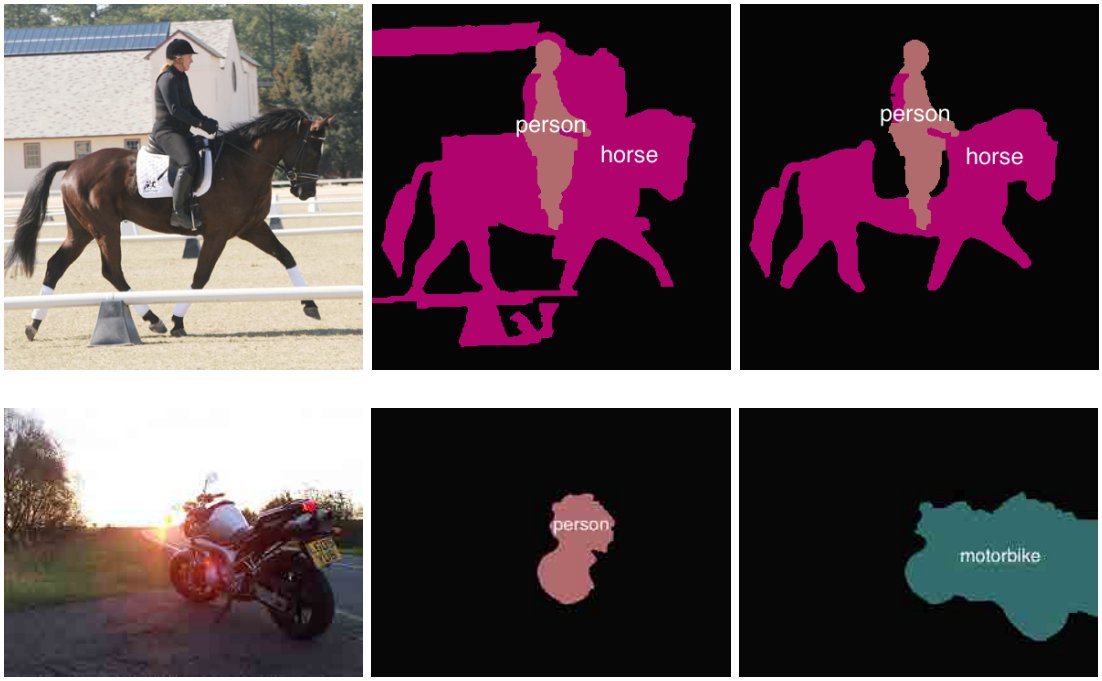


Figure 3.1: Examples where a richer spatial codification improves the object segmentation and recognition. Left: images to be semantic segmented. Middle: solution based on a Figure-Ground spatial feature aggregation [CCBS12]. Right: solution based on a Figure-Border-Ground spatial feature aggregation.

where the object locations are known, and a realistic scenario, where object proposal techniques are used for object location estimation. Finally, conclusions are drawn in Chapter 6.

Spatial context in semantic segmentation

4

In this chapter, Section 4.1 presents an overview of some studies performed about whether considering the context in the local features description is beneficial or not for object recognition tasks. In addition, we analyze as well the works that tackle how important is considering the accurate spatial support which carries shape information or whether the use of bounding boxes is even better. Then, Section 4.2 reviews some works that either exploit or avoid context when local features are aggregated to obtain a single feature for image or region representation using spatial aggregation techniques such as Bag of Features (BoF) or Second Order Pooling (O2P). Finally, Section 4.3 presents our contributions regarding spatial aggregation techniques.

4.1 Context-awareness in local features description

In object recognition, there have been suggestions that a bounding box, which provides some degree of context, may actually be beneficial. Thus, many state-of-the-art techniques [VJ01, DT05, HJS09, FGMR10, ZCYF10] are based on exhaustive search over the image using a sliding window at multiple scales with the goal that some of these windows fit with the bounding boxes of the objects in the image. Next, we give some details about these works that consider spatial context.

In 2001, Viola-Jones [VJ01] brought together new algorithms and insights to construct a framework for robust and extremely rapid object detection, achieving state-of-the-art results while being 15 times faster than previous approaches in frontal face detection. Hundreds of thousands of windows subimages are analyzed in each image through an exhaustive search popularly known as sliding window. Thus, an object detector is scanned across the image at multiple scales and locations. At each scale s, the window is shifted a number of pixels proportional to the scale s. This approach was feasible in an efficient way thanks to the use of a cascade of increasingly complex classifiers which allows a quick discarding of background regions at an early stage while spending more computation on promising object-like regions.

[DT05] proposes grids of histograms of oriented gradient (HOG) descriptors for human detection and uses a detection window which includes a margin around the person on all four sides to provide a significant amount of context that helps detection. Experimental results showed that decreasing the context decreases the performance. However, all these experiments have been performed assuming that the spatial support must be rectangular so that the exhaustive search is performed with a sliding window approach.

The approach of [FGMR10] builds on a framework that represents objects by a collection of parts arranged in a deformable configuration, which is referred to as deformable part

Figure 4.1: Figure taken from [Ram07]. Examples of false positives for a face detector (left), a pedestrian detector (middle), and a car detector (right) using scanning-window template classifiers.

models (DPM). These models are trained using a discriminative procedure that only requires bounding boxes for the objects. Similar to [FGMR10], [ZCYF10] presents a latent hierarchical learning method for object detection where an object is represented by a 3-layer tree structure model. The first layer has one root node that represents the entire object. The root node has nine child nodes at the second layer in a 3×3 grid layout, and each of them has four child nodes at the third layer. All tree nodes are rectangular windows over which HOG descriptors are computed. Also based on an exhaustive search, [HJS09] proposes a two stage sliding window object localization method that combines the efficiency of a linear classifier for pre-selection with the robustness of a sophisticated non-linear one for scoring.

All previous approaches consider spatial context due to a matter of efficiency and avoid exploiting region-based representations. However, [TTK+14] extends the work of [FGMR10] with a segmentation-based approach and without putting efficiency aside. They propose to split the low-level features into object-specific and background features according to a segmentation mask that can be computed fast enough to repeat this process over every candidate window. This approach outperforms the standard DPM in 17 out of 20 categories in PASCAL VOC 2007 (an object recognition benchmark), yielding an average increase of 1.7% in average precision.

In the same direction, according to [Ram07], approaches based on scanning-window template classifiers can be hindered by their lack of explicit encoding of object shape, resulting in high false-positives. Figure 4.1 shows examples where a face detector becomes confused by edges in foliage, a pedestrian detector mistakens strong vertical edges for a person, whereas a car detector mistakes strong horizontal edges. The authors propose to use the scanning-window template classifiers to generate possible object locations and compute a local figure-ground segmentation at each hypothesized detection to prune away those hypotheses with bad segmentations (see Figure 4.2). This strategy leads to significant improvements (10-20%) over established approaches such as Viola-Jones [VJ01] for finding faces and Dalal-Triggs [DT05] for finding pedestrians and cars on a variety of benchmark datasets including the PASCAL challenge [EZWVG06], LabelMe [RTMF08], and the INRIAPerson dataset [DT05]. This approach is similar to the one recently proposed in [TTK+14]. The main difference is that [TTK+14] uses a soft segmentation mask, whereas [Ram07] computes a binary figure-ground segmentation using graphcut [BVZ01].

Instead of an exhaustive search as previous approaches, which needs constraining the computation per location, [vdSUGS11] proposes adopting segmentation as a selective

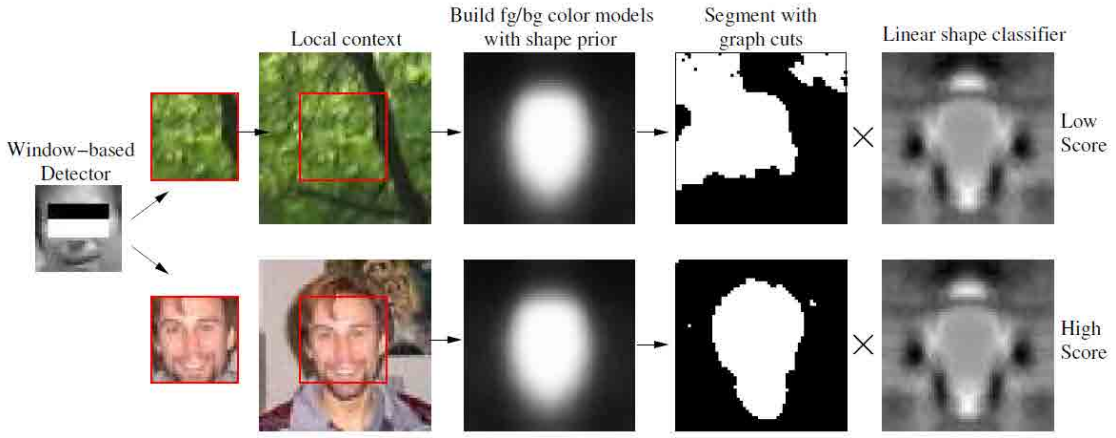


Figure 4.2: Figure taken from [Ram07]. Local figure-ground segmentation for the foliage is detected as a false positive when it is fed into a linear shape classifier. On the other hand, face is correctly detected as a true positive.

search strategy for object recognition. The use of segmentation to generate a limited set of location allows to compute the more powerful yet expensive Bag-of-Features. The class-independent method that [vdSUGS11] proposes is shown to cover 96.7% of all objects in the PASCAL VOC 2007 test set [EVGW+10] using only 1,536 locations per image, instead of the over 100,000 locations visited by sliding window techniques. They propose to generate locations at multiple scales using all locations from a hierarchical segmentation algorithm. However, since local context is thought to be beneficial for object classification [ZMLS07], they prefer object approximations over exact object boundaries and consider the tight bounding boxes around all segments throughout the hierarchy.

Forgetting about efficiency, in [ME07], it is claimed that an accurate spatial support is important for object recognition. Knowing a pixel-wise spatial support leads to substantially better recognition performance for a large number of object categories, especially those that are not well approximated by a rectangle, such as sheep, bike and airplane object categories. Although classic rectangular sliding window approaches are known for outstanding results on faces, pedestrians, and front/side views of cars (all rectangular-shaped objects), they have trouble distinguishing foreground from background when the bounding box does not correctly cover an object (see Figure 4.3). They have also demonstrated remarkable performance recognizing more complicated categories, but in datasets such as Caltech-101 where there is a single object per image and with relatively correlated backgrounds. In [ME07], for each object in an annotated dataset (Microsoft Research Cambridge dataset), they estimate its class label in two scenarios: (*i*) using only the pixels inside the object's ground-truth support region, and (*ii*) using all pixels in the object's tight bounding box. Experiments show that objects that are poorly approximated by rectangles see the largest improvement (over 50%) when object's ground-truth support regions are used and that categories that do not show improvement with better spatial support already have remarkable performance. Overall, the recognition performance using ground-truth segments is 15% better than using the bounding boxes.

For those approaches avoiding spatial context and using texture features such as SIFT, it is also important how context is tackled when the region of support over which the texture

Figure 4.3: Figure taken from [ME07]. Examples from Pascal dataset where up to half of the bounding box pixels do not belong to the object of interest.

descriptor is computed covers part of the context. We focus on texture descriptors since color, shape and location features do not show this problem. The most straight-forward way of considering local features free of context in their description is taking into account only features with a region of support that overlays entirely within the image region of consideration as performed in [USS12].

Apart from this naive approach, other works can be classified into two main groups: (*i*) those that perform a pre-processing of the image before computing the local feature descriptor [RVG+07, CCBS12], and (*ii*) those that perform a post-processing of the local feature descriptor [TLF10, TKSMN13].

On the one hand, [RVG+07] and [CCBS12] are examples of dealing with the context by pre-processing the image before the descriptors extraction. In [RVG+07], each region is considered as a stand-alone unit by masking and zero padding the original image. Then, local features are computed as usual, but discarding any feature that falls entirely outside its boundary. As a consequence, masking greatly enhances the contrast of the region boundaries making features along the boundaries more shape-informative. The work in [CCBS12] extends the original idea from [RVG+07] applied to object candidates [CLS12] where in addition to setting the background to zero intensity value, a color compression is also performed over the foreground such that the foreground colors belong to [50,255] intensity range. This way, the shape information is not lost independently of the color of the foreground.

On the other hand, some research lines such as [TLF10] and [TKSMN13] apply a post-processing to the descriptor once extracted instead of a pre-processing of the image. In [TLF10], a descriptor called Daisy is proposed, which is similar to SIFT but more efficient to be computed. As SIFT, this descriptor consists of several histograms computed over different spatial locations. However, in multiview scenarios, pixels that are close to an occluding boundary will be different when captured from different viewpoints. To handle this, [TLF10] estimates an occlusion map and exploit it to define binary masks

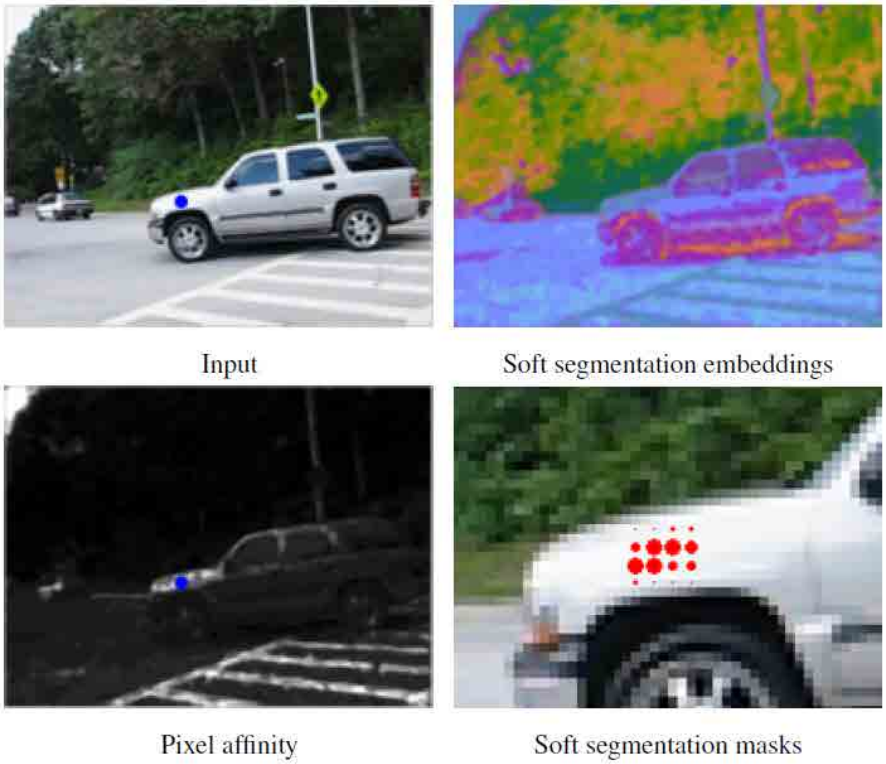


Figure 4.4: Figure taken from [TKSMN13]. Top-left image: source image and a feature point x. Top-right image: RGB encoding of soft segmentation masks. Bottom-left: segmentation-based affinity between x and the whole image. Bottom-right: affinity values at the cells of a SIFT descriptor.

over the Daisy descriptors. The goal of these binary masks is, given a spatial location, deactivating the histograms from locations that are likely to be occluded in the other viewpoints. This way, spatial coherence of the occlusion map is enforced, allowing for proper handling of occlusions. Although this approach is thought for multiview datasets where occlusion map can be estimated, it could be extended to single images considering regions from a segmentation as binary masks analogous to occlusion maps, where strong contour detections would be treated as occlusion boundaries. Closer to our approach, [TKSMN13] proposes to downplay measurements coming from areas that are unlikely to belong to the same region as the descriptor's center, as suggested by soft segmentation masks. The main difference with respect to the work from [TLF10] is that the authors in [TKSMN13] do not make binary decisions, but each histogram that forms the descriptor is given a weight depending on the likelihood that the location of the histogram and the descriptor's center belong to the same region. This is possible thanks to the computation of soft segmentations, which determine the affinity of a pixel to its neighbors in a soft manner. Figure 4.4 shows an example of segmentation-aware descriptors.

4.2 Context-awareness in local features aggregation

Whereas Section 4.1 addresses how techniques isolate the object from the context in local features description using pre-processing or post-processing methods, this section is focused on how context is tackled in local features aggregation. Thus, some works

study whether it is better to jointly pool all features over the image or independently pool those over the foreground from those over the background.

[ZMLS07] addresses the evaluation of background features and shows the pitfalls of training on datasets with uncluttered or highly correlated backgrounds. Thus, this causes overfitting and yields disappointing results on more complex test sets without correlated backgrounds. Their experiments reveal that the features on the objects themselves play the key role for recognition. According to [ZMLS07], using foreground and background features together does not improve the performance despite the discriminative information contained by backgrounds. To determine whether background features provides additional cues for classification, they examine the change in performance when the original background features from an image are replaced by two specially constructed alternative sets: random and constant natural scene backgrounds. The results show that foreground features always give the highest accuracy, indicating that object features play the key role for recognition, and recognition with segmented images achieves better performance than without segmentation. Thus, mixing background and foreground features does not give higher recognition rates than foreground features alone. The experiments also show that when different types of background are used in training but original backgrounds in testing, SVM can find decision boundaries that generalise well to the original training set. Thus, one of the conclusions of [ZMLS07] is that the presence of varied backgrounds during training helps to improve the generalisation ability of the classifier. Therefore, when background from test images may not show a high correlation with background from training images, foreground features give the best performance.

Note that in [ZMLS07] when both background and foreground features are considered, they are jointly pooled and, therefore, mixed. However, other works such as [USS12] and [CCBS12] consider background and foreground features independently pooled, i.e. local features aggregation techniques such as Bag-of-Features (BoF) or Second Order Pooling (O2P) are applied over those regions resulting in two aggregated features, which are later concatenated forming a single aggregated feature. Thus, although both background and foreground features are taken into account, they are not mixed.

In [USS12], an analysis of the visual extent of an object is performed using ground truth annotations on the Pascal VOC dataset with SIFT-based BoF. Whereas in a normal situation, where the object location is unknown and all features are jointly aggregated, the classification accuracy is 0.44 MAP, the accuracy classification rises up to 0.62 MAP when object location is used to separate foreground and background features. The huge difference between the accuracy with and without knowing the object location shows that the classifier cannot distinguish if visual words belong to the object or surround. Furthermore, ground truth object locations are also used to create a separate representation with 3 types of regions: the object's surrounding (Ground), near the object's contour (Border) and the object's interior (Figure). The local features are aggregated within each of these 3 regions and the resulting BoF histograms are concatenated to describe the whole image. As a result of introducing the Border region to the Figure and Ground ones, a gain of 11.3% in accuracy is achieved, leading to an accuracy classification of 0.69 MAP. Therefore, this points out that having separated visual words for describing interior and contour is also useful as previously showed that a classifier takes advantage of having separated descriptions for object and surround.

In [CCBS12], it is assumed that background features carry useful information for object

recognition when they are considered aside but there are no experiments that prove it. As in [USS12], ground truth object locations are used to aggregate the local features over the foreground (Figure) and background (Ground) independently, but using O2P instead of BoF. In contrast with [USS12], they considered object proposals techniques (CPMC [CS12]) to perform figure-ground segmentations instead of only considering an ideal scenario where ground truth object locations are provided.

The spatial coding of pooled features has not only been addressed from the perspective of taking as a reference automatically generated regions as the figure-ground segmentations given by object proposals techniques, but also through an arbitrary partition of the image. This is the case of the popular Spatial Pyramid (SP) [LSP06], which consists in dividing the whole image into a grid and pooling the descriptors over each cell of the grid using a BoF framework. The resulting BoF histograms for each cell as well as the BoF histogram for the whole image are concatenated and used as a richer description of the image. In [AHG+12] and [GAL+12], the concept of Spatial Pyramid is extended to object bounding boxes. The goal of SP is having a more accurate description of different region areas of the image or bounding box that otherwise is lost due to the BoF's own nature.

4.3 Contributions

In this section, we present the contributions for this part of the dissertation. First, Section 4.3.1 extends the Figure-Border-Ground spatial pooling of [USS12] in a ideal scenario to object location hypotheses given by object proposal techniques as CPCM [CS12] or MCG [APTB+14]. Then, in Section 4.3.2, a contour-based Spatial Pyramid is proposed, which extends the SP presented in [LSP06] for images to regions with two spatial configurations. Both contributions will be contrasted in a semantic segmentation benchmark (PASCAL VOC 2011 and 2012 [EVGW+]), being the results presented in Chapter 5.

4.3.1 Figure-Border-Ground with object candidates

In this dissertation, we analyze the assumption from [CCBS12] that considers that there is an increase in performance when background and foreground features are independently aggregated in two scenarios: (*i*) ideal, i.e. when object locations are given, and (*ii*) realistic, i.e. when object proposal methods are used to estimate their locations.

Furthermore, we extend the spatial pooling based on a Figure-Border-Ground image partition from [USS12] by exploring its impact when applied in the realistic case of automatically extracted object candidates instead of ground truth masks for the semantic segmentation challenge. In contrast with [USS12], we define a region pool as the spatial layout where the local features can be centered independently of the extension of the spatial support over which the local descriptors are computed. As a consequence, the local descriptors extracted from a region which are near the region contour can partially describe the neighbour region except for those ones where pre-processing or post-processing techniques presented in Chapter 4.1 are applied (e.g. masked SIFT). In this way, we allow the use of the common 4×4 SIFT descriptors as well as a multiscale dense feature detector instead of the 2×2 SIFT descriptors extracted at one single scale from [USS12]. These small 2×2 SIFT local descriptors are used in [USS12] to ensure that Figure and

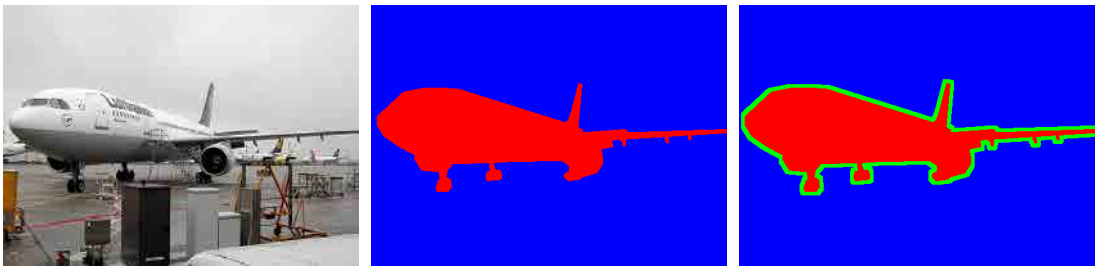


Figure 4.5: Example of a Figure-Ground partition [CCBS12] (in the middle) and a Figure-Border-Ground partition (on the right) of the original image (on the left).

Ground descriptors are completely isolated one from the other. Figure 4.5 shows an example of a Figure-Ground and a Figure-Border-Ground image partitions.

This lack of absolute isolation of the description of each region pool can be justified in two ways. First, multiple authors have highlighted the importance of the spatial context around an object during its recognition [DT05, HJS09, FGMR10]. Second, the fact that in our experiments, in contrast with [USS12], we also use a masked SIFT (MSIFT), which excludes any visual information coming from the neighbour region because it is set to zero before being computed. Therefore, the learning process can differentiate the classes that can take advantage of the context (giving more importance to non-masked descriptors) from the ones where context can lead to confusion (giving more importance to masked descriptors).

The system proposed and released in [CCBS12], which is based on Second Order Pooling (O2P), has been adopted as a baseline to assess the proposal of extending the Figure-Border-Pooling with object candidates from the Pascal VOC Segmentation challenge. Thus, we have used the same object candidates trained for such a solution: CPMCs [CS12]. However, we have also checked the Figure-Border-Ground spatial pooling with another state-of-the-art object candidate technique (MCGs [APTB+14]) to further analyze the robustness of our proposal. In addition, this system allows us to check if the conclusions drawn in [USS12] are also valid when SIFT-based BoF are replaced by O2P features. More details about the system released in [CCBS12] are given in Section 5.2.

4.3.2 Contour-based Spatial Pyramid

Spatial Pyramid (SP) has proven to be successful for enriching the BoF framework in many object recognition techniques but has been broadly applied to obtain a description of the whole image. Few works, such as [AHG+12] and [GAL+12], have applied SP to bounding boxes instead of at the image level. However, applying SP to bounding boxes does not differentiate the object from its context and, therefore, does not take advantage of the accurate spatial support intrinsically given by the region. To our best knowledge, there are no works that extend the SP to a region-based approach where local features are only aggregated within the region, discarding any feature placed in the object's surround.

Therefore, in this dissertation, we propose a contour-based Spatial Pyramid that extends the spatial codification already applied to images and bounding boxes but preserving shape information and the separation of foreground and background features when applied to regions. Two different spatial configurations are proposed: (*i*) a 4-layer crown-

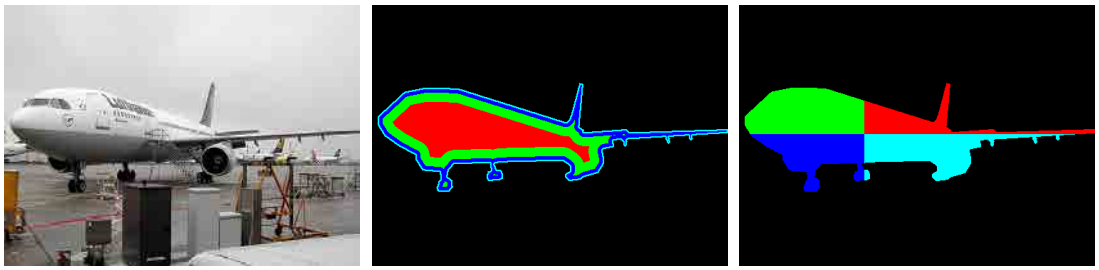


Figure 4.6: Example of a 4-layer crown-based (in the middle) and a cartesian-based (on the right) Spatial Pyramid from an object mask of the original image (on the left).

based SP, and (*ii*) a cartesian-based SP. Both configurations divide the region into 4 non-overlapping regions. The regions for the 4-layer crown-based SP are obtained by applying a distance transform to the Figure mask, which measures the distance from each pixel to the nearest pixel belonging to the background. Then, the maximum value is used to define the different layers on a logarithmic base. On the other hand, the cartesian-based SP divides the Figure region into 4 geometric quadrants which have the center of mass of the region as origin. Figure 4.6 shows an example of a 4-layer crown-based SP and a cartesian-based SP. Whereas the crown-based SP is invariant to changes in rotation and scale, the cartesian-based SP is only scale invariant. Both contour-based SP configurations have been verified in the ideal scenario, where it is applied to the object ground truth, as well as in the real scenario, where it has been applied to the Figure region in conjunction with the previously proposed Figure-Border-Ground spatial pooling (Section 4.3.1) with both CPMC and MCG object candidates over the architecture released in [CCBS12].

Experimental results

5

In this chapter, we present the results achieved in the experiments performed in a benchmark for semantic segmentation assessment, which is briefly introduced in Section 5.1. For such a benchmark, we have adopted as baseline a solution based on the architecture proposed and released in [CCBS12], which is overviewed in Section 5.2. It is over this solution that the proposals of partitioning the image into three different regions (Figure, Border and Ground) and the extension of the Spatial Pyramid to objects are built. The experiments have been performed according to the following scheme:

- *Ideal Object Candidates.* The semantic segmentation challenge is assessed in an ideal situation where object locations are known and they must be assigned to their corresponding category. These experiments are useful to evaluate if the proposed richer description improves the average accuracy under the assumption of perfect object candidates. The results of these experiments are presented in Section 5.3.
- *Realistic Object Candidates.* This configuration addresses the realistic scenario where a ranked list of pixel-wise object candidates are automatically generated. In this work, we have considered the regions proposed by the Constrained Parametric Min-Cuts (CPMC) [CS12], the same technique adopted in [CCBS12], since they allow a fair comparison of results. However, we have also considered the Multiscale Combinatorial Grouping (MCG) [APTB+14], another state-of-the-art technique for object candidate generation, to check the consistency of our two contributions for improving the spatial pooling. The results for CPMC and MCG are presented in Sections 5.4 and 5.5 respectively.

5.1 Pascal VOC semantic segmentation benchmark

The Pascal VOC Segmentation challenge [EVGW+10] provides a benchmark for semantic segmentation assessment where objects are classified into 20 categories: aeroplane, bicycle, bird, boat, bottle, bus, car, cat, chair, cow, diningtable, dog, horse, motorbike, person, pottedplant, sheep, sofa, train and tvmonitor. The dataset is divided into three subsets: train, validation and test. Whereas ground truth annotations are available for both train and validation subsets, annotations for test subset are not provided. Therefore, following the guidelines proposed by the challenge, preliminary experiments have been performed using the train subset for training and the validation subset for test. Once the best configuration has been found based on this methodology, the experiments have been performed again but using both train and validation subsets for training and test subset for testing. Results for this configuration are submitted to Pascal VOC server for their external evaluation.

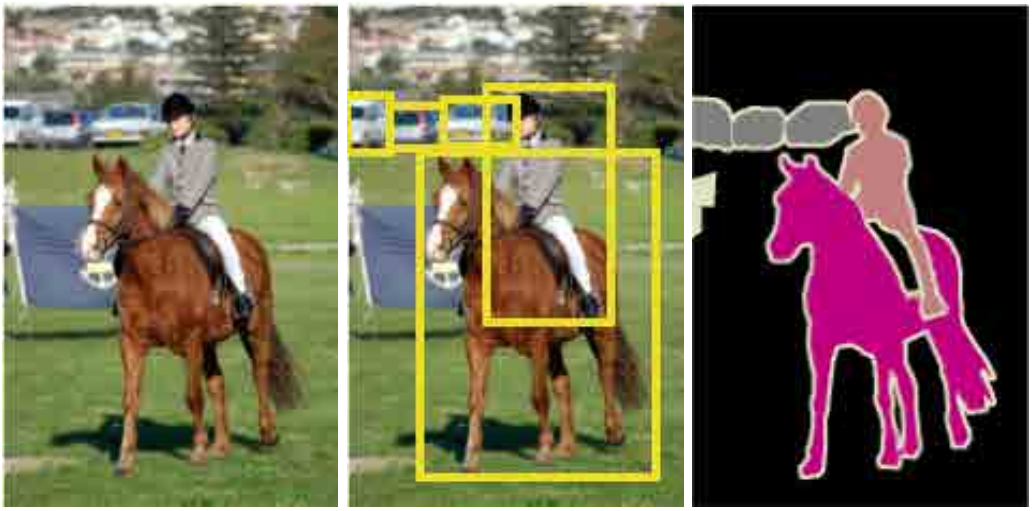


Figure 5.1: Detection and Segmentation challenges. Given an image (on the left), the detection challenge expects categorized bounding box detections as output (in the middle), whereas the segmentation challenge expects categorized pixels as output (on the right).

The Pascal VOC Segmentation challenge is divided in two different challenges or modalities: competition 5 (*comp5*) and competition 6 (*comp6*). The Pascal VOC Segmentation dataset is a subset of the Pascal VOC Detection dataset, which is used for another challenge where the results are given as bounding boxes instead of as accurate segmented regions. Figure 5.1 shows the difference between both challenges. According to [EVGW+10], for *comp5*, only annotations provided in the VOC train and val subsets (from both segmentation and detection datasets) may be used for training. Examples of such types of annotations are segmentation masks, bounding boxes or particular views (e.g. frontal or left). Participants are not permitted to perform additional manual annotation of either training or test data. Alternatively, for *comp6*, any source of training data may be used except the provided test images.

The experiments have been performed on the datasets for the Pascal VOC 2011 and 2012 segmentation challenges. Train and validation subsets for VOC 2011 segmentation consist of 1,112 and 1,111 images, and 2,501 and 2,533 objects, respectively. Train and validation subsets for VOC 2012 segmentation consist of 1,464 and 1,449 images, and 3,507 and 3,422 objects, respectively. For *comp6*, the extended dataset [HAB+11] consists of 12,031 segmented annotations.

The evaluation is performed by means of the Average of the Accuracy per Category (AAC). The Accuracy per Category (AC) is defined as the ratio between the intersection and the union of the pixels classified as category c_k and the pixels annotated in the ground truth as c_k. Once all Accuracy per Category values have been computed, they are averaged to obtain the AAC.

5.2 Baseline framework for semantic segmentation

In this work, we have adopted as baseline the solution proposed and released in [CCBS12] for semantic segmentation. At the time of developing this part of the dissertation, this

was the best technique for PASCAL VOC segmentation challenge *comp*5 and CNN-based solutions for semantic segmentation had not been proposed yet. However, CNN-based solutions are only accepted for *comp6* since they have been already trained with external data such as ImageNet [DDS+09] or Microsoft CoCo [LMB+14]. Thus, non CNN-based solutions are useful for light scenarios where features are manually designed instead of learned, reducing the need for large data collections and costly processing effort.

The adopted solution [CCBS12] is based on using Second Order Pooling (O2P) to aggregate the local features (see Section 2.4.2 for more detail about O2P). First, 150 CPMC object candidates [CS12] are extracted per image. Each object candidate is considered as a Figure-Ground segmentation of the image. Over the Figure region of each object candidate, three types of local features (eSIFT, eMSIFT and eLBP) are densely extracted and pooled independently using O2P. On the other hand, over the Ground region, only eSIFT descriptors are extracted and pooled. For each object candidate, Figure and Ground descriptors are combined by concatenation.

Then, a scoring function f_k is learned for each category c_k using linear regression based on the descriptors computed from the object candidates. Given an object candidate from a training image, f_k measures the overlap between the object candidate and the ground truth object belonging to c_k with highest overlap. Once f_k has been learned, this scoring function is used to predict the overlap that a candidate object from a test image would have with an object belonging to c_k. Then, the category with the highest predicted overlap is assigned to that object candidate.

Finally, a simple inference procedure is applied to generate the final image semantic segmentation. The segments with highest score above a background threshold are pasted onto the image in the increasing order of their scores.

5.3 Results with ideal object candidates

Experiments have been first performed using the ground truth object masks. The use of these masks allows us to isolate pure recognition effects from segment selection and inference problems. This way it is possible to assess the improvements provided by the various spatial codifications in an ideal scenario. These masks are only available for train and validation subsets. Therefore, the results with ideal object candidates use the train subset for training, and the validation subset for testing.

5.3.1 Figure-Border-Ground spatial pooling

In this section, we aim at assessing the addition of a region around the object contour as well as the importance of such region with respect to the rest of the background when ground truth object locations are provided. Table 5.1 shows the results for different image spatial representations. The first and third columns are from [CCBS12] , where the first column corresponds when Figure(F) is considered stand alone, whereas the third column corresponds to the classical Figure(F)-Ground(G) segmentation. In the Figure-Ground configuration, Border region is included in the Ground description. We propose two additional configurations: (*i*) Figure(F)-Border(B), and (*ii*) Figure(F)-Border(B)-Ground(G).

On the one hand, the Figure-Border configuration tries to answer the following question:

	F [CCBS12]	F-B	F-G [CCBS12]	F-B-G
eSIFT	63.85	66.24	66.43	**68.57**
eMSIFT	64.81	68.93	67.59	**70.84**

Table 5.1: Gain of introducing the Border for pooling. Results using GT masks. Training over train11 and evaluation over val11. F refers to Figure, B refers to Border and G refers to Ground.

how important is the whole background in comparison with the bordering region? When eSIFT descriptors are pooled, using only the Figure and Border regions and discarding the Ground region is almost as good as using the classical Figure-Ground partition of the whole image (66.24 and 66.43 respectively). If eMSIFT descriptors are pooled instead, the average accuracy achieved by pooling them over Figure-Border is even better than over Figure-Ground (68.93 and 67.59 respectively). This indicates that the richest contextual information for object recognition is located in the very near neighbourhood of the object itself. The increase of the performance when eMSIFT is used could reflect the intuition that this descriptor is more powerful for regions of arbitrary shape than for fixed-form regions. In this sense, Border region is much more shape informative than Ground region, which has a rectangular shape due to the image frame.

On the other hand, the Figure-Border-Ground configuration aims at showing the benefits of also including the rest of the background (what we call Ground in the Figure-Border-Ground spatial pooling) as a region pool. As shown in Table 5.1, pooling the local descriptors over Figure-Border-Ground image partition gives the best average accuracy. Although pooling over Border can give better results than pooling over Ground as seen before, Ground description still carries useful information for object recognition.

Furthermore, the results have also been analyzed by categories (see Figure 5.2 and Table 5.2). Pooling the eMSIFT descriptors independently over Figure and Border improves the accuracy in 15 out of the 20 categories. A similar conclusion can be drawn when Ground is also considered and the results from partitioning the image into 2 or 3 regions are compared. The use of Border as an independent region from Figure and Ground improves the accuracy in 18 out of the 20 categories.

An additional experiment has been performed to analyze if the increase in the performance when the Border region is introduced could also be achieved by including it into the Figure region, i.e. extending the limits of the Figure region so it includes the Border region. The results show that there is a decrease in the performance when both regions are jointly pooled and eSIFT is used (60.29), whereas the performance is not affected with respect to the use of the not extended Figure region with eMSIFT (65.08). This behaviour could be explained by the masking: when eMSIFT is pooled over Figure and Border jointly, the extension over which local descriptors are computed has increased only 5 pixels (all the background is set to 0) and the pooling still gives a good description of just the object. On the other hand, when eSIFT is used, placing local features in the Border region allows the description to reach further locations in the background since it is not masked. In any case, for both eSIFT and eMSIFT, the results show that pooling independently Figure and Border outperforms the average accuracy achieved by the jointly pooling of these regions.

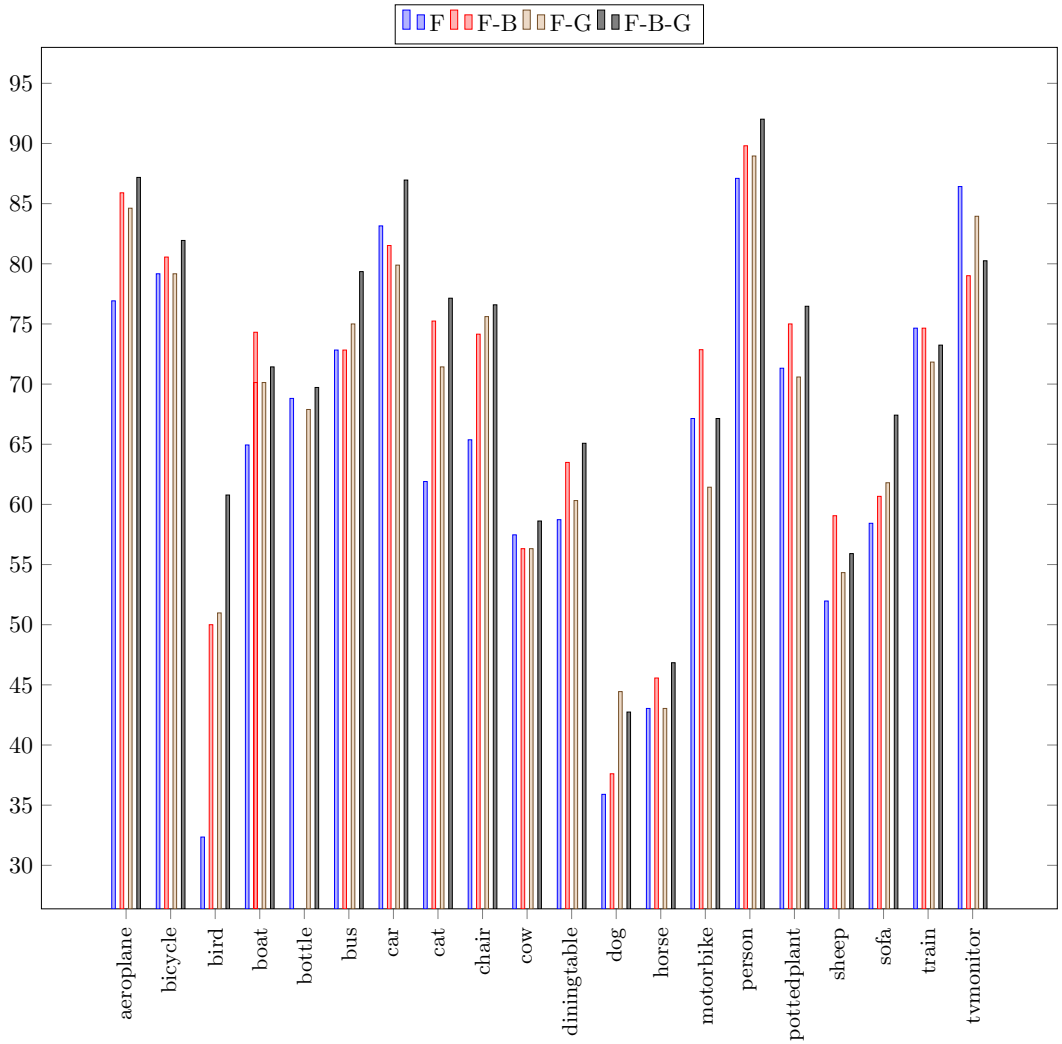


Figure 5.2: Accuracy Classification by Categories using ground truth masks and eMSIFT descriptor. Training over train11 and evaluation over val11.

While Table 5.1 analyzes the eSIFT and eMSIFT independently, Table 5.3 explores the joint combination of different descriptors by concatenation. In this table, each column refers to a region pool and each row refers to the local descriptors that have been pooled for each region. This study is performed to assess the impact of our proposal on the configuration with the best results obtained in [CCBS12]: with eSIFT-F, eSIFT-G, eMSIFT-F and eLBP-F (72.98), where F and G refer to the Figure and Ground regions over which the local features are pooled. Analogously, as shown in Table 5.3, using only eSIFT and eMSIFT descriptors and the proposal of partitioning the image into three regions (Figure-Border-Ground) improves the average accuracy up to 73.84 with respect to the 72.48 obtained in [CCBS12] (eSIFT and eMSIFT over Figure-Ground).

	F [CCBS12]	F-B	F-G [CCBS12]	F-B-G
aeroplane	76.9	85.9	84.6	**87.2**
bicycle	79.2	80.6	79.2	**81.9**
bird	32.4	50.0	51.0	**60.8**
boat	64.9	70.1	70.1	**71.4**
bottle	68.8	**74.3**	67.9	69.7
bus	72.8	72.8	75.0	**79.4**
car	83.2	81.5	79.9	**87.0**
cat	61.9	75.2	71.4	**77.1**
chair	65.4	74.2	75.6	**76.6**
cow	57.5	56.3	56.3	**58.6**
diningtable	58.7	63.5	60.3	**65.1**
dog	35.9	37.6	**44.4**	42.7
horse	43.0	45.6	43.0	**46.8**
motorbike	67.1	**72.9**	61.4	67.1
person	87.1	89.8	89.0	**92.0**
pottedplant	71.3	75.0	70.6	**76.5**
sheep	52.0	**59.1**	54.3	55.9
sofa	58.4	60.7	61.8	**67.4**
train	**74.7**	**74.7**	71.8	73.2
tvmonitor	**86.4**	79.0	84.0	80.3
Average	64.81	68.93	67.59	**70.84**

Table 5.2: Accuracy Classification by Categories using ground truth masks. Results by categories for different image spatial representations using eMSIFT descriptor. Training over train11 and evaluation over val11.

Figure	Border	Ground	AAC
eSIFT+eMSIFT+eLBP		eSIFT	72.98 [CCBS12]
eSIFT+eMSIFT	eSIFT+eMSIFT	eSIFT+eMSIFT	**73.84**

Table 5.3: Gain of introducing Border for pooling and combining eSIFT and eMSIFT. Results using GT masks. Training over train11 and evaluation over val11.

	F	F-B	F-B-G
non SP	64.81 [CCBS12]	68.93	70.84
crown-based SP	**68.67**	71.05	71.69
cartesian-based SP	67.66	**71.64**	**72.68**

Table 5.4: Comparison between the non use of Spatial Pyramid for the Figure region and the crown-based and cartesian-based Spatial Pyramid approaches. Results using GT masks. Training over train11 and evaluation over val11.

5.3.2 Contour-based Spatial Pyramid

In this section, we explore the proposal of improving the visual description by using the contour-based Spatial Pyramid presented in Section 4.3.2. Table 5.4 shows the results of applying the two Spatial Pyramids configurations (crown-based and cartesian-based) over the Figure region for the eMSIFT descriptors. In addition, eMSIFT descriptors are also pooled over the whole Figure region as a global region descriptor.

The results show that both types of Spatial Pyramids give a significative improvement of the average accuracy classification, specially when only the Figure region is considered. Although the crown-based SP is better than the cartesian-based SP for the Figure region, the cartesian-based SP gives the best performance when the Border and Ground regions are also considered. We believe that this behavior is caused by the fact that the description of the Border region is more diverse with respect to the geometric quadrants than the outermost layer of the crown-based SP.

Applying SP over the Border region was discarded due to its thinness (only 5 pixels width from the object contour). Since the average accuracy achieved by using only the Ground region is really low (26.47), we tried to enrich the background description by applying the SP over it. However, the 4-layer crown-based SP only increases the accuracy in 0.34 points whereas the cartesian-based SP results in a drop of 0.02 points with respect to the Figure-Border-Ground pooling without SP. As a result, we also discarded applying SP over Ground since we consider that this small improvement is not worth in comparison with the increment in the dimensionality of the features.

The performance achieved by using only the eMSIFT descriptor (72.68) is almost as good as the accuracy achieved in [CCBS12] by combining eMSIFT, eSIFT and eLBP (72.98). Table 5.5 explores the joint combination of different descriptors by concatenation when both Figure-Border-Ground spatial pooling and cartesian-based Spatial Pyramid are applied. As shown in this table, besides pooling eMSIFT over Figure (with SP), Border and Ground regions, pooling eSIFT descriptors over the Figure, Border and Ground regions (without SP) and also eLBP over the Figure region (without SP) improves the average accuracy up to 75.86.

5.4 Results with CPMC Object Candidates

In order to validate the results of Section 5.3 over the ground truth object masks in a more realistic scenario, we evaluate our two main contributions (partitioning the image into three regions and using Spatial Pyramid over the Figure region) over CPMC object

Figure	SP(F)	Border	Ground	AAC
eSIFT+eMSIFT+eLBP			eSIFT	72.98 [CCBS12]
eMSIFT	eMSIFT	eMSIFT	eMSIFT	72.68
eSIFT+eMSIFT	eMSIFT	eMSIFT	eMSIFT	74.04
eSIFT+eMSIFT	eMSIFT	eMSIFT+eSIFT	eMSIFT+eSIFT	74.83
eSIFT+eMSIFT+eLBP	eMSIFT	eMSIFT+eSIFT	eMSIFT+eSIFT	**75.86**

Table 5.5: Gain of introducing the Border for pooling, applying the cartesian-based Spatial Pyramid over the Figure and combining eSIFT, eMSIFT and eLBP. Results using GT masks. Training over train11 and evaluation over val11.

candidates. Note that there is a tight link between CPMC and the O2P-based architecture from [CCBS12] since these object candidates have been reranked and filtered based on the same features used for classification, i.e. O2P features.

5.4.1 Figure-Border-Ground spatial pooling

As done with ground truth object masks, the experiments have been carried out in Pascal VOC 2011 using first the train subset for training and the validation subset for evaluation. The partitioning of the image for each object candidate into the Figure, Border and Ground regions improves the performance up to 34.81 (with eSIFT) in comparison with the original partitioning into Figure and Ground regions which gives an average accuracy of 28.58 [CCBS12]. However, the improvement is not so relevant when eMSIFT descriptors are used, where the pooling over the three regions increases the performance in 1.82 points with respect to the pooling over only the Figure region (30.99). To our surprise, despite the outperformance of the pooling of eMSIFT over Figure with respect to the pooling of eSIFT over Figure and Ground and the preliminary results obtained using GT masks, it is the eSIFT descriptor the one that takes the most advantage from the Figure-Border-Ground partitioning when using CPMC object candidates. This is the reason why eSIFT will be pooled over Border and Ground regions instead of eMSIFT in the following experiments.

Next, we have performed experiments pooling the three different descriptors (eSIFT, eMSIFT and eLBP) over the three proposed regions. The original performance achieved in [CCBS12] is 37.15, which results from pooling eSIFT over Figure and Ground regions and eMSIFT and eLBP over the Figure region. Our results from Table 5.6 show that using the partitioning of the image into three regions for pooling such a combination of the descriptors increases the average accuracy up to 38.91, which represents an increase of 1.76 points.

Once assessed the proposals over the validation subset, the experiments are validated over the test subset in *comp5* and *comp6* in order to be comparable with other state-of-the-art techniques.

For *comp5*, the experiments have been carried out using only the segmentation annotations available for the train and validation sets of the segmentation challenge, discarding the bounding box annotations provided for the train and validation sets of the detection challenge. The comparison between Figure-Ground and Figure-Border-Ground poolings is shown in Table 5.7 for both Pascal VOC 2011 and 2012. All these experiments have

Figure	Border	Ground	AAC
eSIFT+eMSIFT+eLBP		eSIFT	37.15 [CCBS12]
eSIFT+eMSIFT	eSIFT	eSIFT	37.72
eSIFT+eMSIFT+eLBP	eSIFT	eSIFT	**38.91**

Table 5.6: Gain of introducing the Border region and combining eSIFT, eMSIFT and eLBP. Results using CPMC object candidates. Training over train11 and evaluation over val11.

	F-G[CCBS12]	F-B-G	[LCLS13]	[XSF$^+$12]
VOC11	38.8	43.8	**48.8**	–.-
VOC12	39.9	42.2	**47.5**	47.3

Table 5.7: Results using CPMC object candidates for *comp5* and different image representations: Figure-Ground (F-G), Figure-Border-Ground (F-B-G). Results for the Pascal VOC2011 and VOC2012 Segmentation challenges.

been performed pooling eSIFT, eMSIFT and eLBP over Figure and only eSIFT over Border and Ground. The partitioning of the image into three regions (Figure-Border-Ground) gives the best performance, improving the average accuracy classification 5.0 and 2.3 points with respect to the Figure-Ground pooling for VOC 2011 and VOC 2012 respectively. Note that other results given by the state-of-the-art techniques [LCLS13, XSF$^+$12] have been obtained by using the bounding box annotations from the detection challenge, which is out of the scope of this dissertation. The experiments performed in [CLS12] with Pascal VOC 2010 showed that including the bounding boxes from the detection dataset in the training data resulted in a further 4% performance improvement. Analyzing the results by categories, the Figure-Border-Ground image partitioning improves the classification accuracy in 17 out of 20 categories in VOC 2011. In VOC 2012, the Figure-Border-Ground approach improves the accuracy in 13 out of 20 categories. Figure 5.3 and Table 5.8 show the results of the three configurations by categories.

Regarding *comp6*, the results for both VOC 2011 and VOC 2012 are shown in Table 5.9. In both cases, the partitioning of the image into Figure-Border-Ground improves the average accuracy in 1.4 points with respect to the Figure-Ground pooling. The results obtained in VOC 2012 achieved the same performance as the approach in [YBS13] (48.1), being both the second best results obtained without using deep learning techniques. Analyzing the results by categories (see Figure 5.4 and Table 5.10), the Figure-Border-Ground representation improves the accuracy in 16 out of 20 categories in VOC 2011 and in 13 out of categories in VOC 2012. Therefore, the analysis by categories already done for *comp5* is consistent with the results achieved in *comp6*.

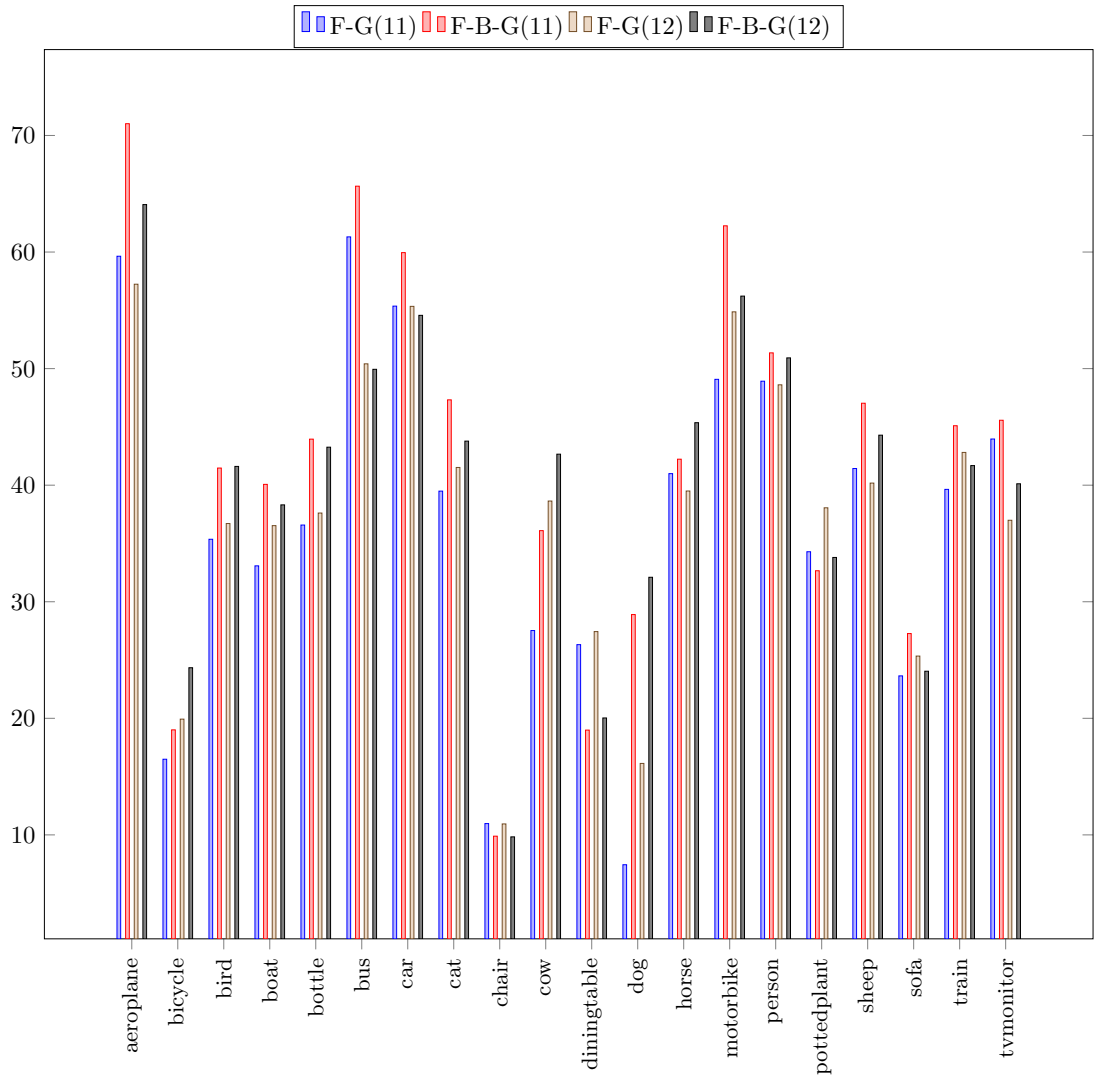


Figure 5.3: Accuracy Classification by Categories using CPMCs in *comp5*. Results by categories for baseline (Figure-Ground representation) and Figure-Border-Ground representation for VOC2011 and VOC2012 *comp5*.

	VOC2011			VOC2012		
	F-G [CCBS12]	F-B-G	SP(F)-B-G	F-G [CCBS12]	F-B-G	SP(F)-B-G
aeroplane	59.6	**71.0**	65.1	57.2	**64.1**	56.2
bicycle	16.5	**19.0**	12.6	19.9	**24.3**	15.6
bird	35.4	**41.5**	39.6	36.7	41.6	**43.0**
boat	33.1	**40.1**	37.5	36.5	**38.3**	29.2
bottle	36.6	44.0	**45.1**	37.6	43.3	**51.4**
bus	61.3	**65.7**	59.0	**50.4**	49.9	50.2
car	55.4	**60.0**	58.6	55.3	54.6	**58.5**
cat	39.5	47.3	**48.9**	41.5	43.8	**49.7**
chair	**11.0**	9.9	9.7	**10.9**	9.8	9.2
cow	27.5	36.1	**41.8**	38.6	42.7	**44.6**
diningtable	**26.3**	19.0	18.5	**27.4**	20.0	13.5
dog	7.4	**28.9**	23.5	16.1	**32.1**	26.2
horse	41.0	**42.2**	27.0	39.5	**45.4**	39.0
motorbike	49.1	**62.3**	49.5	54.9	**56.2**	52.8
person	48.9	**51.4**	49.1	48.6	**50.9**	50.2
pottedplant	**34.3**	32.7	32.6	**38.1**	33.8	37.6
sheep	41.4	**47.0**	44.5	40.2	44.3	**46.8**
sofa	23.6	**27.3**	10.8	**25.3**	24.0	11.8
train	39.6	**45.1**	42.9	42.8	41.7	**44.0**
tvmonitor	44.0	45.6	**45.9**	37.0	40.1	**42.5**
Average	38.8	**43.8**	40.3	39.9	**42.2**	40.8

Table 5.8: Accuracy Classification by Categories using CPMCs in *comp5*. Results by categories for baseline (Figure-Ground representation), Figure-Border-Ground and SpatialPyramid(Figure)-Border-Ground representation for VOC2011 and VOC2012 *comp5*.

	F-G [CCBS12]	F-B-G	[HAGM14]	[DCYY14]	[YBS13]
VOC2011	47.6	**49.0**	–.-	–.-	–.-
VOC2012	46.7	48.1	**51.6**	50.0	48.1

Table 5.9: Results using CPMC object candidates for *comp6* and different image representations: Figure-Ground (F-G), Figure-Border-Ground (F-B-G). Results for the Pascal VOC2011 and VOC2012 Segmentation challenges.

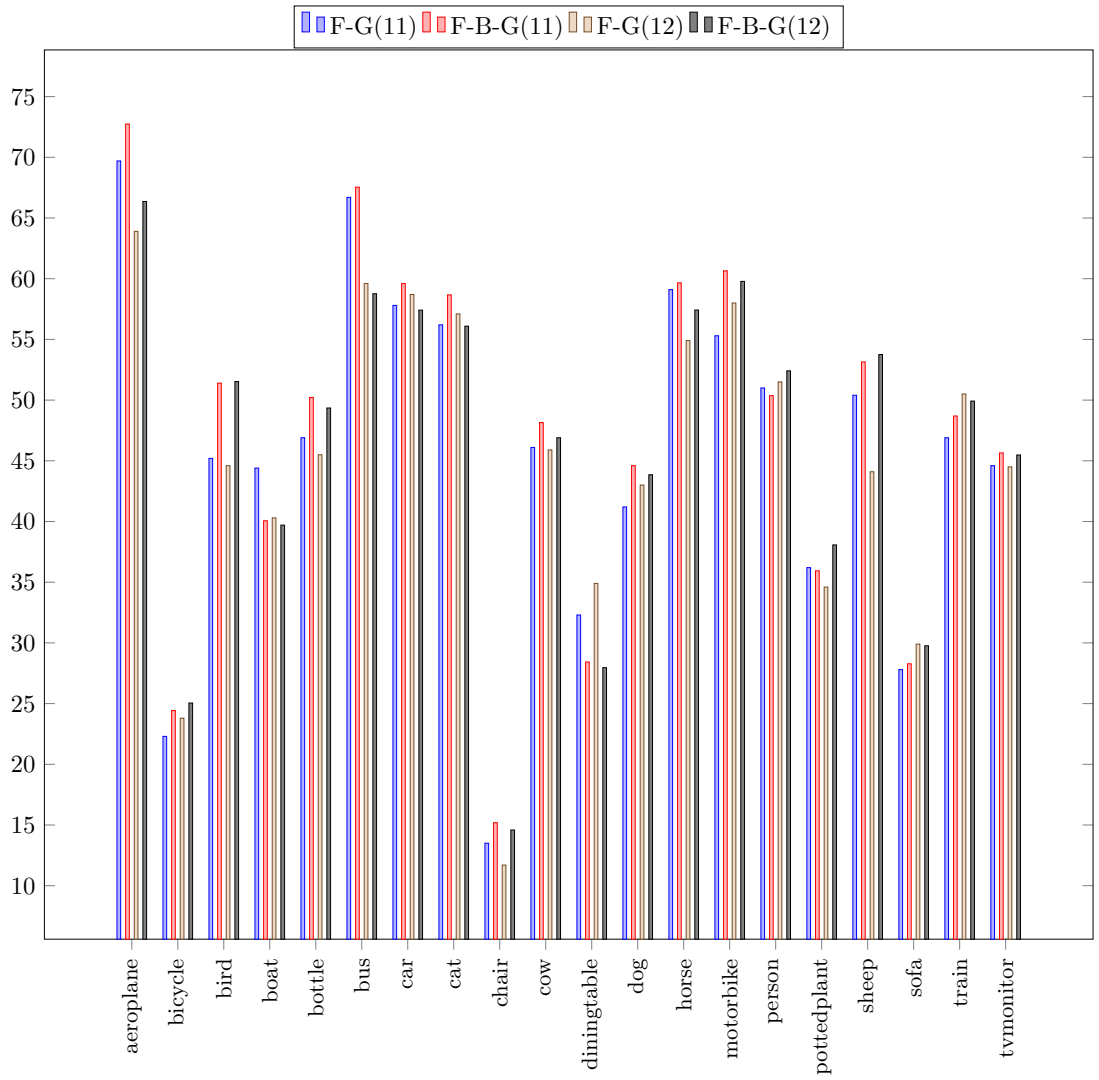


Figure 5.4: Accuracy Classification by Categories using CPMCs in *comp6*. Results by categories for baseline (Figure-Ground representation) and Figure-Border-Ground representation for VOC2011 and VOC2012 *comp6*.

	VOC2011		VOC2012	
	F-G [CCBS12]	F-B-G	F-G [CCBS12]	F-B-G
aeroplane	69.7	**72.7**	63.9	**66.4**
bicycle	22.3	**24.4**	23.8	**25.1**
bird	45.2	**51.4**	44.6	**51.5**
boat	**44.4**	40.1	**40.3**	39.7
bottle	46.9	**50.2**	45.5	**49.4**
bus	66.7	**67.5**	**59.6**	58.8
car	57.8	**59.6**	**58.7**	57.4
cat	56.2	**58.7**	**57.1**	56.1
chair	13.5	**15.2**	11.7	**14.6**
cow	46.1	**48.2**	45.9	**46.9**
diningtable	**32.3**	28.4	**34.9**	28.0
dog	41.2	**44.6**	43.0	**43.9**
horse	59.1	**59.7**	54.9	**57.4**
motorbike	55.3	**60.7**	58.0	**59.8**
person	**51.0**	50.4	51.5	**52.4**
pottedplant	**36.2**	35.9	34.6	**38.1**
sheep	50.4	**53.2**	44.1	**53.8**
sofa	27.8	**28.3**	**29.9**	29.8
train	46.9	**48.7**	**50.5**	49.9
tvmonitor	44.6	**45.7**	44.5	**45.5**
Average	47.6	**49.0**	46.7	**48.1**

Table 5.10: Accuracy Classification by Categories using CPMCs in *comp6*. Results by categories for baseline (Figure-Ground representation) and Figure-Border-Ground representation for VOC2011 and VOC2012 *comp6*.

Figure	SP(F)	Border	Ground	AAC
eSIFT			eSIFT	28.58 [CCBS12]
eSIFT	eSIFT			34.56
eSIFT		eSIFT	eSIFT	34.81
eSIFT+eMSIFT+eLBP			eSIFT	37.15 [CCBS12]
eSIFT	eSIFT	eSIFT	eSIFT	37.38
eSIFT+eMSIFT	eSIFT	eSIFT	eSIFT	39.21
eSIFT+eMSIFT+eLBP	eSIFT	eSIFT	eSIFT	**39.62**

Table 5.11: Results using CPMC object candidates for different image spatial representations and combining eSIFT, eMSIFT and eLBP and applying the cartesian-based Spatial Pyramid over Figure. Training over train11 and evaluation over val11.

5.4.2 Contour-based SP

Following the same methodology as the experiments with the ground truth masks, once the partitioning of the image into three regions has been validated for CPMC object candidates, we proceed to validate the use of the Spatial Pyramid over the Figure region. As before, the experiments are first evaluated over the validation subset. Using the cartesian-based Spatial Pyramid over the Figure region with the eSIFT descriptor and ignoring both the Border and Ground regions increases the performance up to 34.56, which is close to the improvement also achieved by the partitioning of the image into three regions (34.81). As shown in Table 5.11, both proposals result in a significant outperformance with respect to the Figure-Ground spatial pooling baseline (28.58).

Applying both proposals, i.e. the cartesian-based Spatial Pyramid over the Figure region and the partitioning of the image into Figure, Border and Ground regions, results in an average accuracy of 37.38. Notice that this result has been achieved using only eSIFT descriptor, whereas the best performance achieved in [CCBS12] is 37.15, using a combination of eSIFT, eMSIFT and eLBP. An average accuracy of 39.62 is achieved when the three descriptors are combined with the use of the three regions and the cartesian-based Spatial Pyramid (see Table 5.11).

For *comp5*, adding the cartesian-based Spatial Pyramid over the Figure region decreases the performance in 3.5 points for VOC 2011 (40.3) and 1.4 points for VOC 2012 (40.8). This decrease of the average accuracy was not expected based on the tendency shown in the previous experiments using the train set for training and the val set for evaluation for both ground truth object masks and CPMC object candidates. The use of the Spatial Pyramid over the Figure region only improves the accuracy in 4 categories in VOC 2011 and in 8 categories in VOC 2012.

Regarding *comp6*, having seen the decrease of the performance for *comp5* when Spatial Pyramid is used, we have decided to only submit to the Pascal VOC evaluation server the experiment using the Figure-Border-Ground image partitioning explained in Section 5.4.1.

5.4.3 Qualitative assessment

In this section, we show visual results for CPMC. We compare the baseline Figure-Ground spatial pooling [CCBS12] with our proposed Figure-Border-Ground spatial pooling, which give us the best average accuracy classification. The results have been filtered to only detect meaningful examples. This selection has been performed by choosing the results that fullfill the following requirement:

$$\frac{N_d}{N_d + N_s} > 0.5 \tag{5.1}$$

where N_d is the number of pixels which have been assigned different category labels for the two configurations being compared, and N_s is the number of pixels which have not been assigned to the *background* category but to the same category for both configurations. Notice that the fact of not considering the pixels assigned to the *background* category allows us to detect examples where the detected objects are small with respect to the whole image size. This way, the measure is normalized by the area of the pixels labeled with one out of the 20 visual categories. Next, in Figures 5.5 and 5.6 we only show a subset of this selection. The whole selection can be visualized in [VGNV+].

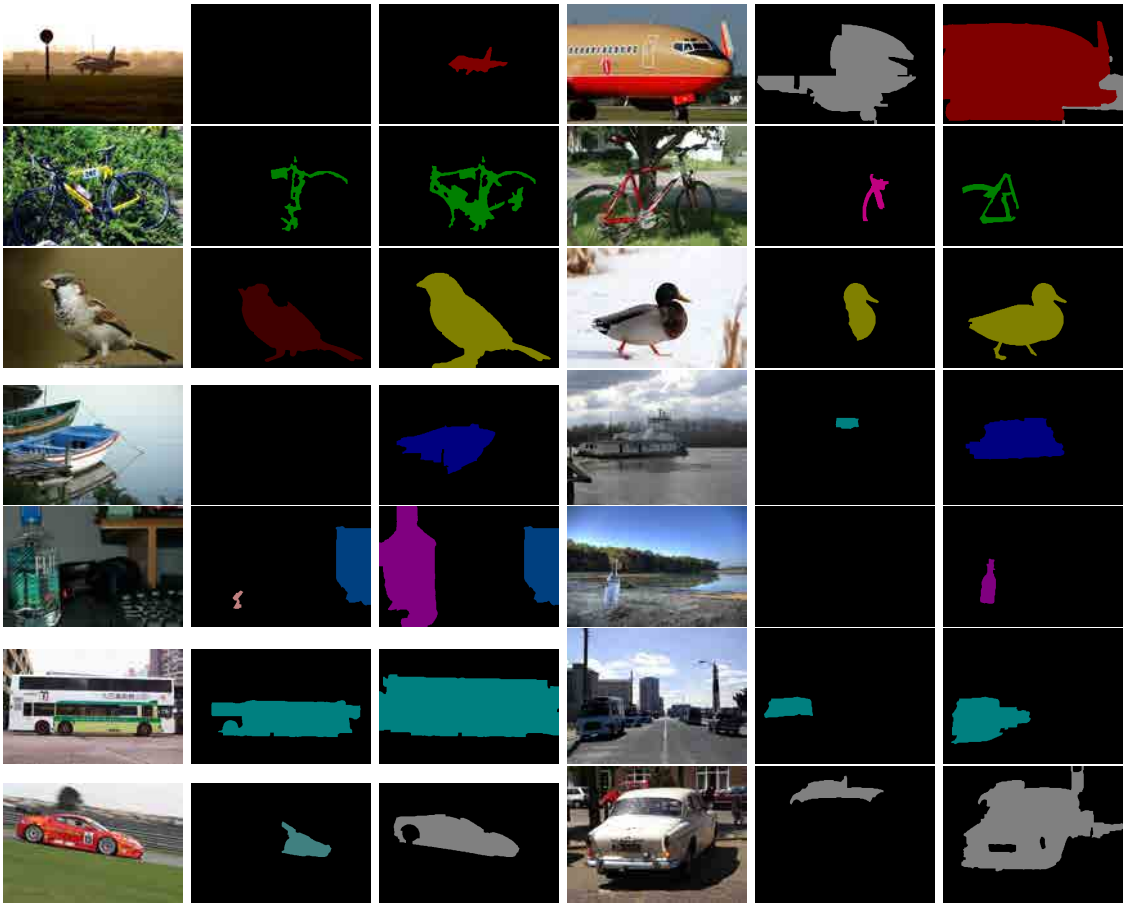


Figure 5.5: Visual results with CPMC candidates. First and fourth columns: images to be semantic segmented. Second and fifth columns: solution based on a F-G spatial pooling [CCBS12]. Third and last columns: solution based on a F-B-G spatial pooling.

Figure 5.6: Visual results with CPMC candidates. First and fourth columns: images to be semantic segmented. Second and fifth columns: solution based on a F-G spatial pooling [CCBS12]. Third and last columns: solution based on a F-B-G spatial pooling.

5.5 Results with MCG Object Candidates

Our spatial pooling approach has also been checked in another state-of-the-art object candidate generation: Multiscale Combinatorial Grouping (MCG) [APTB+14]. The experiments have been carried out in Pascal VOC 2011 using the train subset for training and the validation subset for evaluation. When the baseline solution given by [CCBS12] based on O2P features pooled over Figure-Ground is applied over MCGs instead of CPMCs, the average accuracy drops to 30.88 with respect to the 37.15 with CPMCs.

This drop in the performance was not expected initially since, according to [APTB+14], for the 150 top-ranked object candidates both techniques give a similar performance for segmentation (without considering recognition). We believe that such a difference in the performance regarding the semantic segmentation is due to the fact that CPMC have been specifically reranked for the O2P-based architecture proposed in [CCBS12]. Although about 800 CPMC generic object candidates per image are extracted and ranked based on mid-level descriptors and Gestalt features, a linear regressor also based on the O2P features is learned to rerank and filter them to generate the final pool of up to 150 CPMCs used in [CCBS12], following the same methodology as in [CLS12]. Therefore, the features used for classification (O2P) are also used for CPMC selection. On the other hand, MCG object candidates are ranked based only on mid-level descriptors and Gestalt features.

However, we have also checked our spatial pooling proposals over the 150 top-ranked MCG object candidates. The Figure-Border-Ground spatial pooling increases the performance up to 34.09, which represents a gain of 3.21 points with respect to the Figure-Ground spatial pooling (30.88). For such a spatial pooling, the classification accuracy is improved for 15 out of 20 categories.

Furthermore, when the cartesian-based Spatial Pyramid is applied over the Figure region besides using the Figure-Border-Ground spatial pooling, the average accuracy is increased up to 36.10, a gain of 2.01 points with respect to the Figure-Border-Ground pooling (34.09) and 5.22 points with respect to the Figure-Ground pooling (30.88). Applying the cartesian-based SP improves the accuracy for 16 out of 20 categories with respect to the Figure-Border-Ground pooling and for 19 out of 20 categories with respect to the original Figure-Ground pooling. More detail about the analysis by categories is given in Figure 5.7 and Table 5.12.

Although the results given by MCGs are worse than the ones achieved with CPMCs, we consider that these experiments illustrate the robustness of our spatial pooling contributions with object candidates for semantic segmentation.

5.5.1 Qualitative assessment

In this section, we show visual results for MCG. We perform a comparison between the baseline Figure-Ground spatial pooling [CCBS12] and the combination of our two proposed spatial configurations: Figure-Border-Ground spatial pooling with a cartesian-based Spatial Pyramid applied over the Figure region pool. The results have been filtered to only detect meaningful examples by choosing the results that fullfill the requirement previously stated in Equation 5.1. Next, in Figures 5.8 and 5.9 we only show a subset of this selection. The whole selection can be visualized in [VGNV+].

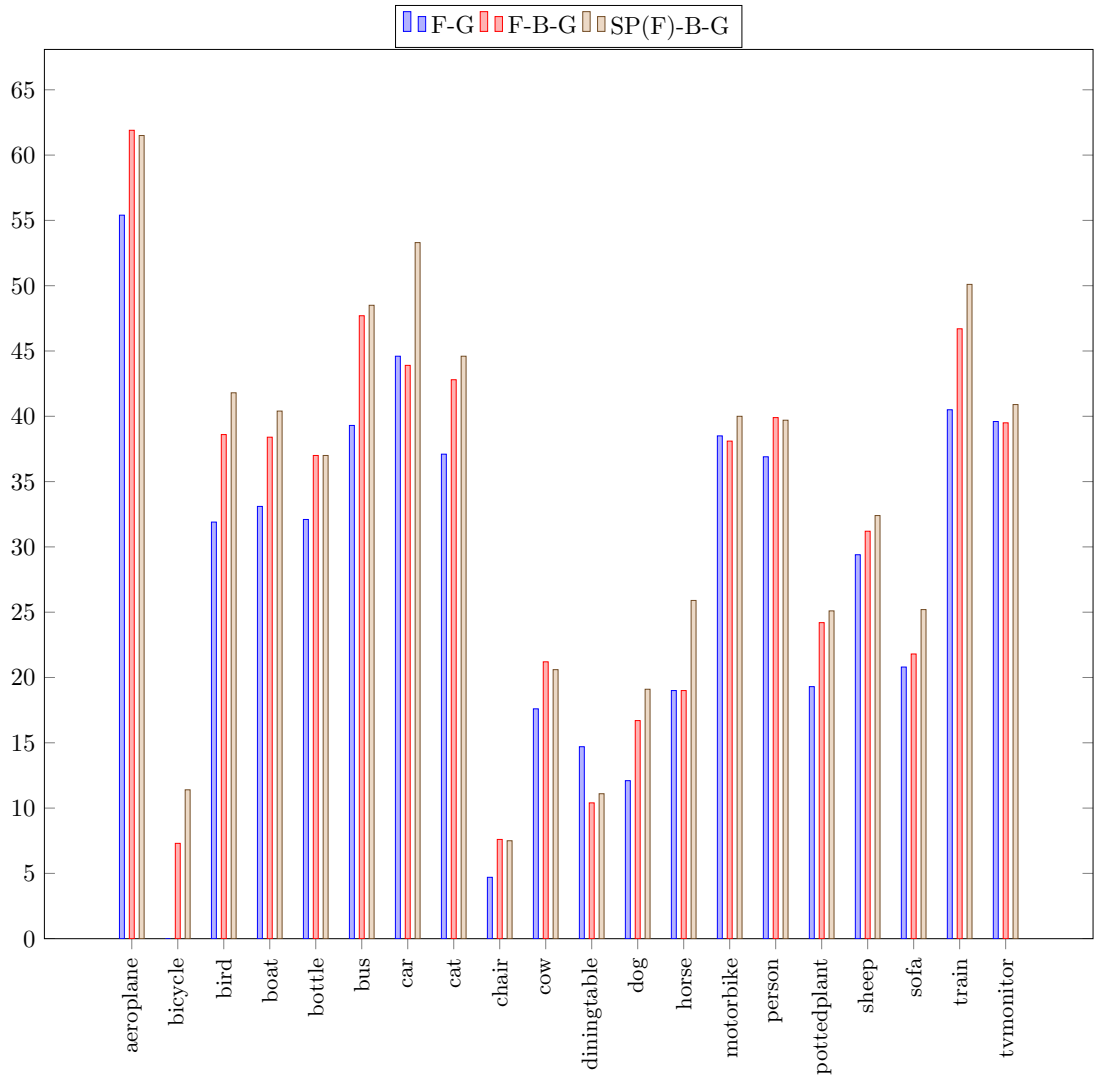


Figure 5.7: Accuracy Classification by Categories using MCGs. Results by categories for baseline (Figure-Ground representation), Figure-Border-Ground representation and SpatialPyramid(Figure)-Border-Ground representation using MCG object candidates. Training over train11 and evaluation over val11.

	F-G	F-B-G	SP(F)-B-G
aeroplane	55.4	**61.9**	61.5
bicycle	0.0	7.3	**11.4**
bird	31.9	38.6	**41.8**
boat	33.1	38.4	**40.4**
bottle	32.1	**37.0**	**37.0**
bus	39.3	47.7	**48.5**
car	44.6	43.9	**53.3**
cat	37.1	42.8	**44.6**
chair	4.7	**7.6**	7.5
cow	17.6	**21.2**	20.6
diningtable	**14.7**	10.4	11.1
dog	12.1	16.7	**19.1**
horse	19.0	19.0	**25.9**
motorbike	38.5	38.1	**40.0**
person	36.9	**39.9**	39.7
pottedplant	19.3	24.2	**25.1**
sheep	29.4	31.2	**32.4**
sofa	20.8	21.8	**25.2**
train	40.5	46.7	**50.1**
tvmonitor	39.6	39.5	**40.9**
Average	30.9	34.1	**36.1**

Table 5.12: Accuracy Classification by Categories using MCGs. Results by categories for baseline (Figure-Ground representation), Figure-Border-Ground and SpatialPyramid(Figure)-Border-Ground representation using MCG object candidates. Training over train11 and evaluation over val11.

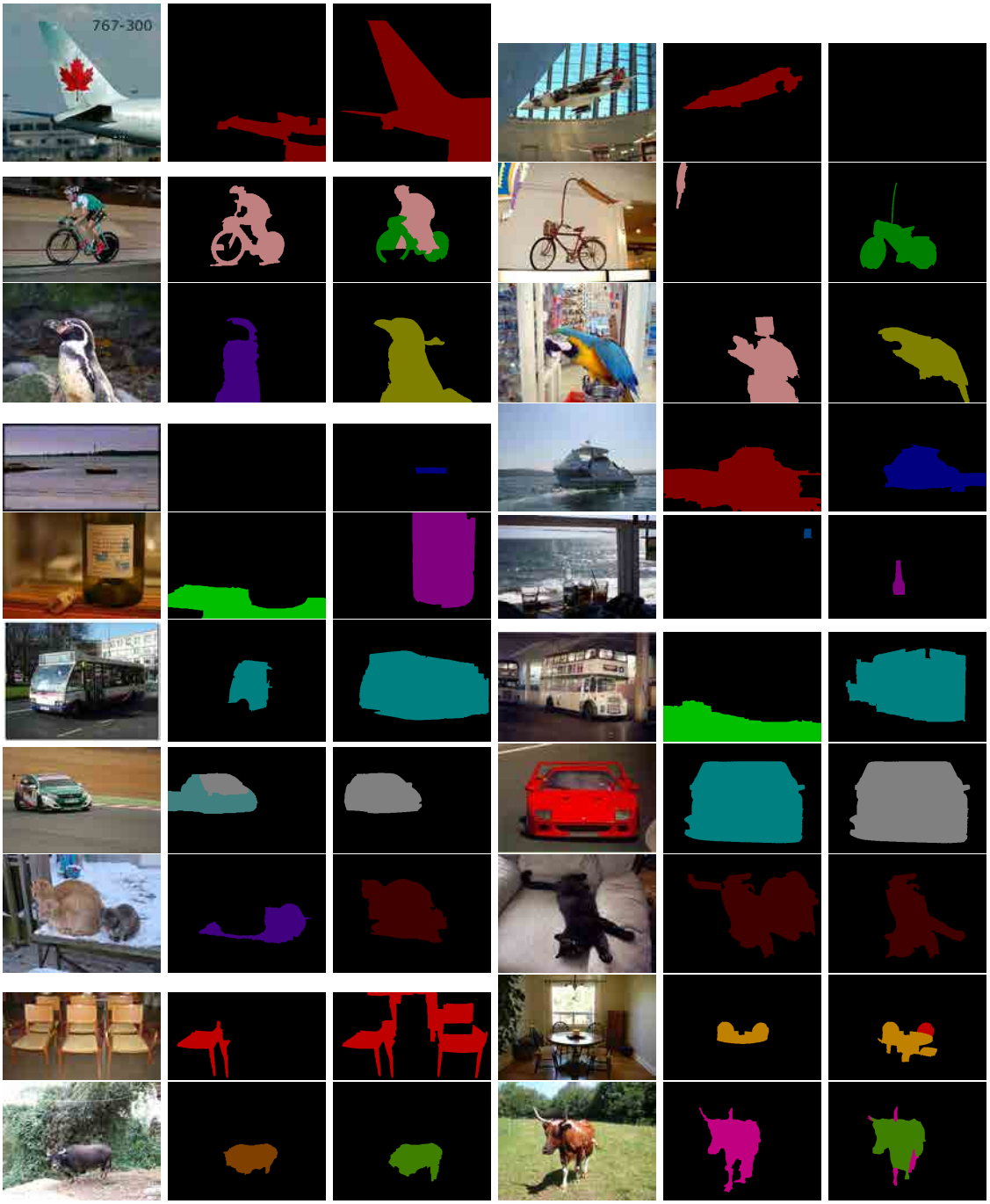


Figure 5.8: Visual results with MCG candidates. First and fourth columns: images to be semantic segmented. Second and fifth columns: solution based on a F-G spatial pooling [CCBS12]. Third and last columns: solution based on a F-B-G spatial pooling and cartesian-based Spatial Pyramid over Figure.

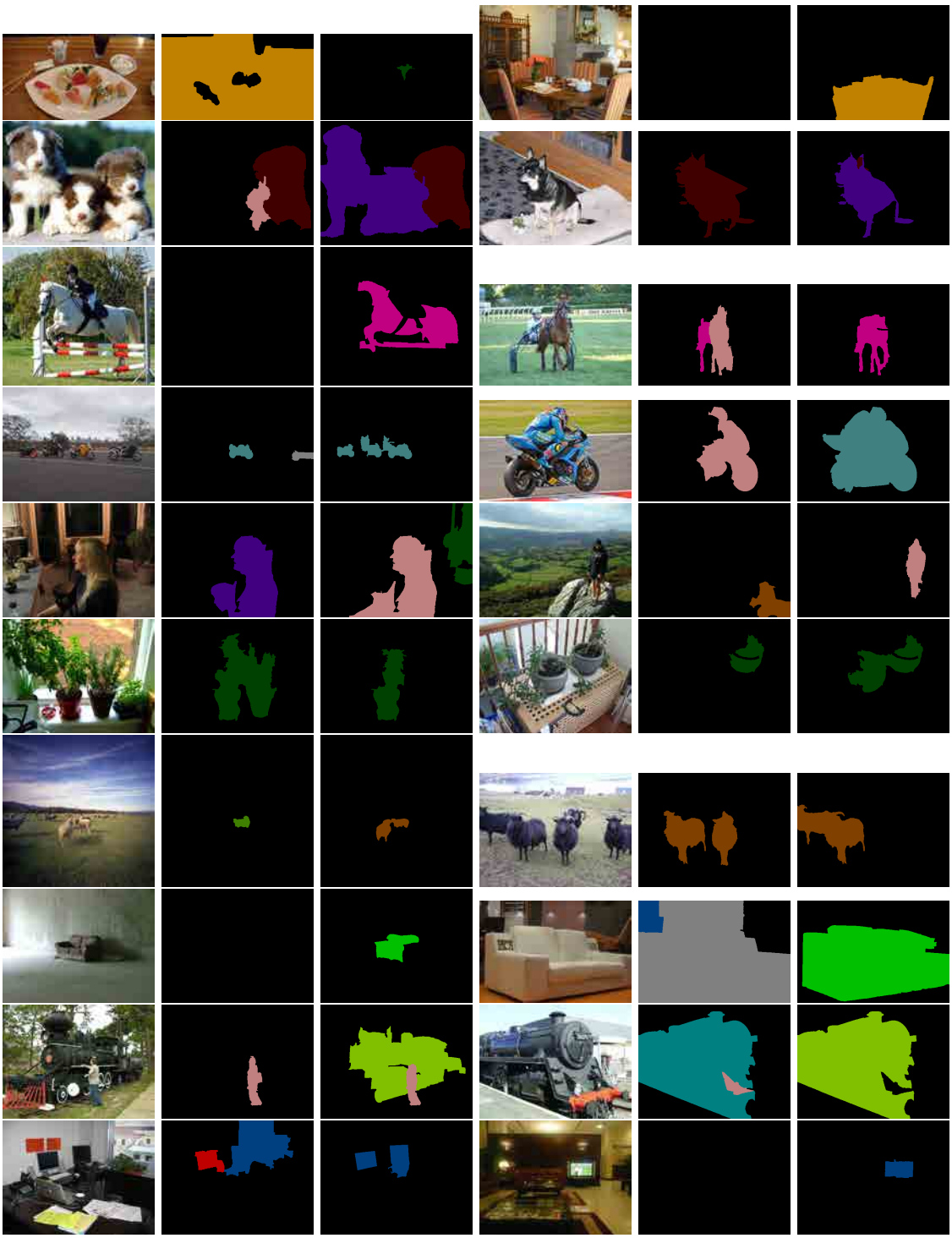


Figure 5.9: Visual results with MCG candidates. First and fourth columns: images to be semantic segmented. Second and fifth columns: solution based on a F-G spatial pooling [CCBS12]. Third and last columns: solution based on a F-B-G spatial pooling and cartesian-based Spatial Pyramid over Figure.

Conclusions and Future Work

6

We have presented two contributions for improving the spatial pooling beyond the classic Figure-Ground partitioning to solve the semantic segmentation problem, resulting in a publication on the International Conference on Image Processing 2015 [VGiNV+15].

On the one hand, we have extended the original idea from [USS12] where a Figure-Border-Ground spatial pooling is applied in an ideal situation to a more realistic scenario with the use of object candidates. This richer spatial pooling has been tested with state-of-the-art techniques (CPMC and MCG object candidates and O2P features) and has led to improvements of the average accuracy in all scenarios. Futhermore, the results obtained when Figure-Border is used indicates that the richest contextual information for object recognition is located in the very near neighbourhood of the object itself.

On the other hand, we have explored two different configurations (crown-based and cartesian-based) of Spatial Pyramid applied over the Figure region. Although this richer spatial pooling increased the performance for the ideal scenario and also when the system was evaluated over the validation subset, this tendency was not kept when it was eventually assessed over the test subset.

A more extended analysis about other ways of masking, as the post-masking methods proposed by [TLF10] and [TKSMN13], is left as future work. Furthermore, it would be also interesting to perform an analysis about the impact that would have replacing the ground truth masks and the object candidates by their bounding boxes to see either the shape or the context plays an important role in object recognition.

Regarding the proposed Cartesian-based Spatial Pyramid, it would be also interesting to align the Cartesian axis to the major and minor orientation axis of the region of interest in order to be invariant to rotation.

Finally, we would like to mention that from the time this part of the dissertation was written to the time of publication, there have been many pixel-wise deep learning techniques that have achieved outstanding results without using object proposals [FCNL13, ZJRP+15, LXL+15, PCMY15]. However, instance-aware semantic segmentation is still a challenging problem and the recently established COCO [LMB+14] dataset and competition only accept instance-aware semantic results. One of the most recent approaches [DHS16] addresses the semantic segmentation problem with deep learning in 3 stages: differentiating instances, estimating masks and categorizing objects. On the one hand, the two first stages address what has been referred in this part of the dissertation as the first challenge, where we decided to use object proposal techniques such as CPMC or MCG. On the other hand, the third stage addresses the second challenge, i.e. labeling the regions with the appropriate object class, which has been the scope of this part.

Part II

Multiresolution co-clustering for uncalibrated multiview segmentation

Introduction

7

In this part, we present a technique for coherently co-clustering uncalibrated views of a scene for both generic and semantic segmentation. Having images of the same scene taken from different viewpoints allows us to extend the problem of semantic segmentation stated in Part I for a single image to a set of images with a high spatial correlation, which will be referred to as multiview semantic segmentation.

Semantic segmentation algorithms have drastically increased their performance since the introduction of Convolutional Neuronal Networks (CNNs) for this task [FCNL13, HAGM14, LSD15, ZJRP+15]. CNNs require large amounts of annotated visual content to train their parameters, but thanks to global scale labels like the ones provided in the ImageNet dataset [DDS+09], combined with pixel-wise annotations, like the ones in the PASCAL Visual Object Challenge (SegVOC12) [EVGW+] database or in Microsoft Common Objects in Context (CoCo) [LMB+14], the training of CNNs for semantic segmentation has been possible.

However, several authors have analyzed the limitations of such annotated databases paying attention, among other aspects, to the generalization across datasets and to the balance, location and size of the annotations [TE11, PTG15]. As a result, a strong bias towards some specific objects has been reported (e.g.: 25-30% of the instances are from the *person* class [PTG15]). On the contrary, the databases do not correctly represent the high variability of other classes (strong differences among instances of a concept, i.e. intra-class variability, or among views of a given instance, i.e. view variability). This leads to a large variation in semantic segmentation performance for different classes as can be observed in SegVOC12 leaderboard. In this part of the dissertation, we focus on the view variability problem, as shown in Figure 7.1 with an example where the accuracy of the semantic segmentation varies significatively with the viewpoint. [ZJRP+15], one of state-of-the-art techniques (average accuracy classification score of 74.7) with available implementation, is independently applied to each viewpoint image for semantic segmentation. Whereas accuracy classification scores above 90 have been achieved in SegVOC12 for some categories such as aeroplane, bird, bus or cat, accuracy classification scores for other categories such as bicycle, chair or sofa are still below 65.

The little availability of annotated multiview datasets has not allowed the training of end-to-end CNNs for such a problem yet. However, putting in correspondence the objects from the different viewpoint images in an unsupervised way, i.e. without considering semantics, and including later the semantic information obtained for each image independently allows palliating the view variability problem. Figure 7.2 shows how we can take advantage of having a set of partitions where their regions have been clustered and put in correspondence along the different views to generate a better semantic segmentation. Using such spatial correlation (see the third row of Figure 7.2), the semantic

Figure 7.1: Changes in semantic segmentation accuracy due to viewpoint variability for GardenChair dataset from [KSS12]. First row: original views. Second row: semantic segmentations obtained with the CNN proposed in [ZJRP+15].

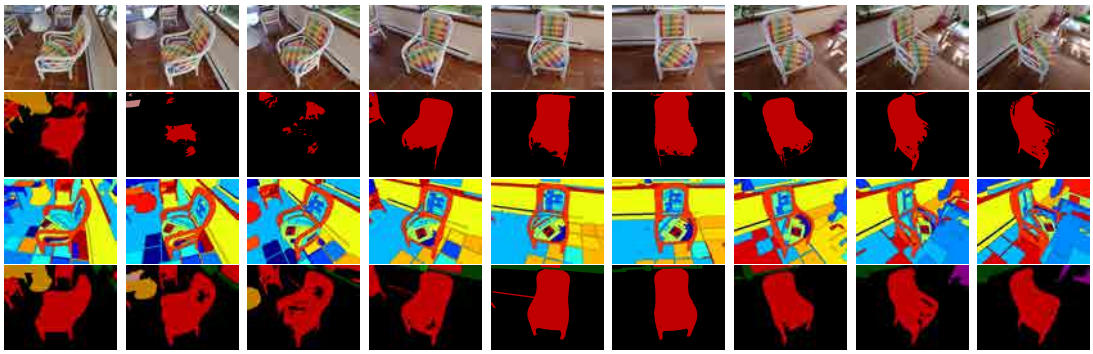


Figure 7.2: Levering generic co-clustering techniques and independent semantic segmentation for a coherent multiview semantic segmentation. First row: original views. Second row: semantic segmentations obtained with the CNN proposed in [ZJRP+15]. Third row: Proposed multiresolution generic co-clustering with automatic resolution selection (Section 8.3.4.1). Fourth row: Proposed automatic multiview semantic segmentation (Section 8.3.5). Note the improvements in the object representation in several views of the semantic segmentation.

segmentation is significatively improved, specially for the views where the object had been hardly detected (compare the second and the fourth rows of Figure 7.2). How the semantic segmentation given by [ZJRP+15] and generic segmentation techniques can be combined to obtain better semantic segmentation is one of the contributions of this part.

The task of multiview segmentation, which can be very accurately solved when the camera parameters are known [KSS12, DFB+13], becomes much more complicated when these parameters are not available. Several approaches can be followed to tackle the generic multiview segmentation problem: (*i*) extending video segmentation techniques such as [GKHE10, XXC12, GCS13], (*ii*) using co-segmentation techniques such as [JBP12, KX12], or (*iii*) using co-clustering techniques such as [VB10, GVB11, VAM15]. There are subtle differences between the three approaches.Whereas video segmentation techniques are focused on video sequences, where there is an intrinsic temporal correlation, co-clustering and co-segmentation techniques are applied to a broader domain that includes any set of related images. Furthermore, co-segmentation techniques, in contrast with co-clustering and video segmentation techniques, aims at a figure-ground segmentation. In [VAM15], it was reported that, in the context of video segmentation of scenes with little motion, co-clustering techniques outperform other approaches. However, we cannot assume that co-clustering techniques are also outstanding in the multiview context. Therefore, it seems necessary revisiting the type of techniques that underperformed

in [VAM15].

In the next sections we give an overview of the three approaches and afterwards we introduce some definitions and notation that will be useful for the following chapters. The rest of Part II is structured as follows. In Chapter 8, we review in more detail co-clustering techniques and we extend them to the multiview scenario for both generic and semantic multiview segmentation. Then, in Chapter 9, state-of-the-art techniques with available implementations are assessed and compared with our proposed co-clustering techniques. Finally, the conclusions are drawn in Chapter 10.

7.1 Video segmentation techniques

Video segmentation techniques are the family of techniques that aim at a coherent segmentation of the frames of a video sequence by exploiting the temporal correlation existing across them. Next, we give a brief overview of three state-of-the-art video segmentation techniques with available implementations that will be compared with state-of-the-art co-segmentation and co-clustering techniques for multiview segmentation in Chapter 9.

In [GKHE10], the authors present a video segmentation technique based on a hierarchical graph-based algorithm. The video frames are oversegmented and a volumetric video graph is built by grouping the regions by appearance and using dense optical flow to establish the temporal connections. The grouping of the regions according to the appearence criteria is applied to generate different levels of granularity, resulting in a hierarchical video representation. A global optimization process is performed for each level of granularity. Here, *global* refers to an optimization process that is jointly applied to all the frames.

A hierarchical video segmentation is also proposed in [XXC12] but, in this case, sequences are processed in bursts, leading to an iterative algorithm. Here, *iterative* refers to a forward-online optimization process, where each video frame is processed only once and does not change the segmentation of previous frame. Thanks to this iterative approach, it is possible to segment videos that otherwise could not be loaded into memory.

Video segmentation is also tackled in [GCS13] as an extension of the image approach in [AMFM11]. This image approach, named gPb-owt-ucm (overviewed in Section 2.1), is extended by including an optical flow channel so that pixels are merged considering also motion affinity besides texture, brightness and color. Furthermore, the authors propose a two-step framework. In the first step, a fine segmentation consisting of superpixels is obtained as a result of applying the previous motion-aware hierarchical image segmentation. Then, in the second step, between-frame affinities are also considered, which can be richer and more powerful than pixel-based affinities.

Some of the contributions for co-clustering techniques presented in Section 8.3 are inspired in some characteristics of the video segmentation techniques reviewed. First, the use of dense optical flow in [GKHE10] to establish connections between the frames. Our proposed co-clustering technique also use the optical flow to connect the different views. Second, the two-step framework from [GCS13] that allows the use of more powerful features in the second step. We also propose a two-step framework but with a different goal. In our proposed architecture, whereas in the first step the clustering of regions is con-

strained to nodes from a hierarchy of partitions towards a coarser resolution, the second step allows region mergings that are not present in the hierarchy. We also take advantage of the high quality of hierarchies in [AMFM11]. On the other hand, the main difference between co-clustering and video segmentation techniques is that the between-frame affinities of co-clustering techniques are based on a contour element additive representation. Our framework covers both iterative and global cases.

7.2 Co-segmentation techniques

Co-segmentation techniques aim at simultaneous segmentation of the same or similar objects that appear in a set of images. In constrast to video segmentation techniques, the set of images are not thought to belong to a video sequence and, therefore, there is no temporal correlation between them. Classically, co-segmentation techniques were designed to be applied over a set of images with the same or similar objects but a different background. However, since background does not change significatively in the multiview scenario that we are considering, we only review co-segmentation techniques that do not assume different backgrounds in the set of images.

The work in [JBP12] proposes an energy-maximization approach that can handle multiple classes and a large number of classes by combining spectral and discriminative clustering. Whereas spectral clustering aims at dividing each image into visually and spatially consistent regions, the discriminative clustering aims at maximizing class separability across images.

Another co-segmentation approach is presented in [KLH12], where it is proposed to build a graph with connections between regions of the same image (intra-image) and between regions of different images (inter-image). Intra-image connections are based upon hierarchical clustering, resulting in hierarchical constraints. Inter-image connections are only defined between the coarsest level of different image partitions. Given these connections, the images are segmented into foreground and background regions.

In contrast to classical approaches, the authors from [KX12] tackle a more realistic scenario where the objects are not assumed to appear over the entire image set. However, in this approach the user is required to provide the number of foreground objects in the set of images and a single segmentation is obtained.

As [KLH12], intra-image connections from the co-clustering technique proposed in [VAM15] are also based upon hierarchical clustering. However, the restriction to coarsest levels for inter-image connections is tackled defining the optimization problem over boundary segments. Whereas in [KX12] the user has to provide the number of objects, we propose a semantic-based automatic resolution selection method that does not require the interaction with the user. Regarding [JBP12], our approach is based on the hierarchies from [AMFM11], which are also based on spectral clustering. As before when analyzing video segmentation techniques, none of the reviewed co-segmentation techniques consider inter-image similarities based on the shape of boundary segments.

7.3 Co-clustering techniques

Co-clustering techniques are defined as the techniques aimed at a joint grouping of segments in the partitions of two or more closely-related images. In contrast with video segmentation techniques, images are not assumed to belong to a video sequence. Images can also be sections of 3D volumes, such as sections of a neuronal tissue acquired using an electron microscope in [VB10], or images taken from different viewpoints, such as the multiview segmentation problem that we are approaching. In contrast with co-segmentation techniques, the objective is not a foreground segmentation, but just having a coherent segmentation with correspondences across all images. Here *coherent* refers to regions in correspondence representing the same object parts along the different views or frames.

In a medical context, [VB10] addresses the co-clustering problem applied to electron microscopy images. They aim at maximizing the agreement between clusters of segments based on two region-based measures: pixel area overlap and merge-confidences computed by a boosted classifier based on color histogram comparison. The co-clustering is formulated as a quadratic optimization problem, specifically a Quadratic Semi-Assignment Problem, where coefficients in the quadratic function encode whether pairs of segments should belong to the same cluster or to different clusters. This NP-hard optimization problem is relaxed with Linear Programming using the work of [CGW].

In constrast with [VB10], the authors in [GVB11] propose a contour-based co-clustering. Region-based measures alone may not be ideal for shape comparison. Large differences in shapes between two regions may contribute very little to the region-based differences, but they could be semantically important. Furthermore, translations between images, which do not affect on the region shapes, may be drastically penalyzed if region-based measures such as pixel area overlap are considered.

Based on [GVB11], the authors in [VAM15] extend the contour-based co-clustering by including hierarchical constraints that exploit the tree information avoiding inconsistencies of previous co-clustering approaches and proposing an iterative approach for video segmentation that combines information at different resolutions. Furthermore, they also propose to use more descriptors for intra and inter similarity measures. In the context of segmenting video sequences with small variations, it was reported in [VAM15] that co-clustering techniques outperform other approaches such as video segmentation and co-segmentation techniques.

Since our work is based on extending the framework in [VAM15] to address the semantic multiview segmentation problem, Sections 8.1 and 8.2 provide more details about [GVB11] and [VAM15] respectively.

7.4 Definitions and Notation

We dedicate this section to introduce the notation and the definitions of some concepts that will be useful to follow the contour-based co-clustering framework explained in Chapter 8. Let us first introduce the concept of partition. A *partition* is defined as a division of an image into non-overlapping regions that cover the entire image domain. More formally, given an image on a domain $\Omega \subset \mathbb{R}^2$, a partition P is a set of N regions R_i such

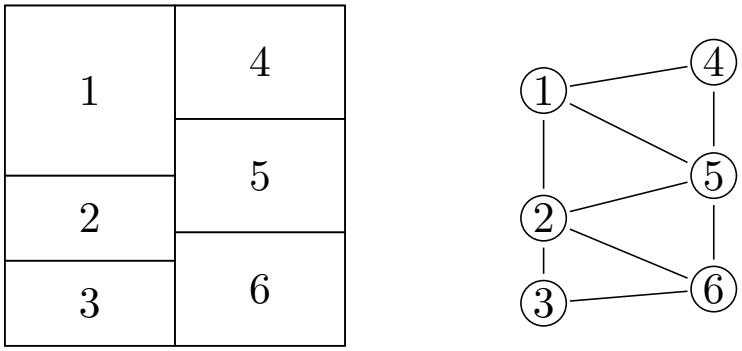


Figure 7.3: Leaves partition and its region adjacency graph (RAG).

that $\Omega = \bigcup_{i=1}^{N} R_i$ and $R_i \cap R_j = \emptyset \ \forall i \neq j$.

Now, we introduce the concepts of adjacency and region adjacency graph (RAG), which are associated with the concept of partition. Two regions R_i and R_j are considered *adjacent* if any pixel p_i from R_i has at least one pixel p_j from R_j among the 4-connected neighborhood of p_i. Following the example showed in Figure 7.3, we can state that regions R_1 and R_4 are adjacent, whereas R_1 and R_6 are not adjacent. If we represent each region as a node in a graph and then we connect with edges the pair of nodes representing adjacent regions, we obtain an unweighted undirected graph, which will be referred to as *region adjacency graph (RAG)*. In Chapter 8, the RAG definition will be extended to a set of partitions. The RAG is also represented in Figure 7.3.

Once reviewed the definition of partition, the concepts of hierarchy of partitions and merging sequence are introduced. If we have an initial partition of the image P_1, which will be referred to as *leaves partition*, and we iteratively merge the regions based on a similarity criteria, we obtain a set of increasingly coarser partitions $\{P_1^{(0)}, P_1^{(1)}, ..., P_1^{(N)}\}$, where $P_1^{(0)}$ represents the initial partition P_1. As a result of each merging, a hierarchical relationship is established between the regions merged, which will be referred to as *children nodes*, and the region resulting from the merging, which will be referred to as *parent node*. The children nodes of a parent nodes will be also referred to as *sibling nodes*. If the mergings are assumed to be binary, i.e. regions are merged by pairs, then the resulting structure is referred to as *Binary Partition Tree* [SG00]. Note that this assumption can be done without loss of generality, as any hierarchy can be transformed into a binary one. The use of a region merging algorithm over the leaves partition does not only results in a hierarchy, but also determines the order in which the mergings are done, which will be referred to as *merging sequence*.

Figure 7.4 shows an example of a leaves partition and its associated hierarchy representation. We have an initial partition, the leaves partition, which is represented by the regions $\{R_1, R_2, R_3, R_4, R_5, R_6\}$ (also referred to as leaves nodes). The rest of nodes in the hierarchy are parent nodes. Thus, node 7 is a parent node with two children nodes 1 and 2. Analogously, parent node 8 with nodes 7 and 3, parent node 9 with nodes 4 and 5, parent node 10 with nodes 9 and 6, and parent node 11 with nodes 8 and 10. Since node 11 has not any ancestor, it is also referred to as *root node*.

Note that the fact of having the merging sequence allows us to represent the hierarchy as a *dendrogram*, in which each parent node is also represented at a height inversely

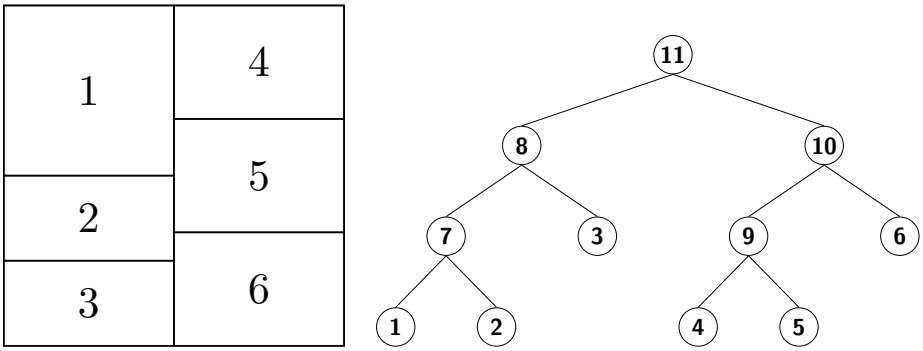


Figure 7.4: Leaves partition and a possible hierarchy representation. The structure of the hierarchy depends on the criteria of the merge algorithm.

proportional to the similarity between its two sibling nodes. Furthermore, the merging sequence also determines a set of increasingly coarser partitions so $P^{(i)}$ has exactly one region less than $P^{(i-1)}$ as a result of two regions from $P^{(i-1)}$ being merged to a parent region in $P^{(i)}$. This property allows to define a *cut* in the hierarchy to obtain a coarser partition with N_r regions in a univoque way as long as the merging sequence is respected.

For instance, following the example showed in Figure 7.4, let us suppose that the identifiers of the parents nodes represent the order in which have been merged. Therefore, the set of increasingly coarse partitions would be given by $P^{(0)} = \{R_1, R_2, R_3, R_4, R_5, R_6\}$, $P^{(1)} = \{R_7, R_3, R_4, R_5, R_6\}$, $P^{(2)} = \{R_8, R_4, R_5, R_6\}$, $P^{(3)} = \{R_8, R_9, R_6\}$, $P^{(4)} = \{R_8, R_{10}\}$ and $P^{(5)} = \{R_{11}\}$. Performing a cut in the hierarchy to obtain a partition with 3 regions univoquely results in the partition formed by nodes $\{8, 9, 6\}$. However, if the merging sequence is not considered, other partitions such as $\{R_7, R_3, R_{10}\}$ that respect the hierarchy of the nodes would be also possible.

Finally, we introduce a notation which will be useful to understand the formulation of the contour-based co-clustering as an optimization problem in Chapter 8. Let us define a boundary boolean variable $D_{i,j}$ that is true if the boundary between two adjacent leaves regions R_i and R_j is active, i.e. R_i and R_j are not merged, and false otherwise. We can associate one boundary variable with each edge of the RAG. Note that, given a leaves partition $P^{(0)}$, any coarser partition $P^{(i)}$ can be represented as a set of active and inactive boundary variables. For instance, and following the example shown in Figure 7.4, the partition $P^{(3)} = \{R_8, R_9, R_6\}$ can be represented as a binary vector $[D_{1,2}\ D_{1,4}\ D_{1,5}\ D_{2,3}\ D_{2,5}\ D_{2,6}\ D_{3,6}\ D_{4,5}\ D_{5,6}] = [0\ 1\ 1\ 0\ 1\ 1\ 1\ 0\ 1]$. This representation is shown in Figure 7.5.

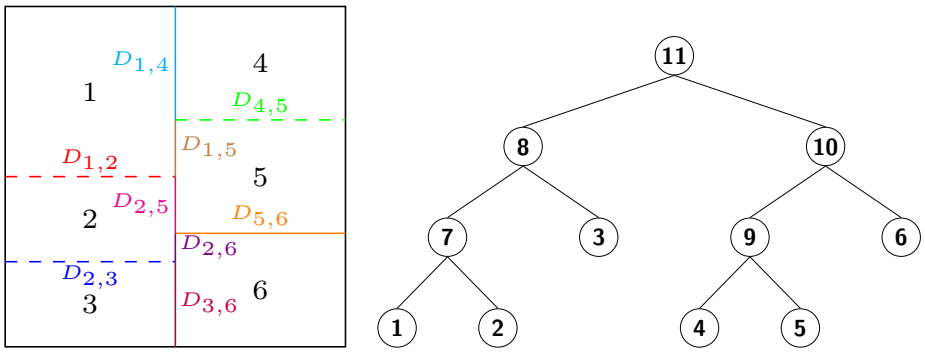


Figure 7.5: Representation of partition $P^{(3)} = \{R_8, R_9, R_6\}$ with respect to $P^{(0)}$ using boundary variables $D_{i,j}$. Dashed lines represent inactive boundary variables ($D_{1,2}$, $D_{2,3}$ and $D_{4,5}$), whereas solid lines represent active boundary variables ($D_{1,4}$, $D_{1,5}$, $D_{2,5}$, $D_{2,6}$, $D_{3,6}$ and $D_{5,6}$).

Co-clustering framework

8

This chapter gives an insight to the contour-based co-clustering framework proposed in [GVB11, VAM15], which was intended for sequences with small variations. This framework will be extended to the multiview scenario and, in conjunction of a state-of-the-art semantic segmentation technique, will provide a solution for the semantic multiview segmentation problem. First, in Section 8.1, the initial contour-based co-clustering framework proposed in [GVB11] is analyzed in detail. Then, in Section 8.2, we review the extension of the previous contour-based co-clustering which exploits hierarchical information from the partitions and proposes an iterative approach to segment coherently all the frames from a video sequence [VAM15]. Finally, Section 8.3 presents our contributions to the contour-based co-clustering framework, which include (i) a motion-aware hierarchical contour-based co-clustering, (ii) a more intuitive resolution parameterization, (iii) a two-step iterative framework, (iv) a feasible global optimization, (v) a semantic-based co-clustering framework, (vi) a semantic-based automatic resolution selection method, and (vii) a co-clustering based semantic segmentation framework.

8.1 Contour-based co-clustering

The authors in [GVB11] propose a contour-based co-clustering technique, which means a significative change in the approach of previous region-based co-clustering techniques such as [VB10]. Their motivation was that region-based measures alone, such as pixel area overlap and color histogram similarity used in [VB10], may not be ideal for shape comparison. Therefore, they present a method that combines contour- and region-based information to produce a joint clustering of two or more closely-related images. The rest of this section presents the details of this technique.

An important concept in the formulation of the contour-based co-clustering technique is the contour element. Given an image I and its associated partition P, P is represented as a collection of N regions $\{R_j\} = \{R_1, R_2, ..., R_N\}$ and q contour elements. Contour elements are defined as elements that connect two adjacent pixels belonging to two different regions from the same partition, where 4-connectivity is taken into account for adjacency. Given an image domain $\Omega = [1, s_x] \times [1, s_y] \subset \mathbb{R}^2$, contour elements are represented in a domain $\Omega' = [1, 2s_x - 1] \times [1, 2s_y - 1]$. Figure 8.1 shows an example where each pixel is considered as a region. In this example, the contour elements are the elements represented as — or |, which denote horizontal and vertical contour elements respectively. In addition to them, the gray elements represent the pixel positions and the empty elements do not represent either contour elements or pixels.

As working with partitions at pixel level would result in algorithms of high complexity in both time and memory, the co-clustering framework takes as input an oversegmen-

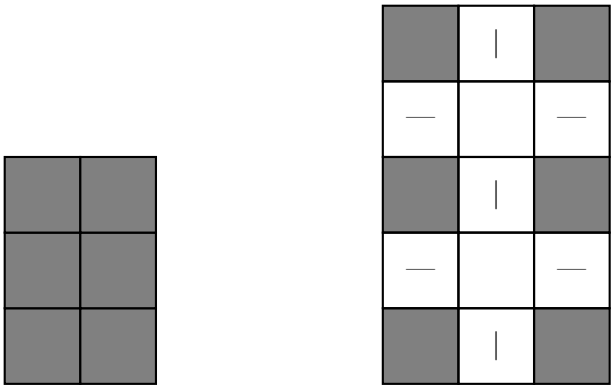


Figure 8.1: The domain Ω' where contour elements are defined is larger than the image domain Ω. Given a 3×2 pixel image as shown on the left, the contour elements are the elements noted as | (vertical) and — (horizontal) in the 5×3 domain on the right. Horizontal contour elements represent contours between vertically adjacent pixels and vertical contour elements represent contours between horizontally adjacent pixels. Gray elements do not represent the pixel values, but the pixel positions.

tation of the image. By oversegmentation we understand a partition of the image that fulfils that every region of such a partition covers part of only one semantic object in the image. Therefore, oversegmentation techniques aim to obtain segmentations so that two semantic objects are not covered by the same region, no matter how many regions are used to represent every object. These oversegmentations can be obtained by many state-of-the-art superpixels algorithms [ASS+12, VdBBR+12], where each region consist of a set of similar connected pixels according to some similarity criteria such as color or texture. In particular, [GVB11] uses the gPb-owt-ucm algorithm [AMFM11] to obtain the initial oversegmentations, which will be referred to as *leaves partitions*. Figure 8.2 represents a leaves partition of the previous image from Figure 8.1, where pixels have been segmented into three regions R_1, R_2 and R_3. As we work with such initial oversegmentations, contour elements previously defined in Figure 8.1 that represent a contour between two pixels that belong to the same region are no longer considered as contour elements, resulting in a reduction of the complexity. The contour-based co-clustering framework will be defined using the contour elements present in the leaves partition, i.e. the ones denoted as x_k in Figure 8.2.

For each contour element $k \in \{1, ..., q\}$ of the partition P, two opposite vectors with normal direction to the region contour at k are assigned, one for each associated region, which will allow the construction of an additive representation. This representation is additive in the sense that whenever a merging of two regions is considered, the representation of the union can be obtained as the addition of the representation of their regions. This is possible thanks to the opposite outward-pointing vectors which are cancelled for the contour elements belonging to the boundary shared by these regions. The computation of these vectors can be done using two types of techniques: (i) region-based techniques, and (ii) image-based techniques. Region-based techniques parameterize the region boundary and compute the gradient at each contour element belonging to the boundary. Then, the vector with normal direction is a vector orthogonal to the gradient. Therefore, orientation of the vector depends on the shape of the boundary. On the other hand, image-based techniques computes horizontal and vertical gradients on the image

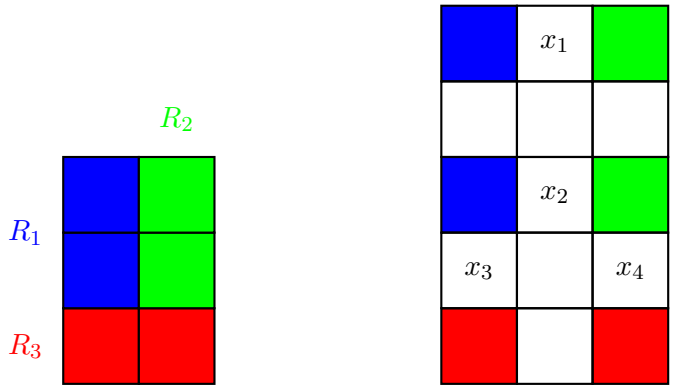


Figure 8.2: Contour-based co-clustering framework only considers the contour elements that represents contours between pixels belonging to different regions. These contour elements are denoted as x_k and each color represents a different region label.

by using some filters, such as the Sobel filter [SF68], and the vector is obtained by addition of such horizontal and vertical gradient vectors. In [VAM15], as leaves partitions are created with the gPb-owt-ucm algorithm [AMFM11] and gPb carries gradient intensity values at eight different orientations, the orientation with the maximum intensity value at each contour element is used to represent its normal vector. Therefore, the orientation of the vector is obtained using an image-based technique.

Following the example in Figure 8.2, the two opposite vectors assigned to one of the contour elements are shown in Figure 8.3. Since the outward-pointing normal vectors assigned to a contour element have opposite orientations, if the angle that forms one of them with reference to the X-axis is θ, then the orientation of the opposite vector is $\theta + \pi$. For each region R_j of the partition P, a $q \times 1$ column vector b_j encodes the contour elements belonging to R_j and their associated orientations, where q is the number of contour elements in P. If the kth contour element belongs to R_j, the kth component of b_j is set to $b_j(k) = e^{i\theta}$, where θ is the orientation of the normal vector. Otherwise, if the kth contour element does not belong to R_j, the kth component of b_j is set to 0 ($b_j(k) = 0$). Then, a $q \times N$ matrix B is obtained as a concatenation of the column vectors $B = (b_1\ b_2\ ...\ b_N)$, where N is the number of regions in P. Following the same example (Figure 8.3), there are $q = 4$ contour elements. Region R_1 has the contour elements $\{x_1, x_2, x_3\}$, R_2 has $\{x_1, x_2, x_4\}$ and R_3 has $\{x_3, x_4\}$. If we consider that each contour element x_k has an orientation θ_k, then the vector b_1 associated to R_1 is $b_1 = [e^{i\theta_1}\ e^{i\theta_2}\ e^{i\theta_3}\ 0]^T$. Analogously, $b_2 = [e^{i(\theta_1+\pi)}\ e^{i(\theta_2+\pi)}\ 0\ e^{i\theta_4}]^T$ and $b_3 = [0\ 0\ e^{i(\theta_3+\pi)}\ e^{i(\theta_4+\pi)}]^T$. This representation is additive: if two regions R_i and R_j are merged into a parent region R_p, the vector assigned to the parent region b_p can be obtained as the addition of the vectors assigned to the children regions b_i and b_j. From the previous example, if R_1 and R_2 are merged into a region R_4, the shared contour elements $\{x_1, x_2\}$ vanish and only the contour elements which lie along the exterior boundary of the union would remain, i.e. $\{x_3, x_4\}$. Therefore, $b_4 = [0\ 0\ e^{i\theta_3}\ e^{i\theta_4}]^T$ can be also obtained as $b_1 + b_2 = [e^{i\theta_1}\ e^{i\theta_2}\ e^{i\theta_3}\ 0]^T + [e^{i(\theta_1+\pi)}\ e^{i(\theta_2+\pi)}\ 0\ e^{i\theta_4}]^T$, taking into account that $e^{i(\theta+\pi)} = -e^{i\theta}$.

Once presented the region representation based on contour elements, let us approach the problem of co-clustering two or more closely-related images. Co-clustering aims at

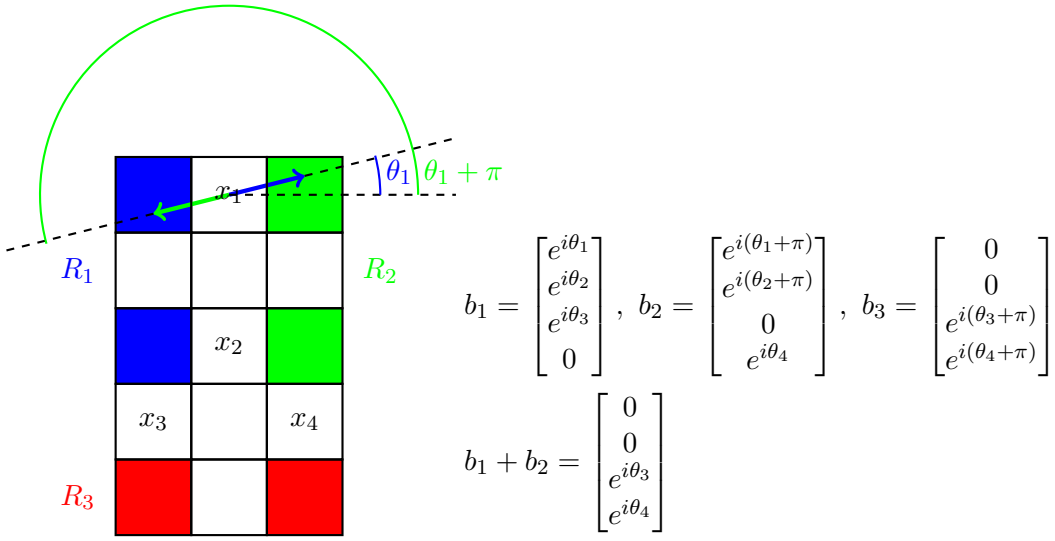


Figure 8.3: Opposite outward-pointing normal vectors assigned to contour element x_1. Vectors b_1, b_2 and b_3 associated to R_1, R_2 and R_3 respectively are also provided. Note that the components associated to the contour elements x_1 and x_2 are cancelled when vectors b_1 and b_2 are added, representing the vector associated to R_1 and R_2 merging.

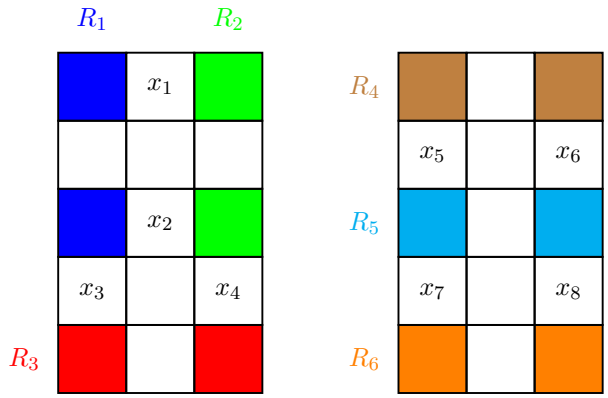


Figure 8.4: A set of two partitions with their corresponding contour elements.

grouping regions from a set of partitions creating clusters based on region similarities. Any co-clustering solution has an associated score that assess how well the clusters created fit with the region similarities. The objective is to find an unknown number of clusters that maximize such score. The region similarities previously introduced can be classified in two different types depending whether regions belong to different partitions or to the same partition: (*i*) the inter image interaction, which is based on the similarity between contour elements belonging to different partitions, and (*ii*) the intra image interaction, which is a region-based affinity measure between regions from the same partition. Next, the details about these two types of interaction are given following the example from Figure 8.4, where two partitions P_1 and P_2 from a pair of two closely-related images I_1 and I_2 are considered.

8.1.1 Inter image interaction

Regarding the inter image interaction, a $q_1 \times q_2$ matrix $W^{(1,2)}$ encodes the similarity between the contour elements from partitions P_1 and P_2, where q_1 and q_2 are the number of contour elements of P_1 and P_2 respectively. More specifically, $W^{(1,2)}_{i,j}$ encodes the similarity between the ith contour element from P_1 and the jth contour element from P_2. Analogously, $W^{(2,1)}$ is a $q_2 \times q_1$ matrix that encodes the similarity between the contour elements from P_2 with respect to those from P_1 and fulfills $W^{(2,1)} = (W^{(1,2)})^T$. Each similarity value represented by $W^{(1,2)}_{i,j}$ is computed as follows. First, feature vectors f_i and f_j are obtained as HOG-type descriptors computed over windows centered in the ith contour element from P_1 and the jth contour element from P_2 respectively. Then, $W^{(1,2)}_{i,j}$ is computed as $exp((f_i - f_j)^T \Sigma^{-1}(f_i - f_j))$, where Σ is a diagonal matrix with values in the diagonal proportional to the estimated variance of the feature vectors. Contour elements at a distance of more than 10 pixels are not considered ($W^{(1,2)}_{i,j}$ is set to 0).

Then, a $N_1 \times N_2$ matrix $Q^{(1,2)}$ is defined as $Q^{(1,2)} = B_1^H W^{(1,2)} B_2$, where X^H denotes the Hermitian transpose of X, N_1 and N_2 are the number of regions of P_1 and P_2 respectively, and B_1 and B_2 are the B matrices that encode the contour elements of P_1 and P_2 respectively. This matrix $Q^{(1,2)}$ encodes the similarity between the regions from partitions P_1 and P_2. More specifically, each component $Q^{(1,2)}_{i,j}$ represents the similarity between the region R_i from P_1 and the region R_j from P_2. Following the example from Figure 8.4, $Q^{(1,2)}$ is:

$$Q^{(1,2)} = \begin{bmatrix} e^{-i\theta_1} & e^{-i\theta_2} & e^{-i\theta_3} & 0 \\ e^{-i(\theta_1+\pi)} & e^{-i(\theta_2+\pi)} & 0 & e^{-i\theta_4} \\ 0 & 0 & e^{-i(\theta_3+\pi)} & e^{-i(\theta_4+\pi)} \end{bmatrix} W^{(1,2)} \begin{bmatrix} e^{i\theta_5} & e^{i(\theta_5+\pi)} & 0 \\ e^{i\theta_6} & e^{i(\theta_6+\pi)} & 0 \\ 0 & e^{i\theta_7} & e^{i(\theta_7+\pi)} \\ 0 & e^{i\theta_8} & e^{i(\theta_8+\pi)} \end{bmatrix}$$

where developing the previous expression we would obtain:

$$\begin{aligned}
Q^{(1,2)}_{1,1} &= (e^{-i\theta_1} W^{(1,2)}_{1,5} + e^{-i\theta_2} W^{(1,2)}_{2,5} + e^{-i\theta_3} W^{(1,2)}_{3,5}) e^{i\theta_5} \\
&\quad + (e^{-i\theta_1} W^{(1,2)}_{1,6} + e^{-i\theta_2} W^{(1,2)}_{2,6} + e^{-i\theta_3} W^{(1,2)}_{3,6}) e^{i\theta_6} \\
Q^{(1,2)}_{1,2} &= (e^{-i\theta_1} W^{(1,2)}_{1,5} + e^{-i\theta_2} W^{(1,2)}_{2,5} + e^{-i\theta_3} W^{(1,2)}_{3,5}) e^{i(\theta_5+\pi)} \\
&\quad + (e^{-i\theta_1} W^{(1,2)}_{1,6} + e^{-i\theta_2} W^{(1,2)}_{2,6} + e^{-i\theta_3} W^{(1,2)}_{3,6}) e^{i(\theta_6+\pi)} \\
&\quad + (e^{-i\theta_1} W^{(1,2)}_{1,7} + e^{-i\theta_2} W^{(1,2)}_{2,7} + e^{-i\theta_3} W^{(1,2)}_{3,7}) e^{i\theta_7} \\
&\quad + (e^{-i\theta_1} W^{(1,2)}_{1,8} + e^{-i\theta_2} W^{(1,2)}_{2,8} + e^{-i\theta_3} W^{(1,2)}_{3,8}) e^{i\theta_8} \\
&\vdots \\
Q^{(1,2)}_{3,3} &= (e^{-i(\theta_3+\pi)} W^{(1,2)}_{3,7} + e^{-i(\theta_4+\pi)} W^{(1,2)}_{4,7}) e^{i(\theta_7+\pi)} \\
&\quad + (e^{-i(\theta_3+\pi)} W^{(1,2)}_{3,8} + e^{-i(\theta_4+\pi)} W^{(1,2)}_{4,8}) e^{i(\theta_8+\pi)}
\end{aligned} \tag{8.1}$$

From the previous development, we can observe that each component $Q^{(1,2)}_{i,j}$ is a weighted

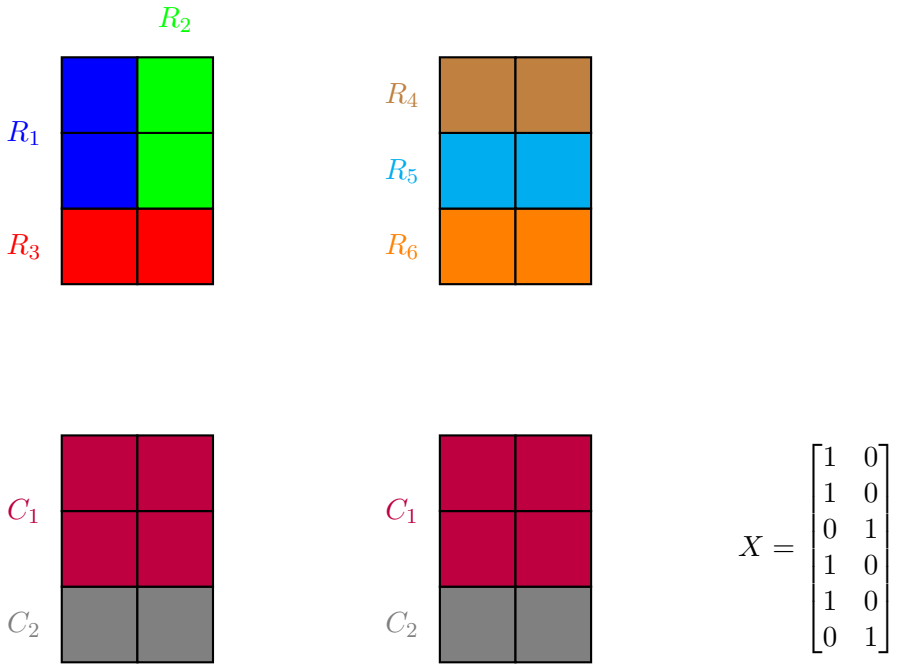


Figure 8.5: A possible co-clustering solution where regions $\{R_1, R_2, R_4, R_5\}$ have been assigned to a cluster C_1 and regions $\{R_3, R_6\}$ to a cluster C_2. Matrix X encodes the clusters to which regions are assigned. First column x_1 represents cluster C_1 and second column x_2 represents cluster C_2.

addition of the similarity values $W_{k,l}^{(1,2)}$ between the contour elements from R_i and R_j as follows:

$$Q_{i,j}^{(1,2)} = \sum_{k,l} e^{-i\theta_k} W_{k,l}^{(1,2)} e^{i\theta_l} \tag{8.2}$$

where k represents all contour elements belonging to R_i from P_1, l represents all contour elements belonging to R_j from P_2, and θ_k and θ_l are their respective orientations.

Then, a complex-value Hermitian matrix Q is built as follows:

$$Q = \begin{bmatrix} Q^{(1,1)} & Q^{(1,2)} \\ Q^{(2,1)} & Q^{(2,2)} \end{bmatrix}$$

where $Q^{(i,i)}$ are intra image submatrices and $Q^{(i,j)}$ are inter image submatrices, being $Q^{(j,i)} = (Q^{(i,j)})^H$. How intra image submatrices $Q^{(i,i)}$ are obtained is explained later in this section (see Equation 8.7).

Let us suppose that we want to compute the score associated with the possible co-clustering solution illustrated in Figure 8.5 that consists of two clusters: (i) $C_1 = \{R_1, R_2, R_4, R_5\}$, and (ii) $C_2 = \{R_3, R_6\}$. Any co-clustering solution can be represented as a $n \times c$ matrix X, where n is the total number of regions, i.e. $N_1 + N_2$, and c is the number of clusters in the co-clustering. Each column of matrix X repre-

sents a single cluster, where $X_{i,j} = 1$ if ith region participates in cluster j and $X_{i,j} = 0$ otherwise. For the particular solution illustrated in Figure 8.5, the matrix X would be $X = [1\ 1\ 0\ 1\ 1\ 0; 0\ 0\ 1\ 0\ 0\ 1]^T$. The unique constraint required to matrix X to be a possible solution of the co-clustering is that each region is only assigned to exactly one cluster. This is achieved by requiring X to have unit norm rows. The score associated with a clustering matrix X is defined as

$$tr(X^T Q X) = \sum_{k=1}^{c} x_k^T Q x_k = \sum_{k=1}^{c} \sum_{i=1}^{n} \sum_{j=1}^{n} x_k(i) Q_{i,j} x_k(j) \tag{8.3}$$

where x_k are the columns of the clustering matrix X, $x_k(i)$ is the ith component of x_k, and $Q_{i,j}$ is the element from Q matrix at row i and column j. Notice that if $i \leq N_1$ and $j \leq N_1$, $Q_{i,j}$ belongs to the intra image submatrix $Q^{(1,1)}$, whereas if $i \leq N_1$ and $j > N_1$, $Q_{i,j}$ belongs to the inter image submatrix $Q^{(1,2)}$. Analogously, if $i > N_1$ and $j > N_1$, $Q_{i,j}$ belongs to the intra image submatrix $Q^{(2,2)}$, whereas if $i > N_1$ and $j \leq N_1$, $Q_{i,j}$ belongs to the inter image submatrix $Q^{(2,1)}$. Moreover, notice that for values r and s belonging to different partitions, i.e. $r \leq N_1$ and $s > N_1$ or $r > N_1$ and $s \leq N_1$, if we add the contribution in Equation 8.3 of the terms corresponding to indices $i = r, j = s$ and $i = s, j = r$

$$x_k(i) Q_{i,j}^{(1,2)} x_k(j) + x_k(j) Q_{j,i}^{(2,1)} x_k(i)$$

and then we replace $Q_{i,j}^{(1,2)}$ by the expression from Equation 8.2 and knowing that $Q^{(2,1)} = (Q^{(1,2)})^H$, the previous expression becomes:

$$\begin{aligned}
&x_k(i)x_k(j) \sum_{m,l} e^{-i\theta_m} W_{m,l}^{(1,2)} e^{i\theta_l} + x_k(j)x_k(i) \sum_{m,l} e^{i\theta_m} W_{m,l}^{(1,2)} e^{-i\theta_l} = \\
=&x_k(i)x_k(j) \left(\sum_{m,l} e^{-i\theta_m} W_{m,l}^{(1,2)} e^{i\theta_l} + \sum_{m,l} e^{i\theta_m} W_{m,l}^{(1,2)} e^{-i\theta_l} \right) = \\
=&x_k(i)x_k(j) \sum_{m,l} W_{m,l}^{(1,2)} \left(e^{-i\theta_m} e^{i\theta_l} + e^{i\theta_m} e^{-i\theta_l} \right) = \\
=&x_k(i)x_k(j) \sum_{m,l} W_{m,l}^{(1,2)} \left(e^{i(\theta_l - \theta_m)} + e^{-i(\theta_l - \theta_m)} \right) = \\
=&x_k(i)x_k(j) \sum_{m,l} W_{m,l}^{(1,2)} 2cos(\theta_l - \theta_m)
\end{aligned} \tag{8.4}$$

where m represents all contour elements belonging to R_i from P_1, l represents all contour elements belonging to R_j from P_2, and θ_m and θ_l are their respective orientations. The previous expression only contributes to the co-clustering score if region R_i from P_1 and region R_j from P_2 belong to the same cluster, i.e. $x_k(i) = x_k(j) = 1$. As can be observed in Equation 8.4, the similarity given by $W_{m,l}^{(1,2)}$ is weighted with a term proportional to the cosine of the angle that form the associated normal vectors. Therefore, contour elements with similar feature vectors but different oriented normal vectors are penalyzed.

Next, we develop the example given in Figure 8.5 when only inter-image interaction is taken into account. If we use Equation 8.3 to compute the score associated with this co-clustering solution

$$
\begin{aligned}
tr(X^T Q X) = tr\left(\begin{bmatrix} 1 & 1 & 0 & 1 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} Q \begin{bmatrix} 1 & 0 \\ 1 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 0 \\ 0 & 1 \end{bmatrix} \right) \\
= Q_{1,1} + Q_{2,1} + Q_{4,1} + Q_{5,1} + Q_{1,2} + Q_{2,2} + Q_{4,2} + Q_{5,2} + Q_{1,4} + Q_{2,4} \\
+ Q_{4,4} + Q_{5,4} + Q_{1,5} + Q_{2,5} + Q_{4,5} + Q_{5,5} + Q_{3,3} + Q_{6,3} + Q_{3,6} + Q_{6,6} \quad (8.5)
\end{aligned}
$$

and we discard all the terms coming from intra image submatrices $Q^{(1,1)}$ and $Q^{(2,2)}$ as we want to focus on inter-image interaction, then we have:

$$
tr(X^T Q X) = Q_{4,1} + Q_{5,1} + Q_{4,2} + Q_{5,2} + Q_{1,4} + Q_{2,4} + Q_{1,5} + Q_{2,5} + Q_{6,3} + Q_{3,6}
$$

Then, replacing these values by the ones given in Equation 8.1 and thanks to the additive property, the previous expression simplifies to:

$$
\begin{aligned}
tr(X^T Q X) = 4W_{3,7}^{(1,2)} cos(\theta_7 - \theta_3) + 4W_{3,8}^{(1,2)} cos(\theta_8 - \theta_3) + 4W_{4,7}^{(1,2)} cos(\theta_7 - \theta_4) \\
+ 4W_{4,8}^{(1,2)} cos(\theta_8 - \theta_4) \quad (8.6)
\end{aligned}
$$

Note that, in the previous expression, all terms coming from contour elements belonging to vanishing boundaries (x_1 and x_2 due to R_1 and R_2 merging, and x_5 and x_6 due to R_4 and R_5 merging) have been cancelled and only terms from the remaining contour elements contribute to the co-clustering score. Moreover, each pair of contour elements i and j contributes with a product of two terms: (i) the similarity between the features vectors associated to i and j and encoded by $W_{i,j}^{(1,2)}$, and (ii) the similarity of the orientations of the normal vectors θ_i and θ_j associated to i and j.

8.1.2 Intra image interaction

Regarding the intra image interaction, a $N_i \times N_i$ matrix $Q^{(i,i)}$ is built for each partition P_i, where N_i is the number of regions in P_i and $Q^{(i,i)}$ is a real symmetric matrix. This matrix encodes the affinity between different regions of P_i and plays an analogous role to affinity matrices in standard segmentation algorithms. For each pair of adjacent regions R_k and R_l, $Q_{k,l}^{(i,i)}$ is computed as follows:

$$
Q_{k,l}^{(i,i)} = \lambda v_{k,l} u_{k,l} \quad (8.7)
$$

where $\lambda \geq 0$ is a parameter that controls the resolution of the co-clustering result, $v_{k,l}$ denotes the length of the common boundary of R_k and R_l, and $u_{k,l}$ encodes a motion and color-based similarity between R_k and R_l. The resolution of the co-clustering result is defined as the number of clusters to which regions are assigned. Higher values of λ lead to a smaller number of clusters by encouraging mergings between similar regions with long common boundaries. Furthermore, $Q_{k,l}^{(i,i)} = 0$ for pairs of non-adjacent regions. Motion and color-based similarity $u_{k,l}$ is defined in [GVB11] as follows:

$$u_{k,l} = \frac{1}{2}\left[exp\left(\frac{-||d_k - d_l||^2}{\sigma_c^2}\right) + exp\left(\frac{-||f_k - f_l||^2}{\sigma_f^2}\right)\right]$$

where d_k is the normalized $L*a*b$ color histogram of region R_k, f_k is the median optical flow inside region R_k, and σ_c^2 and σ_f^2 are the color and motion feature variances.

8.1.3 Inter and intra image interactions

Following the example illustrated in Figure 8.5 and considering now both inter and intra image correspondences, the terms becoming from intra-image sub-matrices $Q^{(1,1)}$ and $Q^{(2,2)}$ are not discarded any more and Equation 8.5 results in:

$$\begin{aligned} tr(X^T QX) = {} & Q_{2,1} + Q_{1,2} + Q_{5,4} + Q_{4,5} \\ & + Q_{4,1} + Q_{5,1} + Q_{4,2} + Q_{5,2} + Q_{1,4} + Q_{2,4} + Q_{1,5} + Q_{2,5} + Q_{6,3} + Q_{3,6} \end{aligned}$$

where $Q_{i,i}$ terms from Equation 8.5 have been removed because the similarity of a region R_i with respect to itself is not considered. Considering that $Q^{(i,i)}$ matrices are symmetric and the result from Equation 8.6 when only inter-image correspondence is considered, the previous expression can be formulated as:

$$\begin{aligned} tr(X^T QX) = {} & 2Q_{1,2} + 2Q_{4,5} + 4W_{3,7}^{(1,2)} cos(\theta_7 - \theta_3) + 4W_{3,8}^{(1,2)} cos(\theta_8 - \theta_3) \\ & + 4W_{4,7}^{(1,2)} cos(\theta_7 - \theta_4) + 4W_{4,8}^{(1,2)} cos(\theta_8 - \theta_4) \end{aligned} \tag{8.8}$$

Note that, regarding the intra image interaction, only affinities between regions that participate in the same cluster are considered. Thus, $Q_{1,3}$, $Q_{2,3}$ and $Q_{5,6}$ are not taken into account as R_1 and R_3, R_2 and R_3, and R_5 and R_6 have not been respectively assigned to the same cluster.

Previous Equation 8.8 only gives the score of the possible co-clustering illustrated in Figure 8.5. However, the goal of the co-clustering is obtaining the solution that gives the maximum score. Therefore, the optimization objective is

$$\begin{aligned} \max_X \quad & tr(X^T QX) \\ \text{s.t.} \quad & X_{i,j} \in \{0,1\} \ \forall i,j \text{ and } \sum_j X_{i,j} = 1 \ \forall i. \end{aligned}$$

This is a Quadratic Semi-Assignment Problem (QSAP) [VB10], which can be written as

$$\begin{aligned} \max_{Y} \quad & tr(QY) \\ \text{s.t.} \quad & Y \succeq 0,\ Y_{i,j} \in \{0,1\}\ \forall i,j \text{ and } Y_{i,i} = 1\ \forall i. \end{aligned}$$

where $Y = XX^T$ is of (unknown) rank c. The requirement that every region participates in exactly one cluster is expressed in the constraint $Y_{i,i} = 1$.

In [CGW], it is shown that this solution can be tackled with a Linear Programming relaxation approach. However, as metric properties are imposed using linear constraints that enforce triangular inequality, this relaxation has a crucial limitation -the number of triangular inequalities grows as $O(n^3)$ where n is the number of regions. In [VB10], the authors present a further relaxation by enforcing only triangular inequalities for three-cliques of adjacent regions. This further relaxation bounds the number of constraints to $O(n^2)$ and in practice is almost linear in n. As a result, the optimization problem becomes

$$\begin{aligned} \min_{D} \quad & \sum_{i,j} Q_{i,j} D_{i,j} \\ \text{s.t.} \quad & D_{i,j} \in \{0,1\} \\ & D_{i,i} = 0\ \forall i,\ D_{i,j} = D_{j,i}\ \forall i,j \\ & D_{i,j} \leq D_{i,k} + D_{k,j}\ \forall e_{i,j}, e_{i,k}, e_{k,j} \in G, \end{aligned} \tag{8.9}$$

where $D_{i,j} = 0$ implies that regions i and j should belong to the same cluster and $D_{i,j} = 1$ otherwise, and G is the region adjacency graph from which three-cliques of adjacent regions are considered to impose the triangular inequalities. Whereas the concept of adjacency was introduced in Section 7.4 for a single partition, we extend here the concept for a set of partitions. Given a set of partitions, two kinds of adjacency are considered: the intra adjacency, which refers to regions from the same partition, and the inter adjacency, which refers to regions from different partitions. Intra adjacency is defined as in Section 7.4, i.e. two regions R_i and R_j from the same partition are adjacent if any pixel p_i from R_i has at least one pixel p_j from R_j among the 4-connected pixels of p_i. On the other hand, inter adjacency is defined as follows. Two regions R_m and R_n from partitions P_i and P_j respectively are considered adjacent if at least one pixel from R_m overlaps with a pixel of R_n, i.e. the intersection of their sets of pixel coordinates is not empty. For instance, regions R_1 and R_4 from Figure 8.5 are adjacent, whereas regions R_3 and R_4 are not.

8.2 Multiresolution Hierarchy Co-clustering

This section gives an insight to the multiresolution hierarchy co-clustering [VAM15], which is based on the contour-based co-clustering [GVB11] presented in Section 8.1. Different problems detected from the solution given in [GVB11] are tackled by [VAM15].

First, none of the constraints imposed in [GVB11] guarantees that the solution obtained for each image is a partition. To solve that, [VAM15] imposes as input independent

hierarchies obtained for each frame. Section 8.2.1 presents how these hierarchies are used as a constraints of the optimization problem.

Second, [VAM15] focuses on segmenting video sequences with small variations. In this new scenario, motion cues are not further trustful to compute the intra-image interactions. Therefore, they propose to remove motion cues from them. Section 8.2.2 gives the details about how intra and inter similarities are obtained.

Third, in [GVB11], the resolution of the co-clustered partitions is set using the similarity multiplier λ, which leads to non-homogeneous multiresolution representations, i.e. consecutive resolutions may be almost equal or present very large variations. [VAM15] proposes an alternative parameter based on the number of active boundaries. However, this alternative parameterization presented in Section 8.2.3 does not lead to homogeneous multiresolution representations either.

Last, [VAM15] proposes an iterative algorithm that makes the framework less complex in terms of time and memory with respect to the global optimization presented by [GVB11]. Here, *global* refers to an optimization process that is jointly applied to all the frames, whereas *iterative* refers to a forward-online optimization process where the segmentation of each frame depends on the segmentation obtained for previous frames, which are not further modified. Section 8.2.4 gives the details about the iterative approach.

8.2.1 Co-clustering of hierarchies

The approach presented in Section 8.1 gives solutions to the optimization problem that present inconsistencies because the proposed constraints do not force the solution to be a partition. In [VAM15], further constraints are considered by imposing the structure of the hierarchies, which have been independently obtained for each frame. More specifically, the gPb-owt-ucm segmentation technique is used to obtained such hierarchies. Since these hierarchies convey information about how likely are the regions to be merged and, therefore, about the merging order, it is expected to obtain partitions closer to the semantic level. Therefore, in contrast to previous approaches, the associated hierarchies $\{H_i\}_{i=1}^{M} = \{H_1, H_2, ..., H_M\}$ to the collection of the M closely-related images $\{I_i\}_{i=1}^{M} = \{I_1, I_2, ..., I_M\}$ and partitions $\{P_i\}_{i=1}^{M} = \{P_1, P_2, ..., P_M\}$ are also considered. Each hierarchy H_i is imposed through two constraints that are applied to each parent node of H_i.

Before introducing such constraints, let us define intra-sibling boundary and inter-sibling boundary. Given a parent node, which has two sibling nodes, intra-sibling boundaries are defines as boundaries connecting adjacent regions from the same sibling and inter-sibling boundaries as those connecting adjacent regions from different siblings. Given the hierarchy illustrated in Figure 8.6, let us use it as example to identify the intra-sibling and inter-sibling boundaries. As these boundaries are defined for each parent node, nodes 7, 8, 9, 10 and 11 have different sets of boundaries associated with them. Let us focus on one of them, for instance the node 11, which has the nodes 8 and 10 as sibling nodes. The former consists of the regions 1, 2 and 3, whereas the latter consists of the regions 4, 5 and 6. Intra-sibling boundaries are those connecting adjacent regions either from {1, 2, 3} or {4, 5, 6}. From the first sibling, we have the intra-sibling boundaries $D_{1,2}$ and $D_{2,3}$. Analogously, intra-sibling boundaries $D_{4,5}$ and $D_{5,6}$ from the second sibling. $D_{1,3}$ and $D_{4,6}$ are not considered because neither regions 1 and 3 nor regions 4 and 6

are adjacent. As a result, $\{D_{1,2},\ D_{2,3},\ D_{4,5},\ D_{5,6}\}$ are the intra-sibling boundaries for the parent node 11. On the other hand, inter-sibling boundaries are those connecting adjacent regions from one sibling to the other one. Therefore, $\{D_{1,4},\ D_{1,5},\ D_{2,5},\ D_{2,6},\ D_{3,6}\}$ are the inter-sibling boundaries for the parent node 11.

The first constraint forces that, given two siblings, all their common boundaries (inter-sibling boundaries) are either jointly active or inactive. If we arbitrarily select one of the inter-sibling boundaries and we denote it as $D_{m,n}$, the first constraint is:

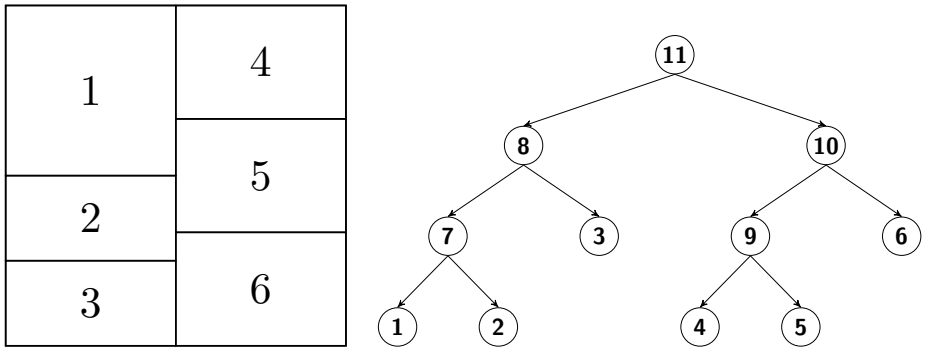
$$\sum_{k,l} D_{k,l} = N_{inter} D_{m,n} \tag{8.10}$$

where $D_{k,l}$ represents an inter-sibling boundary (including $D_{m,n}$) and N_{inter} is the number of inter-sibling boundaries. Following the previous example (Figure 8.6), for the parent node 11, if we arbitrarily select $D_{1,4}$ among its inter-sibling boundaries ($\{D_{1,4},\ D_{1,5},\ D_{2,5},\ D_{2,6},\ D_{3,6}\}$), the previous constraint becomes $D_{1,4} + D_{1,5} + D_{2,5} + D_{2,6} + D_{3,6} = 5D_{1,4}$. Such a constraint forces that all inter-sibling boundaries are either jointly active or inactive. The solutions obtained if any other inter-sibling boundary was selected instead of $D_{1,4}$ would be the same. If some of the previous boundary variables had different values, i.e. some of them active and the other ones inactive, there would be contradictions about the merging of nodes 8 and 10. The inactive boundary variables would indicate that they are merged, whereas the active ones would not. Note that the constraint has to be obtained for each parent node and, therefore, a different constraint is applied to each parent node. For instance, for the parent node 8 from Figure 8.6, as $D_{2,3}$ is the unique inter-sibling boundary, the constraint for such a node becomes: $D_{2,3} = D_{2,3}$, which, in this case, does not imply any restriction. Analogously, $D_{5,6} = D_{5,6}$ for parent node 10, $D_{1,2} = D_{1,2}$ for parent node 7 and $D_{4,5} = D_{4,5}$ for parent node 9.

In turn, the second constraint imposes that two siblings can only be merged as long as the regions that form their respective subtrees (encoded with the intra-sibling boundaries) have also been merged. Given the previously arbitrarily selected inter-sibling boundary denoted as $D_{m,n}$, the second constraint is:

$$\sum_{k,l} D_{k,l} \leq N_{intra} D_{m,n} \tag{8.11}$$

where $D_{k,l}$ represents an intra-sibling boundary and N_{intra} is the number of intra-sibling boundaries. Following the example from Figure 8.6, for the parent node 11, the previous constraint becomes $D_{1,2} + D_{2,3} + D_{4,5} + D_{5,6} \leq 4D_{1,4}$, where the left-side variables in the inequation are the intra-sibling boundaries, the right-side variable $D_{1,4}$ is the selected inter-sibling boundary and 4 is the number of intra-sibling boundaries. This constraint can be interpreted as follows. If the boundary $D_{1,4}$ is inactive, i.e. nodes 8 and 10 are merged, all intra-sibling boundaries must be also inactive. Thus, if nodes 8 and 10 are merged, all nodes from their subtrees should also be merged, otherwise the hierarchy would be violated. On the contrary, if $D_{1,4}$ is active, there are no constraints imposed over the inner boundary variables, i.e. they can take any value. As the first constraint (Equation 8.10), the second constraint has to be also obtained for each parent node and, therefore, a different constraint is applied to each of them. Regarding the parent



For parent node 8: $D_{1,2} \leq D_{2,3}$ (Equation 8.11)
For parent node 10: $D_{4,5} \leq D_{5,6}$ (Equation 8.11)
For parent node 11: $D_{1,5} + D_{2,5} + D_{2,6} + D_{3,6} = 4D_{1,4}$ (Equation 8.10)
$D_{1,2} + D_{2,3} + D_{4,5} + D_{5,6} \leq 4D_{1,4}$ (Equation 8.11)

Figure 8.6: Illustrative example for intra constraints imposed by Equations 8.10 and 8.11.

node 8, since $D_{1,2}$ is its unique intra-sibling boundary, the second constraint becomes $D_{1,2} \leq D_{2,3}$. Analogously, the constraint becomes $D_{4,5} \leq D_{5,6}$ for parent node 9. The second constraint has no effect when applied to parent nodes 7 and 9 because they do not have any intra-sibling boundary.

These coupled hierarchical constraints are added to the optimization problem stated in Equation 8.9, resulting in the following formulation:

$$\begin{aligned} \min_{D} \quad & \sum_{i,j} Q_{i,j} D_{i,j} \\ \text{s.t.} \quad & 0 \leq D_{i,j} \leq 1 \\ & D_{i,i} = 0 \ \forall i, \ D_{i,j} = D_{j,i} \ \forall i,j \\ & D_{i,j} \leq D_{i,k} + D_{k,j} \ \forall e_{i,j}, e_{i,k}, e_{k,j} \in G \\ & \sum_{k,l} D_{k,l} = N_{inter} D_{m,n}, \ \sum_{r,s} D_{r,s} \leq N_{intra} D_{m,n} \ \forall \mathfrak{p} \in \{H_i\}_{i=1}^{M}, \end{aligned} \tag{8.12}$$

where $\mathfrak{p}$ represents a parent node in the collection of hierarchies, $D_{k,l}$ is an inter-sibling boundary of $\mathfrak{p}$ and $D_{m,n}$ is a single arbitrarily selected boundary among them, N_{inter} is the number of inter-sibling boundaries of $\mathfrak{p}$, $D_{r,s}$ is an intra-sibling boundary of $\mathfrak{p}$ and N_{intra} is the number of intra-sibling boundaries of $\mathfrak{p}$.

8.2.2 Intra and inter similarities

As in [GVB11], two types of similarities are computed: intra similarities (between regions from the same partition) and inter similarities (between regions from different partitions).

However, as [VAM15] aims to cluster regions from video sequences with small variations, both types of similarities have been adapted to such scenario.

Regarding the intra similarities, in [VAM15] it is proposed to change Equation 8.7 by the following equation:

$$Q_{k,l}^{(i,i)} = \alpha_{k,l}(1 - e^{1-d_B(k,l)})$$

where $\alpha_{k,l}$ is the length of the common boundary between regions R_k and R_l and $d_B(k,l)$ is the Bhattacharyya distance [Bha46] between the 8-bin separated channel RGB color histograms of regions R_k and R_l. As in [GVB11], the intra similarity also depends on the length of the common boundary and a color region-based affinity, but motion is no longer considered. This is because motion information does not help to infer the semantic in video sequences with global motion or little variation in the scene.

Regarding the inter similarities, contour element based feature vectors are modeled as a concatenation of three types of cues: color, texture and position. For color and texture, color histogram and HOG descriptors are computed in a window centered on the contour element respectively. Regarding position, contour element coordinates are considered. Whereas in [GVB11] HOG descriptors are also used to compare the contour elements, position is only used to truncate some similarities to 0. In contrast, in [VAM15], positions and color histograms are also used to compare the contour elements. However, as in [GVB11], contour elements at a distance of more than 10 pixels are not considered to reduce the complexity of the problem.

8.2.3 Multiresolution co-clustering

The previous hierarchical co-clustering (see Section 8.2.1) is extended to a multiresolution framework so that different co-clustering solutions are given for multiple resolutions, all of them respecting the hierarchical constraints imposed by each hierarchy. There is a direct relation between the number of active boundaries and the resolution of the resulting partition. When imposing a low (high) number of intra contours, coarser (finer) resolutions are obtained. Formally, to obtain a co-clustering solution of resolution r, the following constraint is added to the optimization problem presented in Equation 8.12:

$$(T_r - \beta)N_b \leq \sum_{m,n} D_{m,n} \leq T_r N_b \tag{8.13}$$

where N_b is the number of active boundaries to encode the leave contours, T_r is the maximum fraction of these contours to describe the r-th coarse level and β represents the maximum difference in number of boundaries between consecutive levels.

8.2.4 Iterative approach

Although the hiearchy co-clustering framework presented in Section 8.2.1 could be processed globally as in [GKHE10], such approach would require high memory resources. Thus, [VAM15] proposes an iterative approach as in [GKBS14] following the scheme illustrated in Figure 8.7. More specifically, the proposed approach is a forward-online

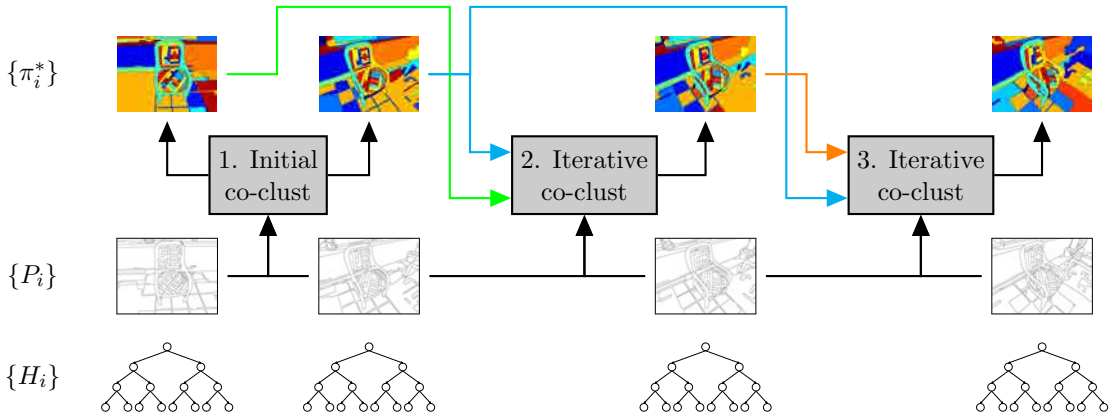


Figure 8.7: Co-clustering flowchart for iterative approach presented in [VAM15]. *Initial co-clustering.* It generates two coherent partitions (π_1^*, π_2^*) from a pair of partitions (P_1, P_2) and their associated hierarchies (H_1, H_2), used as constraints in the process. These are the first analyzed frames. *Iterative co-clustering.* The remaining set of partitions $\{\pi_i^*, i > 2\}$) are obtained using the iterative approach. To obtain π_i^*, partitions $\{P_{i-1}, P_i\}$ and their hierarchies $\{H_{i-1}, H_i\}$ are considered, as well as the two previous resulting partitions $\{\pi_{i-2}^*, \pi_{i-1}^*\}$. This information is used to impose coherence on the set of resulting partitions using Equations 8.14 and 8.15. The indices of the blocks denotes the order in which they are processed.

processing, where frames already processed do not suffer any segmentation change when the following frames are processed.

In particular, for each image I_i, a joint hierarchical co-clustering with the clustering result of the two previous frames at two different resolutions (the resolution level under analysis and the initial partition resolution) is performed by imposing two additional constraints to the optimization problem. Let us denote the partitions resulting from the co-clustering as $\{\pi_i^*\}$. To obtain π_i^*, partitions π_{i-2}^* and π_{i-1}^* are included in the optimization to keep coherence with the previous co-clustering results. Figure 8.8 shows an example to illustrate how the iterative approach is applied. On the one hand, there are boundaries that are forced to be active: (i) intra-image boundaries connecting adjacent clusters from π_{i-2}^* ($D_{A,B}$, $D_{A,C}$ and $D_{B,C}$ in Figure 8.8) , (ii) intra-image boundaries connecting adjacent regions from P_{i-1} that belong to different clusters in π_{i-1}^* ($D_{1,4}$, $D_{2,3}$, $D_{2,4}$ and $D_{3,4}$ in Figure 8.8), and (iii) inter-image boundaries connecting clusters from π_{i-2}^* with adjacent regions from P_{i-1}, where the region is assigned to a different cluster in π_{i-1}^* ($D_{A,3}$, $D_{A,4}$ and $D_{C,3}$ in Figure 8.8). Therefore, the first constraint is:

$$\sum_{m,n} D_{m,n} = N_v \tag{8.14}$$

where $D_{m,n}$ are the three types of intra-image and inter-image boundaries that must be active and N_v is the number of these boundaries. In the previous example, this constraint becomes $D_{A,B} + D_{A,C} + D_{B,C} + D_{1,4} + D_{2,3} + D_{2,4} + D_{3,4} + D_{A,3} + D_{A,4} + D_{C,3} = 10$. On the other hand, some boundaries must be inactive: (i) intra-image boundaries connecting adjacent regions from P_{i-1} that belong to the same cluster in π_{i-1}^* ($D_{1,2}$ in Figure 8.8),

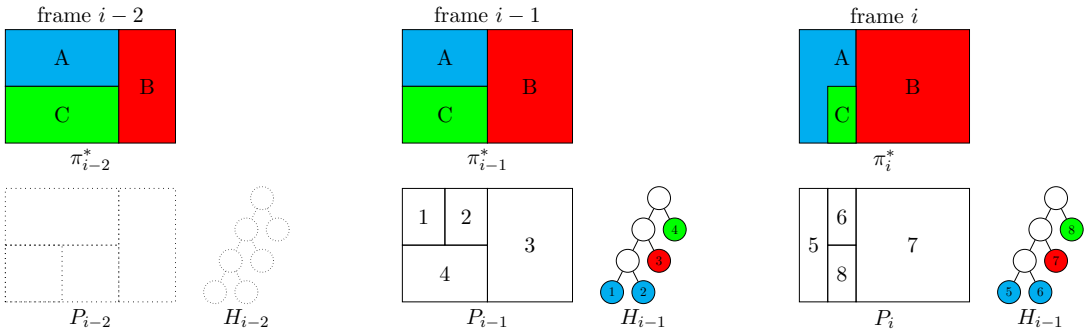


Figure 8.8: Illustrative example for constraints imposed by Equations 8.14 and 8.15 in the iterative approach. Red and blue framed variables denote intra and inter boundaries respectively. Regions are indexed by numbers while clusters are indexed by letters.

and (*ii*) inter-image boundaries connecting clusters from π^*_{i-2} with adjacent regions from P_{i-1}, where the region is assigned to the same cluster in π^*_{i-1} ($D_{A,1}$, $D_{A,2}$, $D_{B,3}$ and $D_{C,4}$ in Figure 8.8). Therefore, the second constraint is:

$$\sum_{m,n} D_{m,n} = 0 \tag{8.15}$$

where $D_{m,n}$ are the two types of intra-image and inter-image boundaries that must be inactive. Following the same example, this constraint becomes $D_{1,2} + D_{A,1} + D_{A,2} + D_{B,3} + D_{C,4} = 0$. Note that π^*_{i-1} is used to relate regions from P_{i-1} with clusters from π^*_{i-2} in both constraints. Also note that the given solutions in the previous example fulfill the constraints imposed by the hierarchies and each cluster can be represented by a single node from the hierarchy.

8.3 Multiresolution co-clustering for uncalibrated multiview segmentation

The framework overviewed in Section 8.2 is used in [VAM15] to propose an algorithm that iteratively clusters image regions in sequences with small variations. Despite its good results in this context, this optimization framework, as previous ones [GVB11, VB10], suffers from a major drawback in scenarios where variations are not negligible such as multiview sequences. In such scenarios, computing region adjacency graphs between different partitions as well as the similarity between contour elements encoded by W matrices without considering motion cues have a significative impact on the performance. Section 8.3.1 presents an adaptation of the co-clustering framework introducing motion cues.

Previous approaches (Sections 8.1 and 8.2) include parameters to set the resolution of the resulting co-clustered partitions, these parameters are not intuitive and lead to non-homogeneous multiresolution representations. Given a desired resolution r, it is hard to decide the value of the similarity multiplier λ (see Equation 8.7) from [GVB11] or the number of active boundaries N_b (see Equation 8.13) from [VAM15]. Section 8.3.2 presents a new resolution parameterization more intuitive than previous approaches.

Both previous contributions aim to adapt the iterative approach proposed in Section 8.2.4 for uncalibrated multiview sequences without any significative change in the architecture. However, as already pointed out in [APTB+14], the use of hierarchies to reduce the set of possible unions of regions may excessively constrain the partition solution space. Therefore, we propose a new architecture consisting in a two-step iterative co-clustering that enlarges the set of possible partition solutions. Section 8.3.3 presents this new architecture.

Moreover, in a multiview scenario, since all images are available at the same time, a global optimization is more suitable to robustly capture inter relations between different views. However, in contrast to the iterative approach, high memory resources are required in a global optimization [XXC12]. To overcome this limitation, we propose to consider partitions resulting from the proposed two-step iterative co-clustering as inputs for the global optimization. In addition to the generic low-level features used in [GVB11, VAM15], semantic information, whenever available, can be used to drive the global optimization towards a set of coherent semantic partitions. Section 8.3.4.1 presents the architecture for the generic global optimization, whereas Section 8.3.4.2 presents how the semantic information is tackled in a semantic global optimization.

Finally, Section 8.3.5 presents an unsupervised resolution selection technique that, using the semantic information, obtains a single, multiview coherent labeling with an accuracy close to the multiresolution representation. A resolution selection technique is necessary in some applications, such as semantic segmentation, where a single resolution is required.

8.3.1 Adjacency based on motion cues

Adjacency definition is crucial in any co-clustering process. Whereas there is no ambiguity in region adjacency between regions belonging to the same partition, how adjacency is defined for regions belonging to different partitions has a direct impact on the inter-image interactions. Previous co-clustering approaches [VB10, GVB11, VAM15] define inter-image region adjacency as region overlapping without considering motion. For instance, in [GVB11, VAM15], contour elements belonging to different partitions that are at a distance greater than 10 pixels are not considered. Similarly, regions belonging to different partitions with no overlapping are not considered adjacent.

In order to robustly link objects through different views, we compute the optical flow between consecutive views using [BBM09]. As pre-processing the images with motion compensation before to be clustered results in another problem - how to infer the clusters in the original images from the motion compensated images -, we propose to introduce this motion information in the co-clustering optimization at two stages:

Similarity computation: Similarities between regions R_m and R_n from partitions P_i and P_j respectively, are computed comparing their contour elements ([GVB11, VAM15]). In our work, we propose to compare a given contour element c at position (x, y) with all contour elements close to $(x + f_x, y + f_y)$, where $f(x, y) = (f_x, f_y)$ is the optical flow. Note that in [GVB11, VAM15] (see Sections 8.1 and 8.2) a given contour element c at position (x, y) is compared with all contour elements close to that position in the other partitions since no motion information is used.

RAG definition: Regions R_m and R_n from partitions P_i and P_j respectively are consid-

ered adjacent if at least one pixel from the motion compensated version of R_m overlaps with a pixel of R_n. More specifically, whereas in [GVB11, VAM15] (see Sections 8.1 and 8.2) two regions from different partitions were defined as adjacent if the intersection of their set of pixel coordinates is not empty, now the pixel coordinates from R_m are motion compensated using the optical flow before computing their intersection with the set of pixel coordinates from R_n.

8.3.2 Resolution parameterization

In previous approaches [GVB11, VAM15], parameters to set the resolution of the co-clustered partitions are not intuitive and lead to non-homogeneous multiresolution representations. Consecutive resolutions obtained using the similarity multiplier λ from [GVB11] (see Equation 8.7) or the number of active boundaries N_b from [VAM15] (see Equation 8.13) may be almost equal or present very large variations. Therefore, it is hard to decide which values the previous parameters should take. That is the reason why we propose to use the number of clusters as an optimization parameter to set the resolution.

As seen in Equation 8.10, the merging of two sibling nodes is equivalent to set as inactive all the inter-sibling boundaries that form the common contour. Moreover, the number of regions is reduced by one with each merging. Therefore, a relation between the number of active boundaries and the number of clusters can be formulated. We have been able to compact all this information into a single constraint for the whole hierarchy.

To take into account each parent node p in H_i, it is sufficient to select a single arbitrary boundary ($D^p_{k,l}$) among its inter-sibling boundaries. The following constraint is imposed to set the resolution:

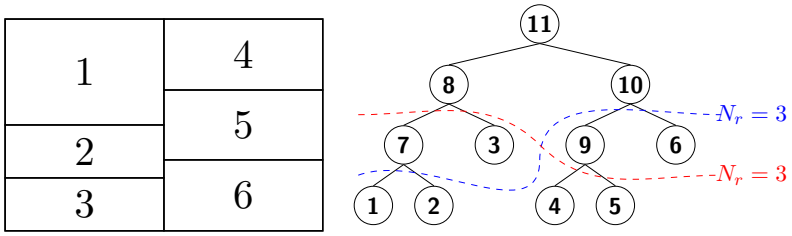
$$\sum_p D^p_{k,l} = N_r - 1 \tag{8.16}$$

where N_r is total number of clusters desired for this resolution. Whereas the hierarchical constraints presented in Equations 8.10 and 8.11 are imposed to each parent node from the hierarchy H_i, the resolution constraint given by Equation 8.16 is globally imposed to the hierarchy. Note that, in order to have a direct relation between active boundaries and number of regions, a single boundary of the inter-sibling boundaries is included in the constraint for each merging. This can be done because hierarchical constraint from Equation 8.10 ensure that, in order to merge two siblings, all inter-sibling boundaries that form their contour are forced to be inactive. Moreover, thanks to this hierarchical constraint, which inter-sibling boundary is selected for each parent node does not affect on the resulting set of possible solutions.

Figure 8.9 shows an illustrative example to help the reader to understand this constraint on the number of clusters. Note that the partition and the hierarchy are the same as the ones used in Figure 8.6 to explain the hierarchical constraints. First, let us arbitrarily select a single inter-sibling boundary for each parent node from the hierarchy. For parent node 11, we arbitrarily select $D_{1,4}$ among its inter-sibling boundaries ($\{D_{1,4}$, $D_{1,5}$, $D_{2,5}$, $D_{2,6}$, $D_{3,6}\}$). Analogously, we select $D_{2,3}$ for parent node 8, $D_{5,6}$ for parent node 10, $D_{1,2}$ for parent node 7 and $D_{4,5}$ for parent node 9. Therefore, Equation 8.16 becomes:

$$D_{1,2} + D_{2,3} + D_{4,5} + D_{5,6} + D_{1,4} = N_r - 1.$$

Equation 8.16 is jointly considered with the two hierarchical constraints (Equations 8.10 and 8.11), can be interpreted as the possible cuts that could be performed to the hierarchy resulting in N_r leaves nodes, where leaves nodes are the nodes with no children. Blue and red dashed lines in Figure 8.9 represent the possible cuts that could be performed to the hierarchy when $N_r = 3$. Note that the initial hierarchy has 6 leaves nodes, but once any of the cuts represented in the example is done, the resulting hierarchy only has 3 leaves nodes. The selection of the optimal cut depends on the optimization process and, therefore, on the intra-image and inter-image interactions.



Selected inter-sibling boundaries: $D_{1,2}, D_{2,3}, D_{4,5}, D_{5,6}, D_{1,4}$

Equation 8.16 for $N_r = 3$ regions: $D_{1,2} + D_{2,3} + D_{4,5} + D_{5,6} + D_{1,4} = 2$

Possible solutions for $N_r = 3$ regions: $D_{1,2} = 0, D_{2,3} = 1, D_{4,5} = 0, D_{5,6} = 0, D_{1,4} = 1$

$D_{1,2} = 0, D_{2,3} = 0, D_{4,5} = 0, D_{5,6} = 1, D_{1,4} = 1$

Figure 8.9: Illustrative example for the resolution constraint (Equation 8.16). $D_{1,4}$ has been arbitrarily selected from node 11's inter-sibling boundaries ($\{D_{1,4}, D_{1,5}, D_{2,5}, D_{2,6}, D_{3,6}\}$).

.

8.3.3 Generic two-step iterative co-clustering

The goal of imposing hierarchies in [VAM15] is to force the optimization process towards hierarchy nodes. However, as already pointed out in [APTB+14], the use of hierarchies to reduce the set of possible unions of regions may excessively constrain the partition solution space. For instance, in Figure 8.8, note that a single cluster representing clusters A and C would be more coherent among the different frames, but such a cluster would violate the hierarchical constraints imposed for each frame. Therefore, we propose a two-step iterative co-clustering that enlarges the set of possible partition solutions. Whereas the first step allows the process to reach a given resolution using hierarchy nodes, the second step introduces coherence to the final co-clustered partitions allowing region mergings that were not present in the hierarchy. The proposed architecture is presented next.

For each resolution, two optimization steps are coupled as represented by the block diagram in Figure 8.10. Let us denote π_i^* and π_i^{**} the optimal partitions resulting from

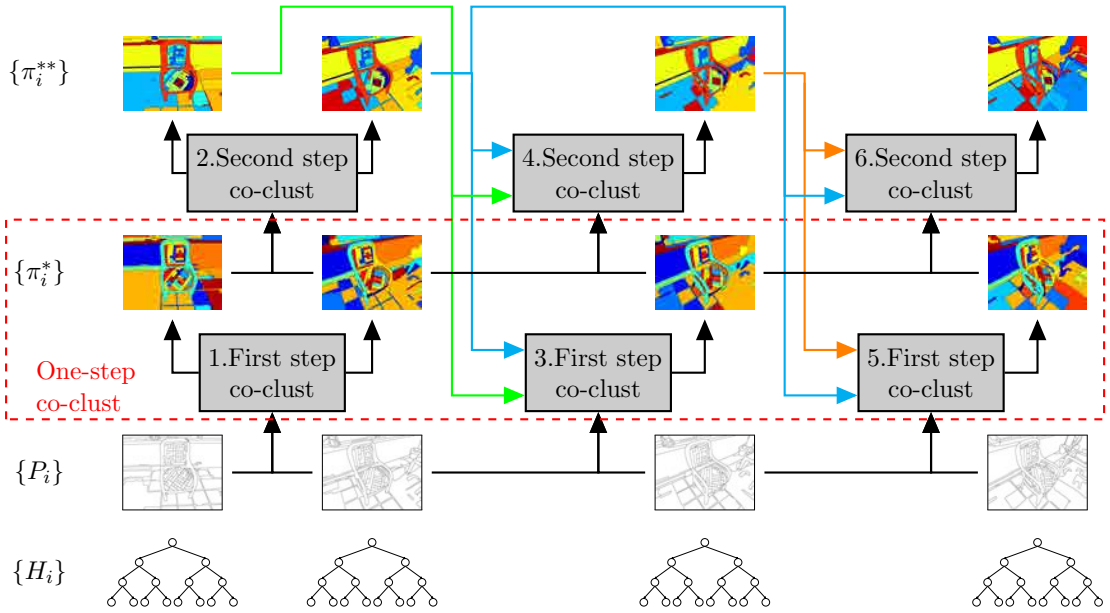


Figure 8.10: Two-step iterative co-clustering flowchart. The numbers of the blocks denote the order in which they are applied.

the first and second step respectively. Moreover, let us differentiate *intra* and *inter* partitions in any co-clustering step. On the one hand, inter partitions are the direct result from the co-clustering, such as $\{\pi_i^*\}$ from the first step or $\{\pi_i^{**}\}$ from the second step, where clusters are defined across the different partitions, i.e. regions from different partitions can belong to the same cluster. On the other hand, intra partitions have their clusters defined within the partitions, i.e. regions from different partitions do not belong any more to the same cluster. Intra partitions are obtained by activating all inter-image boundaries $D_{i,j}$ from the resulting co-clustering, i.e. any boundary $D_{i,j}$ representing a connection between regions from different partitions is activated ($D_{i,j} = 1$). Although intra partitions do not show explicitly the correspondences between the clusters across the partitions, these clusters have been obtained coherently because the inter image interactions have been taken into account in the optimization process.

In the first step, to obtain π_i^*, an iterative approach is applied in a similar way as done in [VAM15]. In our approach, partitions from the second step π_{i-2}^{**} and π_{i-1}^{**} are included in the optimization to keep coherence with the previous co-clustering results. Figure 8.11 shows an example to illustrate how this first step is applied. On the one hand, there are boundaries that are forced to be active: (*i*) intra-image boundaries connecting adjacent clusters from π_{i-2}^{**} ($D_{A,B}$ in Figure 8.11) , (*ii*) intra-image boundaries connecting adjacent regions from P_{i-1} that belong to different clusters in π_{i-1}^{**} ($D_{2,3}$ and $D_{3,4}$ in Figure 8.11), and (*iii*) inter-image boundaries connecting clusters from π_{i-2}^{**} with adjacent regions from P_{i-1}, where the region is assigned to a different cluster in π_{i-1}^{**} ($D_{A,3}$, $D_{B,2}$ and $D_{B,4}$ in Figure 8.11). Note that inter adjacency between clusters from π_{i-2}^{**} and regions from P_{i-1} is computed considering motion information as presented in Section 8.3.1. Otherwise, $D_{B,2}$ and $D_{B,4}$ would not be considered. In the example, this results in the constraint $D_{A,B} + D_{2,3} + D_{3,4} + D_{A,3} + D_{B,2} + D_{B,4} = 6$. On the other hand, some boundaries must be inactive: (*i*) intra-image boundaries connecting regions from P_{i-1} that belong to the

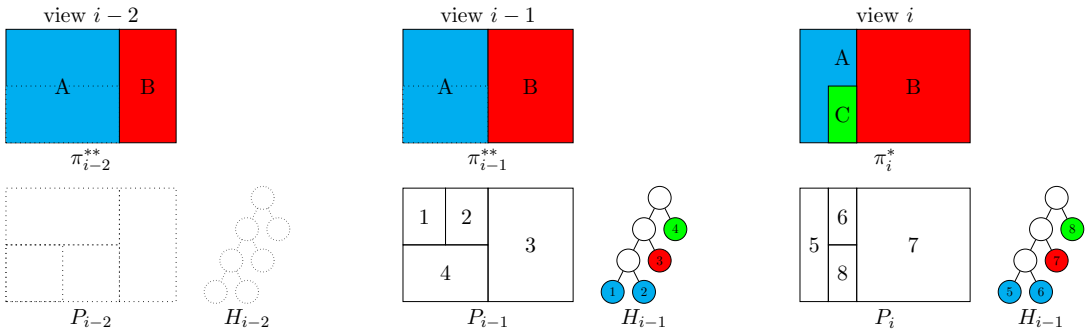


Figure 8.11: Example illustrating the need of the two-step iterative co-clustering (1st step). Regions are indexed by numbers while clusters are indexed by letters. Dashed boundaries in π^{**}_{i-2} and π^{**}_{i-1} represent π^{*}_{i-2} and π^{*}_{i-1} respectively, further used in Figure 8.12.

same cluster in π^{**}_{i-1} ($D_{1,2}$, $D_{1,4}$ and $D_{2,4}$ in Figure 8.11), and (*ii*) inter-image boundaries connecting clusters from π^{**}_{i-2} with regions from P_{i-1}, where the region is assigned to the same cluster in π^{**}_{i-1} ($D_{A,1}$, $D_{A,2}$, $D_{A,4}$ and $D_{B,3}$ in Figure 8.11). Note that π^{**}_{i-1} is used to relate regions from P_{i-1} with π^{**}_{i-2}. Following the same example, this results in the constraint $D_{1,2} + D_{1,4} + D_{2,4} + D_{A,1} + D_{A,2} + D_{A,4} + D_{B,3} = 0$.

These two iterative constraints are added to the optimization problem formulated in Equation 8.12, which is only applied to the set of partitions formed by π^{**}_{i-2}, P_{i-1} and P_i. Hierarchical constraints (Equations 8.10 and 8.11) are only imposed over P_i since iterative constraints are responsible for keeping coherence with the previous co-clustering results.

As hierarchies in $\{H_i\}$ are built independently, the optimal combination of hierarchy nodes to represent the scene may not be coherent among views, leading to oversegmentations or inconsistencies in the resulting partitions of the first step $\{\pi^{*}_{i}\}$. Note, for instance, that regions 5, 6 and 8 from P_i in Figure 8.11 cannot be assigned to the same cluster without also including region 7 due to the hierarchical structure. To palliate this effect, hierarchical constraints are not further used in the second step. Iterative constraints are analogous to those applied in the first step to keep coherence, but now considering π^{*}_{i-1} and π^{*}_{i} instead of P_{i-1} and P_i. More specifically, the intra partitions resulting from π^{*}_{i-1} and π^{*}_{i} are those being used. As in the first step, motion information is considered for inter adjacency. Figure 8.12 illustrates the second step of the co-clustering following the previous example. In this case, the constraints to be applied are $D_{A,B} + D_{1,2} + D_{2,3} + D_{A,2} + D_{B,1} + D_{B,3} = 6$ and $D_{A,1} + D_{A,3} + D_{B,2} + D_{1,3} = 0$. As hierarchical constraints are not further applied, regions 4 and 6 from π^{*}_{i} can now be assigned to the same cluster. Note that, as shown in Figure 8.10, first and second steps have to be alternated since the computation of π^{*}_{i} requires π^{**}_{i-1} and the computation of π^{**}_{i} requires π^{*}_{i}.

Figure 8.13 shows a real example where a teddy bear is not coherently segmented in the partition resulting from the first step (third column). In such example, we can observe that in one of the partitions resulting from the first step, the regions representing the head and the rest of the body do not belong to the same cluster, whereas in the other partition both regions have been assigned the same cluster. The reason why such regions

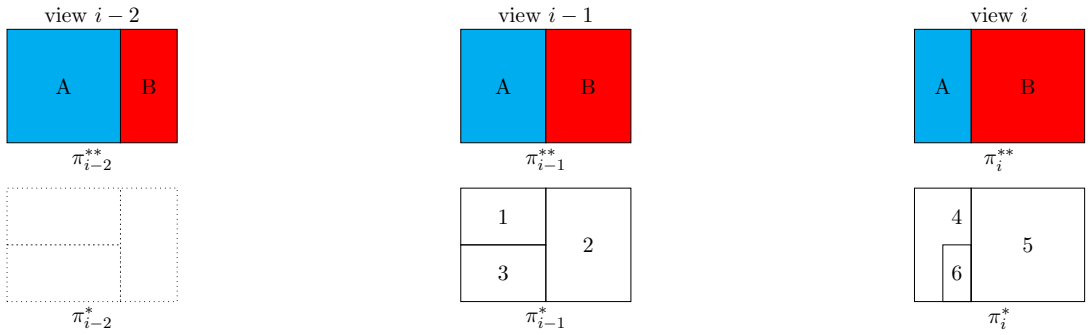


Figure 8.12: Example illustrating the need of the two-step iterative co-clustering (2nd step).

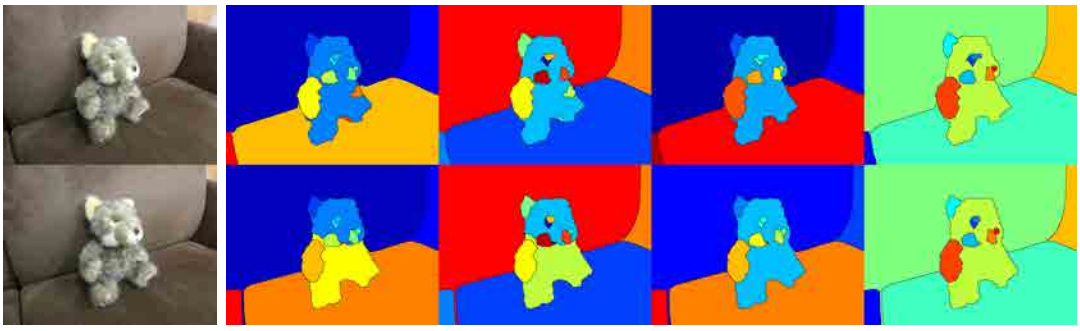


Figure 8.13: Motivation of two-step co-clustering. Second and third columns: intra and inter partitions from first step. Fourth and fifth columns: intra and inter partitions from second step.

have not been assigned the same cluster in one of the partitions is that the hierarchy structure has not allowed such a merging. In this specific example, the region associated to the body of the teddy is considered more similar to the background (the sofa) than to the teddy head. Therefore, due to the hierarchal constraint, if the teddy body and head regions should belong to the same cluster, then the background should be also assigned to such a cluster. It is to overcome this limitation that we propose the second step in which nodes that form clusters in $\{\pi^*_i\}$ can be merged without considering the hierarchy constraints. This way, the problem observed in the teddy bear example can be solved as shown in the last column of Figure 8.13.

8.3.4 Global co-clustering

In a multiview scenario, since all images are available at the same time, a global optimization is more suitable to robustly capture inter relations between different views. In contrast to the iterative approach, high memory resources are required in a global optimization [XXC12]. As a result, partitions with an arbitrarily large number of regions cannot be used. To overcome this limitation, partitions from higher levels of hierarchies could be considered as in [KLH12]. However, as these partitions are created independently, they may not coherently represent objects in the scene. Therefore, we propose to consider partitions resulting from the previous two-step iterative co-clustering as inputs for the global optimization. This section is structured in two parts: a generic approach

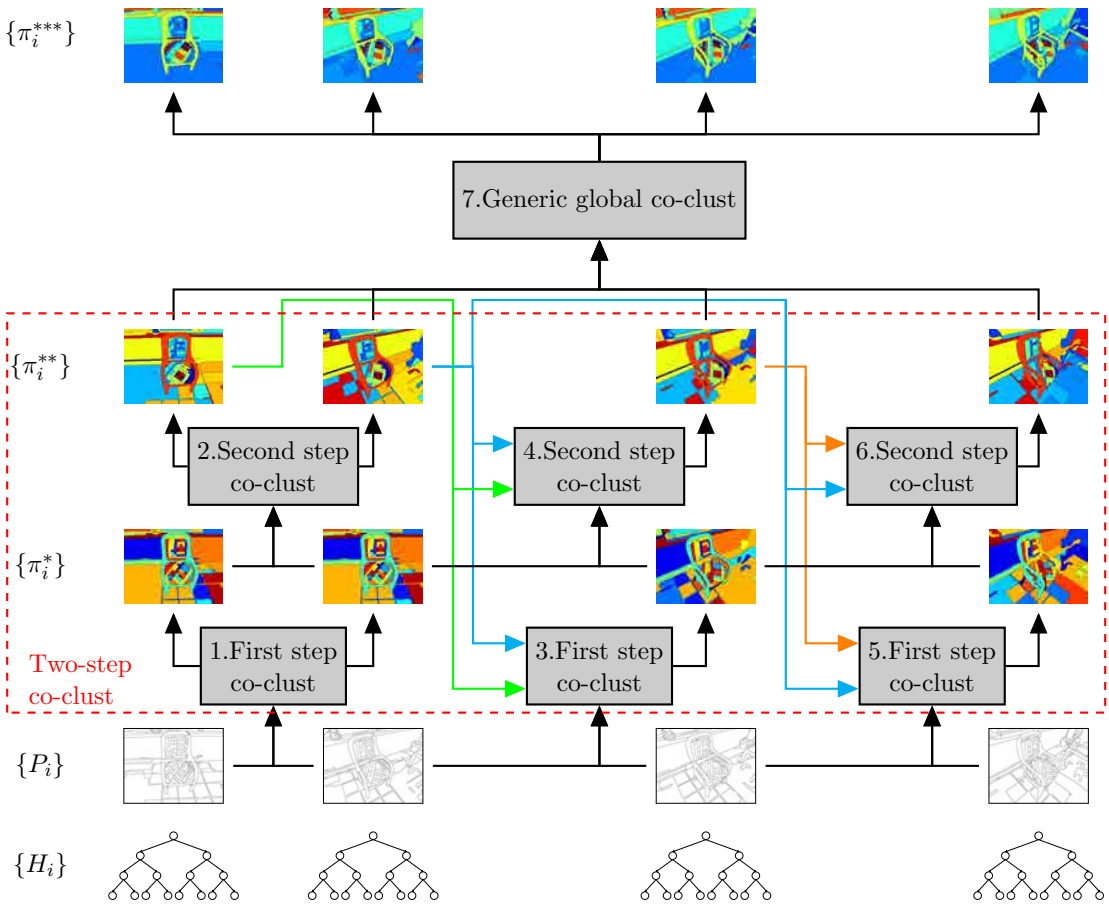


Figure 8.14: Global co-clustering flowchart. The numbers of the blocks denote the order in which they are applied.

(Section 8.3.4.1), where the same low-level features from the iterative approach are used, and a semantic approach (Section 8.3.4.2), where semantic information is included.

8.3.4.1 Generic global co-clustering

In order to robustly cluster regions from a set of multiview partitions using a global optimization, we propose to consider the intra partitions resulting from the two-step iterative co-clustering (see Section 8.3.3) as inputs for the global optimization. Intra partitions resulting from the second step optimization are obtained from $\{\pi_i^{**}\}$ inter partitions analogously as done for intra partitions from $\{\pi_i^{*}\}$ in Section 8.3.3, i.e. activating any boundary $D_{i,j}$ representing a connection between regions from different partitions. For each resolution, the optimization process from Equation 8.12 is jointly applied to all intra partitions $\{\pi_i^{**}\}$, where motion is also considered for inter adjacency and inter image similarities. Hierarchical and resolution constraints are not imposed since they have already been considered in the first step of the iterative co-clustering. Although all views are jointly processed, inter adjacency is constrained to the two previous and the two subsequent views in order to restrict the number of boundary variables in the optimization process. This restriction is also imposed since, in a multiview scenario, corresponding contour elements among a large number of views would show a significative

disparity of the normal vectors' orientation values assigned to them. Figure 8.14 shows a block diagram that represents the generic global co-clustering. Resulting partitions are denoted as $\{\pi_i^{***}\}$.

8.3.4.2 Semantic global co-clustering

The global optimization presented in Section 8.3.4.1 relies on the same low-level features used in the generic two-step iterative co-clustering (see Section 8.3.3). However, semantic information, whenever available, can be used to drive the global optimization towards a set of coherent semantic partitions. We propose a method that exploits the information provided by [ZJRP+15], which introduced a new form of convolutional neural network (CNN) that combines the strengths of CNNs and Conditional Random Fields. The convolutional neural network from [ZJRP+15] was trained to be applied to the challenging Pascal VOC 2012 segmentation benchmark [EVGW+10] and has its implementation available. We consider two results by [ZJRP+15]: the semantic segmentations, where every pixel from the image is assigned a semantic category, and the confidence scores for each category.

The proposed semantic global optimization is also based on the coherent partitions resulting from the generic two-step iterative co-clustering $\{\pi_i^{**}\}$, from which the resulting intra partitions are considered. Regions from these partitions are assigned a semantic label as follows: each pixel is assigned the semantic class from the CNN confidence scores with higher confidence at its position. This confidence should be above a certain threshold (T_{sp}). Otherwise, no semantic label is assigned. Then, each region is labeled with the predominant semantic class over its pixels if this percentage is larger than T_{sr}. As in the previous case, if no class fulfills this condition, no label is assigned to the region. Figure 8.15 shows an illustrative example for semantic category assignment to regions for a 4×4 pixel region and 5 possible semantic categories. Finally, the following similarity penalizations and optimization constraints are only imposed over regions to which a semantic label has been assigned:

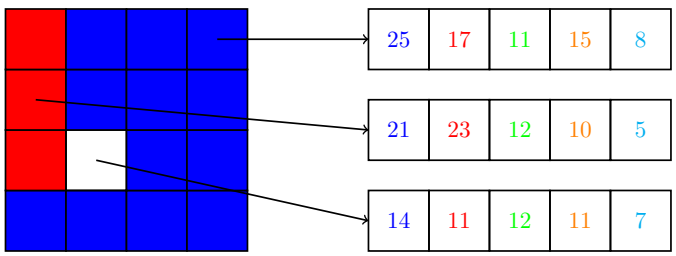


Figure 8.15: Semantic class assignment to regions. Illustrative example for a 4×4 pixel region where pixels are assigned one of five possible categories or left as not assigned if any of the predictions is trustful. The five different colors corresponds to the five possible categories and white is left for unassigned pixels. 12 out of 16 pixels have been assigned to category C_1 (blue), 3 out of 16 to C_2 (red) and 1 out of 16 has not been assigned. In this example, a pixel is left assigned if none of the classification scores is above $T_{sp} = 15$. The rest of pixels are assigned to the category with the highest score. For $T_{sr} = 70$, category C_1 is assigned to this region because 75% of pixels have been assigned to C_1.

- Similarity penalizations: Fusions between regions R_k and R_l from the same partition belonging to different semantic labels are penalized. Since thier similarity is encoded by $Q_{k,l}$ (see Section 8.1), a constant K_s is subtracted to $Q_{k,l}$.

- Optimization constraints: Two constraints are included in the optimization process to introduce the semantic information. First, adjacent regions from the same partition with the same semantic label must be merged. Therefore, their intra-image boundaries must be inactive:

$$\sum_{k,l} D_{k,l} = 0$$

where k, l are adjacent regions from the same partition which have been assigned the same semantic category. Second, adjacent regions from different partitions with different semantic labels cannot be assigned to the same cluster. Therefore, thier inter-image boundaries must be active:

$$\sum_{k,l} D_{k,l} = N_{sep}$$

where k, l are adjacent clusters from the different partitions which have been assigned different semantic labels and N_{sep} is the cardinality of these variables, i.e. the number of pairs of adjacent clusters from different partitions with different semantic labels.

Figure 8.16 shows the block diagram for the semantic co-clustering, where the main difference with respect to the flowchart presented in Figure 8.14 for the generic global co-clustering is that the semantic partitions $\{SP_i\}$ from [ZJRP+15] are also taken as input.

8.3.5 Semantic-based resolution selection

Our approach creates a multiresolution of co-clustered partitions, providing a rich framework for image and video analysis [APTB+14, GKHE10]. However, in some applications such as semantic segmentation, a single resolution is required. For such cases, we propose a semantic-based method for automatic resolution selection.The proposed selection method is based on the semantic segmentation [ZJRP+15] already used for semantic global co-clustering in Section 8.3.4.2.

First, for each semantic label l, we select the clusters that maximize the Jaccard index with respect to the mask formed by all pixels detected as l. If the same cluster is selected for different semantic labels, it receives the label l^* that maximizes the sum of the confidence scores over the cluster. Figure 8.17 shows an illustrative example where the same cluster C_2 is selected for two semantic labels l_1 and l_2 and how the conflict is tackled. Regarding the cluster selection process, the union of clusters C_1 and $C2$ maximizes the Jaccard index with respect l_1 ($J = 20/24$) and the cluster C_2 maximizes the Jaccard index with respect l_2 ($J = 4/12$). As a result, cluster C_2 is selected for both semantic labels l_1 and l_2. The conflict is solved by adding the confidence scores for each category across the cluster. Let us suppose that the pixels detected as l_1 have

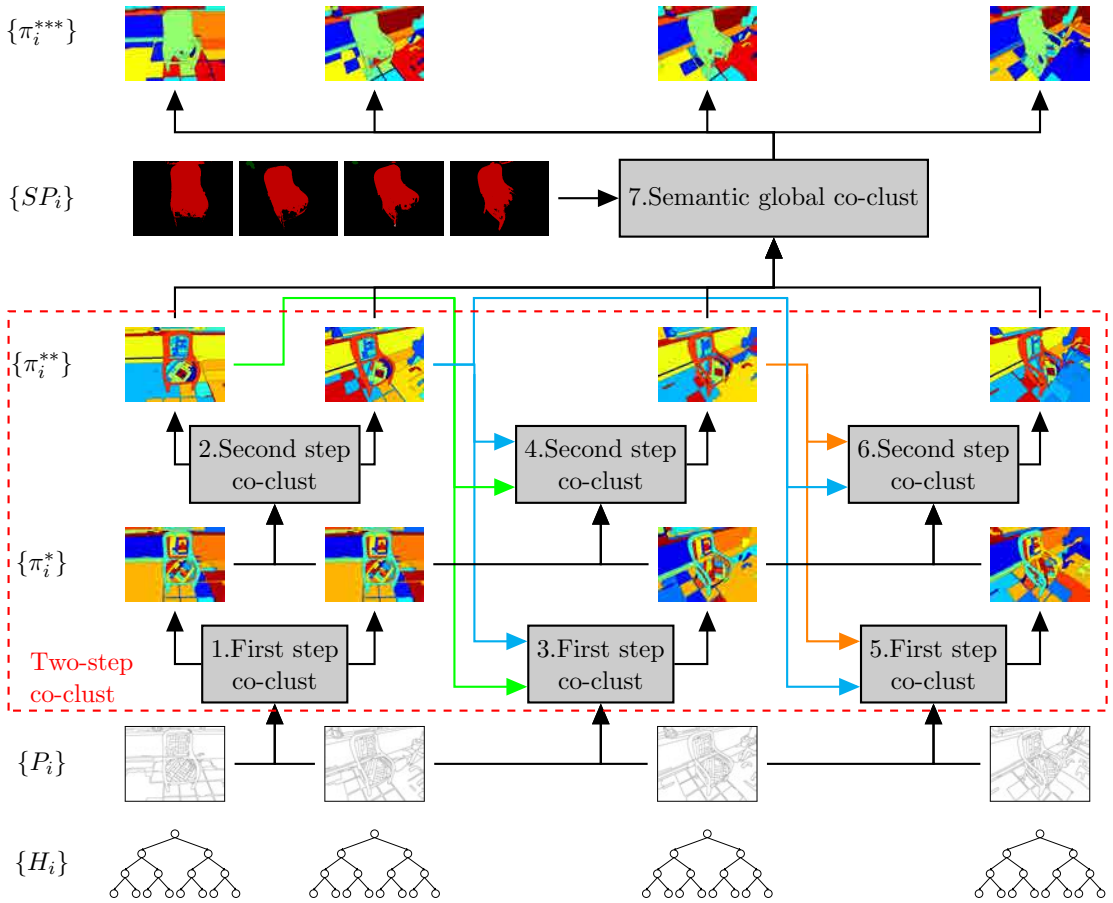


Figure 8.16: Semantic global co-clustering flowchart. The numbers of the blocks denote the order in which they are applied.

confidence scores 25 and 15 for l_1 and l_2 respectively, the ones detected as l_2, their confidence scores are 20 and 22 for l_1 and l_2 respectively, and the ones with no category assignment have confidence scores 10 for both categories. The addition of l_1 scores for C_2 is $8*25+4*20=280$, whereas the addition of l_2 scores for C_2 is $8*15+4*22=208$. Therefore, the semantic category l_1 is assigned to C_2. The other selected cluster, C_1, is directly assigned the semantic label l_1 since there is no conflict to be solved.

Then, a foreground score s_{fg} is computed as the addition of the confidence scores for all the selected clusters for their respective semantic labels. Since all pixels have also associated a background confidence score in [ZJRP+15], the set of unselected clusters is also considered to compute a background score s_{bg} by adding their background confidence scores. Finally, the score for a given resolution is obtained as $s_{fg}+s_{bg}$. This process is performed for each resolution and the resolution with the greatest score is selected as the proposed single resolution co-clustering.

Figure 8.18 shows an illustrative example where the problem about considering or not the background is tackled. In this example, it is shown that when the background is not considered, the resolution selection method is biased to resolutions with selected clusters covering the largest possible image area instead of the clusters that best fit the objects in the scene. As in the previous semantic label assignment to clusters example (Figure 8.17),

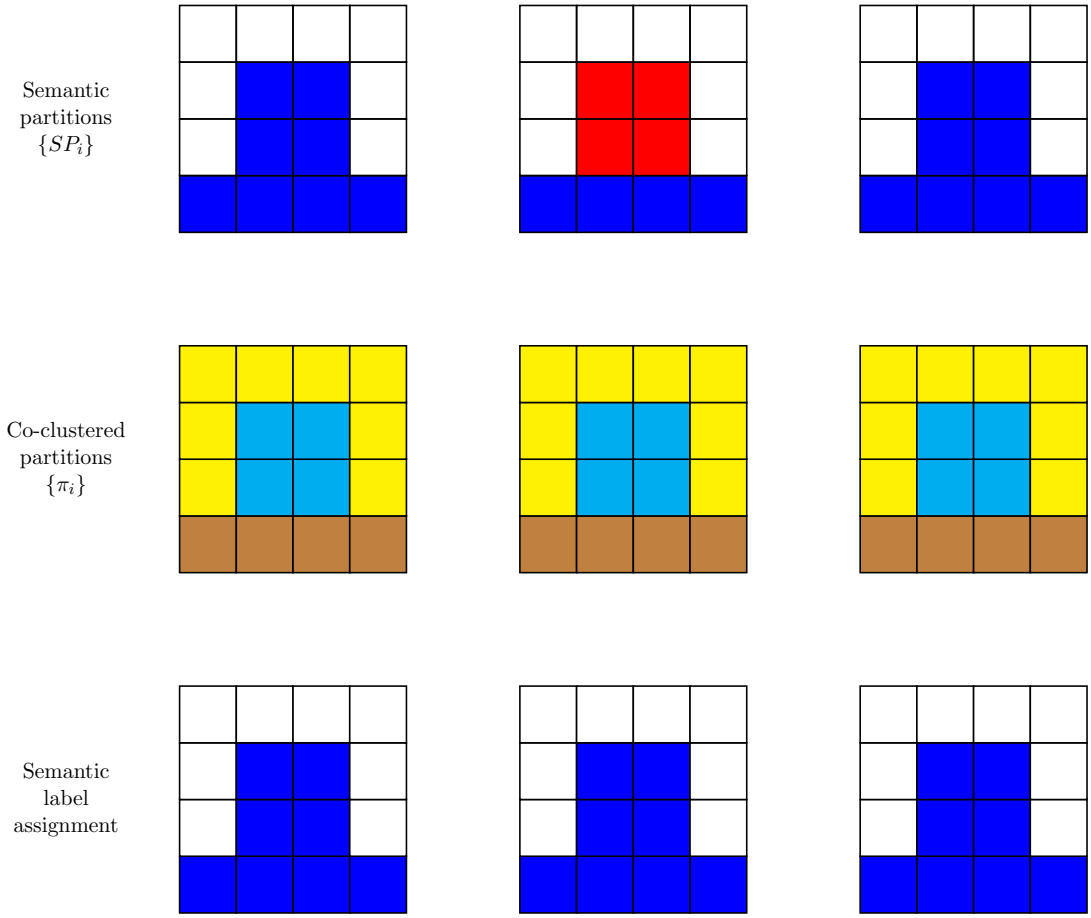


Figure 8.17: Semantic label assignment to clusters. Illustrative example where two semantic categories l_1 (blue) and l_2 (red) are detected in the semantic partitions $\{SP_i\}$ and the co-clustered partitions $\{\pi_i\}$ resulting from the any generic or semantic co-clustering have been segmented into three clusters C_1 (brown), C_2 (cyan) and C_3 (yellow). As a result, the semantic category l_1 is assigned to cluster C_1 and C_2 whereas no semantic category is assigned to cluster C_3.

let us suppose that pixels detected as l_1 (blue) have confidence scores of 25 and 15 for l_1 and l_2 respectively, the ones detected as l_2 (red) have confidence scores of 20 and 22 for l_1 and l_2 respectively, and the ones with no category assignment have confidence scores of 10 for both categories. Furthermore, let us suppose a background score of 30 for unassigned pixels and 10 for the rest. Let us also suppose that we have a resolution r_1 where the cluster C_1 (brown) includes part of the background and another resolution r_2 where the clusters fit better the object according to Jaccard criteria. Following the same reasoning as in Figure 8.17, l_1 is assigned to C_1 at resolution r_1 and to C_1 and C_2 at resolution r_2. If we only consider the foreground score s_{fg}, confidence scores for l_1 are added along C_1 for r_1 and C_1 and C_2 for r_2. This way, for resolution r_1, $s_{fg} = (8 * 25 + 5 * 10) + (4 * 25 + 4 * 20 + 5 * 10) + (8 * 25 + 5 * 10) = 730$, whereas for resolution r_2, $s_{fg} = (8 * 25) + (4 * 25 + 4 * 20) + (8 * 25) = 580$. Therefore, despite being $J_1 = 24/39 < J_2 = 24/24$, it is the resolution r_1 the one that maximizes s_{fg}. This is because background is not being considered and, as a result, the selection is biased to larger clusters. On the other hand, when background is also considered and background

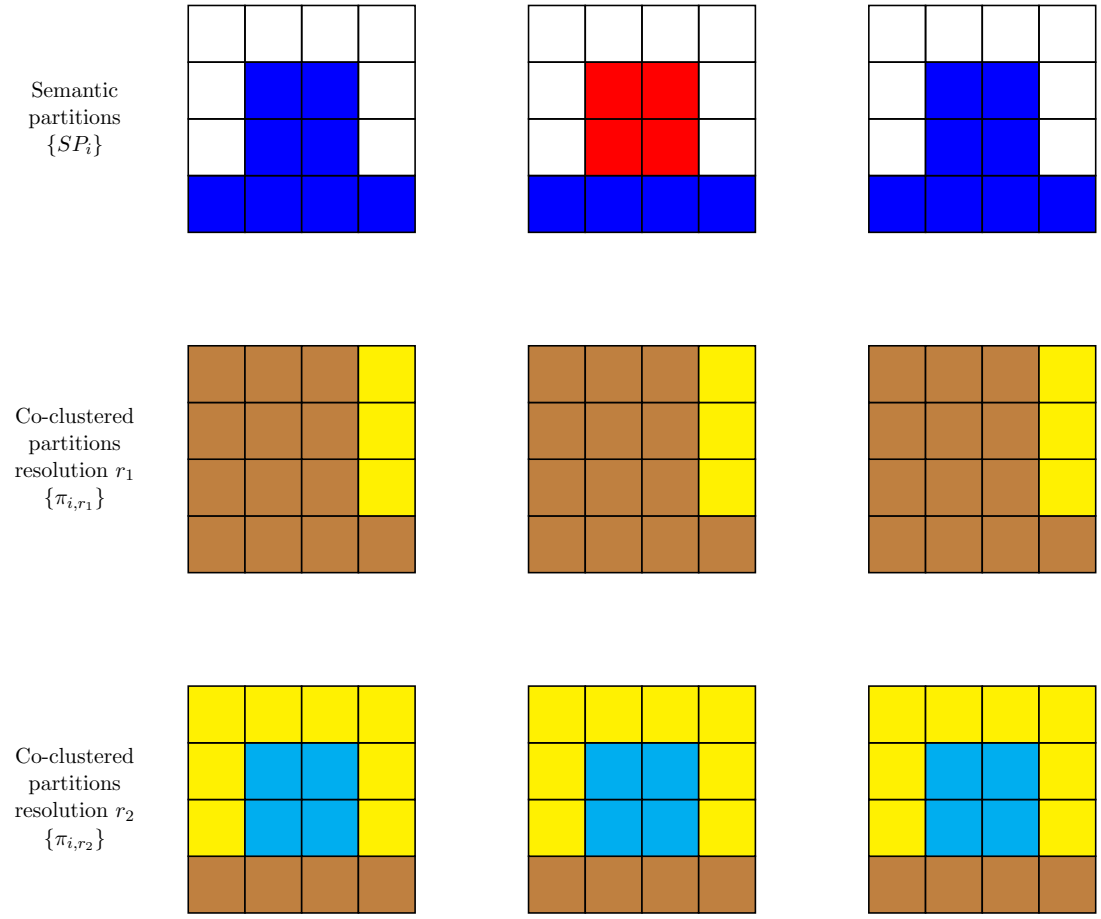


Figure 8.18: Considering background for resolution selection. First row: semantic partitions obtained by any semantic segmentation technique. Second and third row: co-clustered partitions obtained at two different resolutions r_1 and r_2 by any generic or semantic co-clustering technique.

scores are added for unselected clusters, for resolution r_1, $s_{bg} = 3*30+3*30+3*30 = 270$, resulting in $s_{fg}+s_{bg} = 730+270 = 1000$, whereas for resolution r_2, $s_{bg} = 3*8*30 = 720$, resulting in $s_{fg} + s_{bg} = 580 + 720 = 1300$. As a result, when background is considered, it is the resolution r_2 the one that maximizes the score.

This resolution selection method can be applied to both generic and semantic co-clusterings. Although it needs a semantic segmentation, this fact does not imply that the semantic segmentation must be also used in the co-clustering algorithm as proposed in Section 8.3.4.2.

The same method for resolution selection also provides a multiview semantic segmentation since, once all conflicts have been solved, each cluster is assigned only one semantic label. The last column of Figure 8.17 illustrates how the semantic label assignment to clusters results in a semantic segmentation. Whereas a different semantic segmentation could be obtained for each resolution, we propose to use the automatic resolution selection method to obtain a single resolution semantic segmentation. This method has been used to obtain the semantic segmentations shown in Figure 7.2 (last row).

Experimental results

9

The experiments have been carried out over six datasets proposed in [KSS12], where each dataset consist of a set of images captured around an object of interest (a car, a couch, a motorbike, a teddy bear and two kind of chairs), which is fully visible in every image. A subset of viewpoint images are shown for each dataset in Figure 9.1.

In [KSS12], the evaluation is performed as a task of object segmentation. Given a ground truth object segmentation for each viewpoint image, each resulting segmentation is evaluated as the accuracy of the pixels labeled as object. The accuracy is computed as the intersection over the union of the pixels predicted as object and the pixels labeled as object in the ground truth. However, the co-clustering algorithm does not aim at an object segmentation, but a set of regions in correspondence in a multiresolution partition. Therefore, using the same evaluation measure as in [KSS12] would ignore these attributes that differentiate the coclustering from any other cosegmentation technique.

That is the reason why we propose to extend the classic evaluation measure used in co-clustering algorithms. In [GVB11], a single reference frame is assessed as the number of selected regions required (referred as *efficiency*) to achieve a minimum Jaccard (referred as *consistency*). However, this approach does not take into account coherence between this partition and those from other views. As [KSS12] provides ground truth annotations for each view, the concept of efficiency is extended to selected clusters and Jaccard is computed over the entire set of annotations. The result of this evaluation is a curve which is summarized by the *Area Under the Curve* (AUC) figure.

The experiments have been performed using this evaluation measure and considering the co-clustering from [VAM15] (presented in Section 8.2) but including motion compensation (see Section 8.3.1) and clusters parameterization (see Section 8.3.2) as baseline. The six datasets from [KSS12] over which the experiments have been performed can be classified into two categories: (*i*) 180° multiview sequences, and (*ii*) 360° multiview sequences. Couch, GardenChair and Teddy (see third, fourth and sixth rows from Figure 9.1) datasets belong to 180° multiview sequences since the images have been only taken from frontal viewpoints. On the other hand, BMW, Chair and Motorbike (see first, second and fifth rows from Figure 9.1) datasets are 360° multiview sequences, where the objects have been captured from viewpoints all around the object of interest. No matter what category the sequences belong to, the change of angle between consecutive viewpoints is approximately constant. Since 360° multiview sequences consist of about 40 images and 180° multiview sequences consist of about 15 images, the change of angle between consecutive viewpoints is about 9°. Whereas 180° multiview sequences have been processed as a single block, we have decided to divide the 360° sequences into blocks of 10 consecutive image viewpoints, where each block would represent a 90° multiview sequence. The reasons for this division is twofold. First, regions from 360° sequences

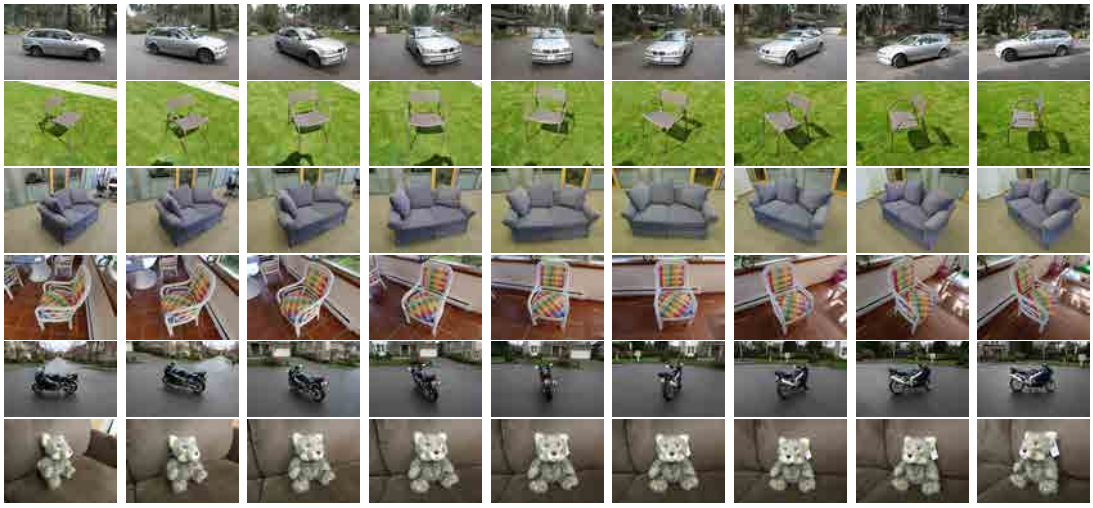


Figure 9.1: Subset of image viewpoints from six datasets proposed in [KSS12].

are hard to be correctly clustered along sequences that suffer significative changes in the viewpoints, specially when an iterative approach is considered. Second, dividing long sequences into blocks reduces the complexity of the algorithm as already done in state-of-the-art segmentation techniques such as [XXC12]. In order to have a fair comparison with the other state-of-the-art techniques, 360° sequences have been divided into blocks for all techniques being assessed. Although each block could be considered as an independent dataset, such a consideration would give more relevance to 360° multiview sequences than to 180° ones. Therefore, the curves obtained from the assessment of blocks belonging to the same dataset are averaged, resulting in one only curve for each of the six datasets used in the experiments. Then, the curves obtained for each dataset are again averaged whenever an overall assessment of a technique is desired.

More specifically, each 180° multiview sequence and each block from the 360° multiview sequences is assessed in the following way. Given a number of clusters to be selected N_{sc} where $1 \leq N_{sc} \leq 20$, a Jaccard index J is obtained by comparing the ground truth annotation mask M_{GT} and the mask $M_{N_{sc},r}$ that results from the union of at most N_{sc} of clusters belonging to the co-clustered partition at a given resolution r. $M_{N_SC,r}$ is obtained as the selection of clusters that maximize the Jaccard index with respect to M_{GT} using the implementation from [PTM12]. Note that the number of clusters to be selected is at most N_{sc}, but it is not constrained to be exactly N_{sc}. If the number of selected clusters that maximizes the Jaccard index N^*_{sc} is smaller than N_{sc}, then $M_{N_SC,r}$ is obtained as the union of these $N^*_{sc} < N_{sc}$ clusters. As a result, for each resolution r and each number of selected clusters N_{sc}, a Jaccard index is obtained. Finally, for each number of selected clusters N_{sc}, the resolution with the greatest Jaccard index is considered. Therefore, (N_{sc}, J) points of the evaluation curves can belong to different resolutions. Thus, the evaluation curves represent the upperbound given by the multiresolution co-clustered partition.

Regarding the experiments that have been performed using the proposed co-clustering framework, the leaves partitions $\{P_i\}$ have been obtained by applying the gPb-owt-ucm algorithm [AMFM11] and performing a cut on the hierarchy so that the initial over-segmentations consist of 200 regions. Furthermore, 22 different resolutions r have been

considered to obtain the multiresolution co-clustered partitions (see Section 8.3.2), where $r \in \{2, 4, 6, 8, 10, 12, 14, 16, 18, 20, 22, 24, 26, 28, 30, 40, 50, 60, 70, 80, 90, 100\}$.

This chapter is structured as follows. First, Section 9.1 presents the experiments that have been performed in a generic approach without using semantic information. Then, Section 9.2 shows the gain obtained in the experiments when semantic information is included in the global optimization. Furthermore, the semantic-based resolution selection method is assessed. Finally, some qualitative results are also provided in Section 9.3, including two more datasets (Ballet and Breakdancers) from [ZKU+04] and the Video Occlusion/Object Boundary Detection Dataset [GVB11].

9.1 Generic co-clustering

In this section, we assess the proposed algorithms in a generic segmentation context; that is, without using semantic information. Three different configurations of the proposed co-clustering are compared:

- *Two-step iterative co-clustering* (I-2S) (see Section 8.3.3).
- *UCM followed by one-step iterative co-clustering* (UCM+I-1S): The result of the first step in the two-step iterative co-clustering (I-2S) is replaced by a cut over the UCM hierarchy [AMFM11] that provides the same number of regions, i.e. the same resolution. As the hierarchies have been built independently for each view, coherence between the partitions resulting from the cuts performed in the first step cannot be assumed. This technique has been considered to show if it is better either to do a cut independently for each hierarchy and apply a single step iterative co-clustering over the resulting partitions or including the cut decision in the first step of the two-step iterative co-clustering with the proposed resolution parameterization presented in Section 8.3.2.
- *Two-step iterative co-clustering followed by a generic global optimization* (I-2S+GG): The algorithm presented in Section 8.3.4.

Furthermore, state-of-the-art methods in the fields of video segmentation [GKHE10, XXC12, GCS13] and co-segmentation [JBP12, KX12] are also evaluated. We also propose two baseline approaches: (i) the iterative algorithm in [VAM15], introducing motion information (Section 8.3.1) and resolution parameterization (Section 8.3.2), which will be referred as one-step iterative co-clustering (I-1S), and, (ii) a system that propagates labels from regions obtained with gPb-owt-ucm [AMFM11] using [BBM09] (UCM+P), as done in [GCS13, VAM15]. The results obtained by [VAM15] when directly applied to the addressed problem are not included since it is thought that it is not fair to compare an approach not using motion cues with motion-based techniques. That is the reason why I-1S is considered instead.

As shown in Figure 9.2, the proposed two-step algorithms (I-2S and I-2S+GG) outperform all the state-of-the-art techniques. Only [JBP12] achieves similar performance when a single cluster is considered. Although including a global optimization after the two-step iterative co-clustering (I-2S+GG) generates better correspondences for few clusters, not including it (I-2S) results in a better performance when a larger number of clusters

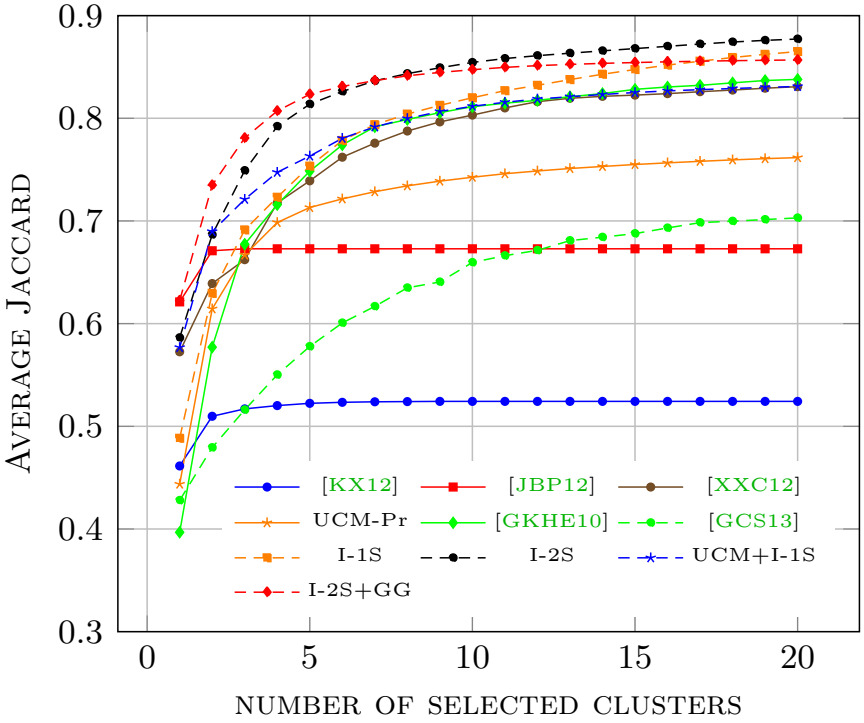


Figure 9.2: Evaluation of generic co-clustering with state-of-the-art videosegmentation and co-segmentation techniques

	I-2S	UCM+I-1S	I-2S+GG	[KX12]	[JBP12]	[XXC12]	[GKHE10]	[GCS13]	UCM+Pr	I-1S
BMW	**0.721**	0.683	0.702	0.417	0.563	0.701	0.645	0.633	0.622	0.665
Chair	0.787	0.768	0.759	0.533	0.782	**0.801**	0.764	0.474	0.591	0.776
Couch	0.932	**0.953**	0.943	0.782	0.900	0.850	0.875	0.728	0.891	0.895
GardenChair	0.838	0.629	**0.865**	0.308	0.515	0.699	0.677	0.629	0.839	0.797
Motorbike	0.762	**0.769**	0.765	0.391	0.391	0.711	0.727	0.464	0.540	0.704
Teddy	0.918	0.919	**0.920**	0.688	0.870	0.884	0.844	0.849	0.822	0.897
Average	**0.826**	0.787	**0.826**	0.520	0.670	0.774	0.755	0.630	0.718	0.789

Table 9.1: Comparison between the different configurations using the area under the curve (AUC) evaluation measure for proposed generic co-clustering techniques and state-of-the-art techniques. AUC values per dataset and averaged over the six datasets from [KSS12] are given.

are selected. On the other hand, the one-step iterative co-clustering baseline (I-1S) also outperforms state-of-the-art techniques, but with a lower performance with one cluster.

In Table 9.1, the figure *Area Under the Curve* (AUC) per dataset and averaged over the six datasets is presented. According to it, both proposed two-step co-clustering algorithms (I-2S and I-2S+GG) give the best performance on average. It is also remarkable that the use of the co-clustering in the first step instead of the UCM (I-2S vs UCM+I-1S) leads to a better performance due to the more coherent partitions achieved by the co-clustering in comparison with the UCM.

9.2 Semantic co-clustering

In this section, the techniques related with the use of semantic segmentations as side information presented in Sections 8.3.4.2 and 8.3.5 are evaluated. The following values for the parameters have been used: $T_{sp} = 15$, $T_{sr} = 70$ and $K_s = 1000$. As defined in Section 8.3.4.2, T_{sp} is the minimum score so that a semantic class is assigned to a pixel, T_{sr} is the minimum percentage so that the predominant semantic class is assigned to a region and K_s is the similarity penalization parameter for adjacent regions with different semantic classes. Figure 9.3 shows the comparison between the following proposed techniques:

- *Two-step iterative co-clustering followed by a generic global optimization* (I-2S+GG). Both multiresolution (MR) and single resolution (SR) obtained by using the automatic resolution selection presented in Section 8.3.5 are assessed.
- *Two-step iterative co-clustering followed by a semantic global optimization* (I-2S+SG). Both multiresolution (MR) and single resolution (SR) are also assessed.
- *Co-clustering based semantic segmentation* (CSS). Two configurations are considered: (*i*) the semantic segmentation obtained from the generic co-clustering I-2S+GG (SR) and denoted as GCSS, and (*ii*) the semantic segmentation obtained from the semantic co-clustering I-2S+GG (SR) and denoted as SCSS.

As CNN [ZJRP+15] and the proposed CSS are semantic segmentation techniques, there are no correspondences between the regions from different viewpoints. In order to perform a fair evaluation, the regions from the different views classified as the same semantic category have been considered as a single cluster, i.e. establishing artificial correspondences between them. Otherwise, semantic segmentation techniques would be significatively penalyzed since each considered cluster would belong to only one viewpoint. However, notice that these artificial correspondences can be arbitrarily applied in these six datasets because there are not more objects belonging to the same semantic category as the object of interest in the same dataset.

In Figure 9.3, it can be observed that for more than four clusters the best resolution of the semantic co-clustering (I-2S+SG (MR)) outperforms the average Jaccard obtained by CNN [ZJRP+15]. We have also assessed the semantic co-clustering when the resolution is automatically selected (I-2S+SG (SR)) and the performance is still better than CNN when eight or more clusters are considered. Furthermore, note that the proposed automatic resolution selection of the semantic co-clustering (I-2P+SG (SR)) outperforms the generic multiresolution co-clustering (I-2S+GG(MR)). In fact, if we also use the semantic information to automatically select a resolution from the generic multiresolution co-clustering, the difference between both single resolution semantic and generic co-clusterings increases significatively. Therefore, whenever a single resolution co-clustering is desired, the best option is including also the semantic information in the global optimization process as this information is exploited by the automatic resolution selection.

Finally, proposed co-clustering based semantic segmentations (GCSS and SCSS) have been compared with CNN [ZJRP+15]. Whereas the CNN [ZJRP+15] gives an average

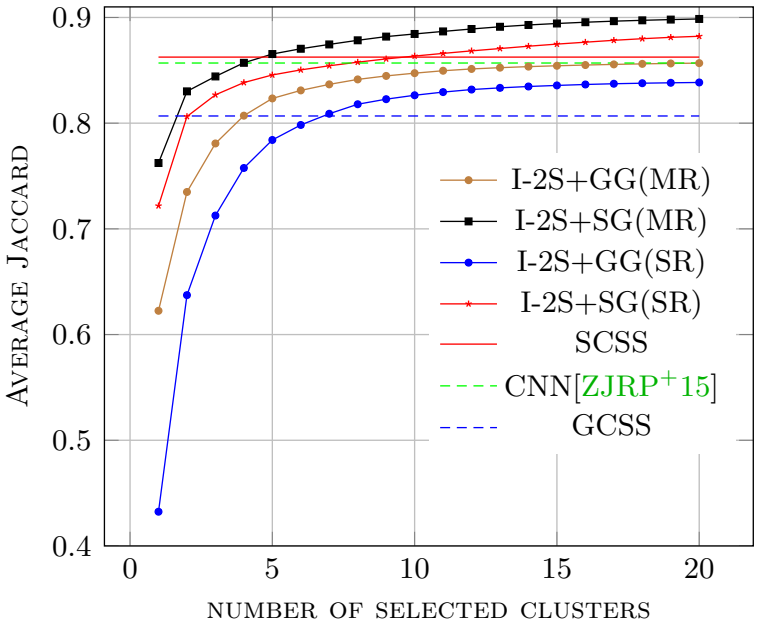


Figure 9.3: Evaluation of semantic co-clustering and generic co-clustering including global optimization. Results for resolution selection and semantic segmentation are also provided.

Jaccard of 85.69%, our Semantic Co-clustering based Semantic Segmentation (SCSS) gives a better average Jaccard of 86.25%. SCSS has a standard deviation of 9.23, being therefore more robust than CNN with a standard deviation of 14.09 over the six datasets. Generic Co-clustering based Semantic Segmentation (GCSS) is considered as baseline.

Previously, we have assessed the proposed techniques and compared them with five state-of-the-art and two baseline approaches, using a common co-clustering evaluation measure; namely, the maximum achievable Jaccard index obtained for a given number of regions (clusters) selected from the multiresolution representation. Nevertheless, a similar evaluation could be carried out looking at the problem of multiple view joint segmentation as a problem analogous to the video segmentation task. This way, we evaluate all the previous techniques using the Volume Precision-Recall measure [GSNJC+13], giving the curve that represents their upper bound through the different resolutions.

Given a computer generated segmentation $\mathbb{S}$ and a ground truth segmentation $\mathbb{G}$, normalized Precision P_{norm} and normalized Recall R_{norm} measures are defined as

$$P_{norm} = \frac{\left(\sum_{s\in\mathbb{S}} max_{g\in\mathbb{G}}|s\cap g|\right) - max_{g\in\mathbb{G}}|g|}{|\mathbb{S}| - max_{g\in\mathbb{G}}|g|}$$

$$R_{norm} = \frac{\sum_{g\in\mathbb{G}} \left(max_{s\in\mathbb{S}}|s\cap g| - 1\right)}{|\mathbb{G}| - \Gamma_{\mathbb{G}}}$$

where $\cap$ denotes the intersection operator, $|.|$ denotes the number of pixels in the volume, s represents each region from $\mathbb{S}$, g represents each region from $\mathbb{G}$ and $\Gamma_{\mathbb{G}}$ is the number of ground truth volumes in $\mathbb{G}$. Whereas Precision measures how well ground truth

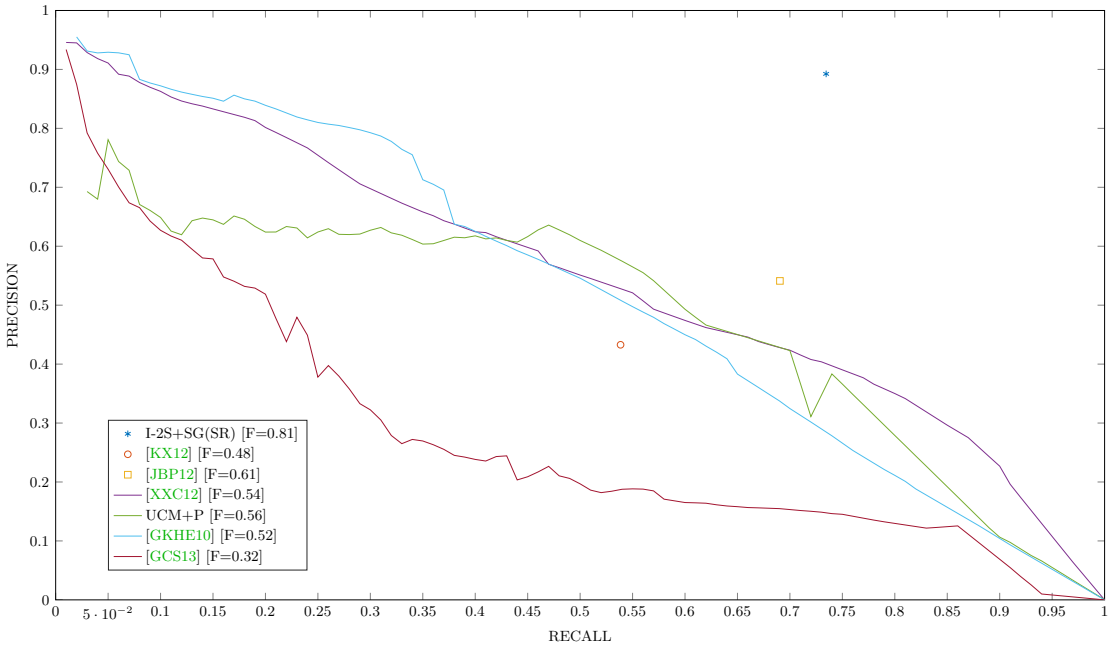


Figure 9.4: Volume Precision-Recall evaluation. I-2S+SG(SR) corresponds to the two-step iterative co-clustering followed by a semantic global optimization with the automatic resolution selection method. The F-measure is also provided for each technique in the legend.

annotations can be generated as a union of clusters (no matter the number of clusters), the Recall measures how well a single cluster fits with the ground truth annotation. Since oversegmentations lead to ideal Precision ($P_{max} = 1$) and a single cluster covering the whole image to ideal Recall ($R_{max} = 1$), both measures are normalized.

Figure 9.4 shows the Volume Precision-Recall for the five state-of-the-art techniques and the proposed semantic co-clustering (I-2S+SG(SR)). Whereas the techniques resulting in multiresolution partitions (video segmentation techniques [GKHE10, XXC12, GCS13]) are represented by curves (each resolution results in an evaluation point), co-segmentation techniques [JBP12, KX12] and I-2S+SG(SR) are represented by a single point since a single resolution is assessed. Furthermore, all techniques are also assessed with the F-measure, which is defined as follows:

$$F = \frac{2 \cdot P \cdot R}{P + R}$$

F-measure considers both precision and recall and allows the comparison of the different techniques. Whereas F-measure $\in [0.32, 0.61]$ for state-of-the-art techniques, our automatically selected resolution (I-2S+SG(SR)) reaches an F-measure of 0.81.

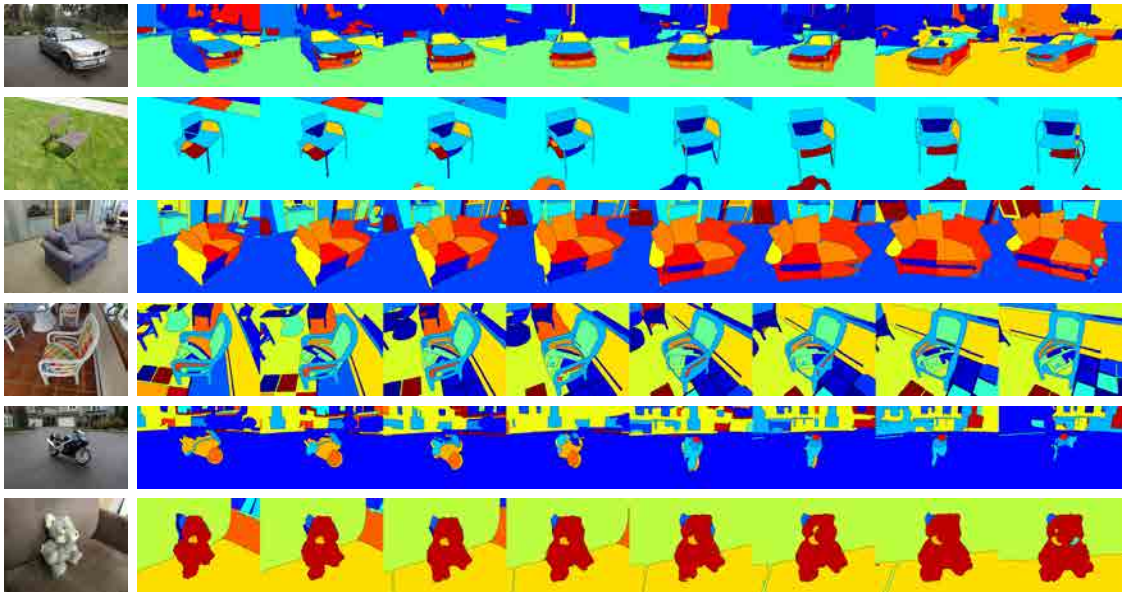


Figure 9.5: Qualitative assessment for generic co-clustering applied to BMW, Chair, Couch, GardenChair, Motorbike and Teddy datasets[KSS12]. First column: original images. Other columns: results for co-clustering where regions with same color represent the clusters obtained.

9.3 Qualitative assessment

We present some visual results obtained for the previous six datasets. The results of applying the generic two-step iterative co-clustering (I-2S) are shown in Figure 9.5 for each dataset. An example of semantic co-clustering in comparison with the generic co-clustering is also presented in Figure 9.6. Figures 9.7-9.12 provides more detailed results also showing the intermediate results (from left to right: original image, initial partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by a semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS)). The resolution has been selected using the automatic resolution selection technique presented in Section 8.3.5.

Furthermore, visual results for the datasets *Ballet* and *Breakdancers* [ZKU+04] where no ground truth is available are shown in Figure 9.13. In these examples, we can see that one limitation of the co-clustering is the quality of initial partitions on which the co-clustering is applied. For instance, a generic segmentation algorithm may fail in segmenting the right hand of the ballet dancer from the handrail. For such images, we consider that the hierarchical segmentation algorithm could also leverage the semantic information to create a hierarchy that respects the semantics. On the other hand, for images like breakdancers where several objects of the same semantic label are together, CNN techniques that tackle both semantic segmentation and object segmentation would be required.

Finally, for the sake of completeness, we are presenting illustrative results of the proposed technique in a video segmentation framework, more specifically in the context of scenes with small variations, as those contained in the Video Occlusion/Object Boundary

Detection Dataset [GVB11]. In that case, some of the objects represented in the video database did not correspond with the same semantic categories defined in the PASCAL Visual Object Challenge (SegVOC12) [EVGW+10] and, therefore, we do not have reliable semantic information. Thus, we are only presenting qualitative results on this database for the generic global co-clustering (I-2S+GG) presented in Section 8.3.4 (see Figure 9.14).

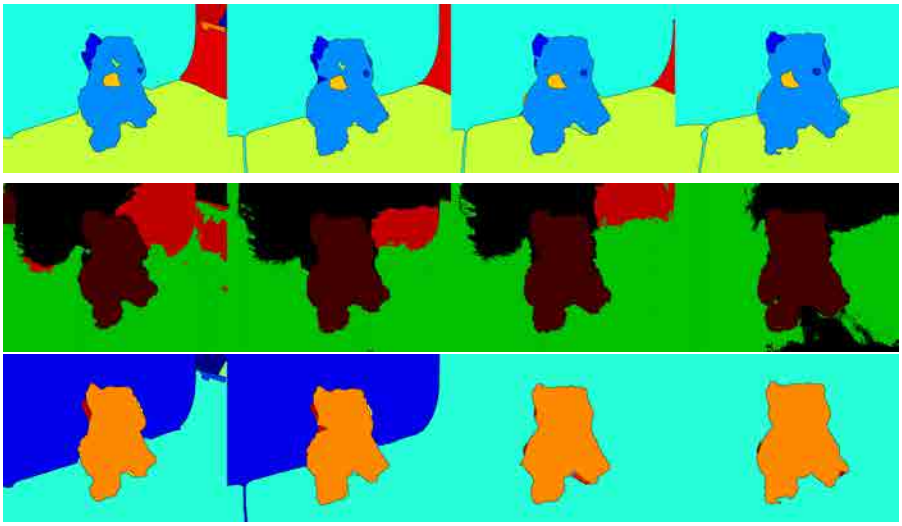


Figure 9.6: Qualitative assessment for semantic co-clustering in Teddy. First row: generic co-clustering. Second row: semantic segmentation from [ZJRP+15]. Third row: semantic co-clustering.

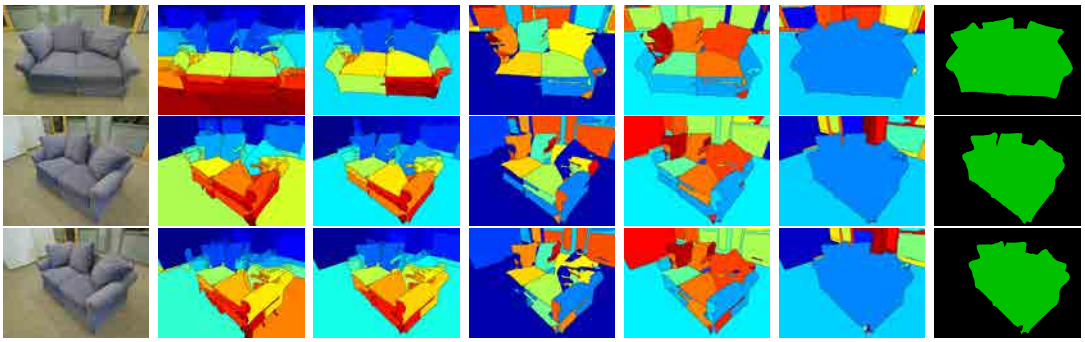


Figure 9.7: Sequence *Couch* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

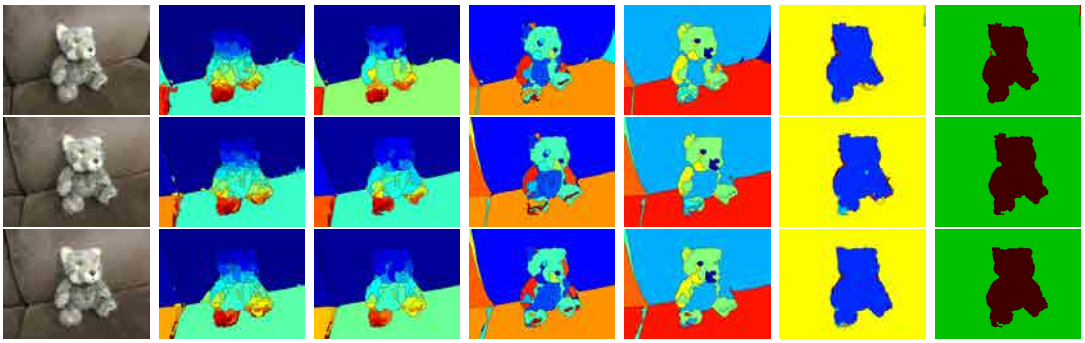


Figure 9.8: Sequence *Teddy* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

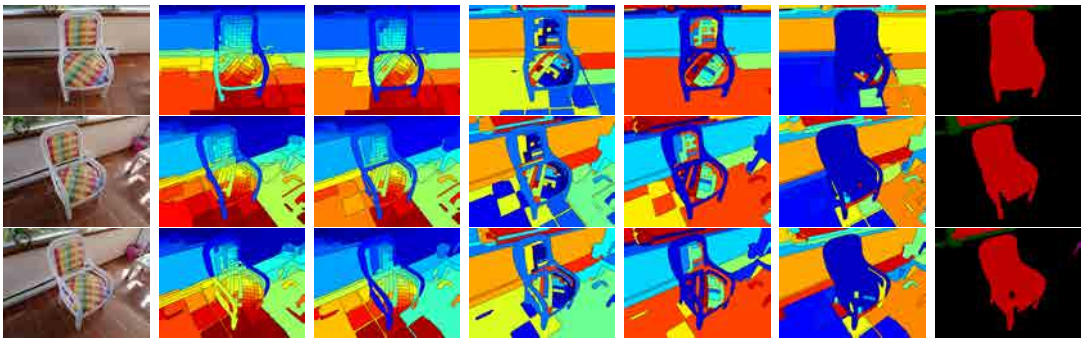


Figure 9.9: Sequence *GardenChair* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

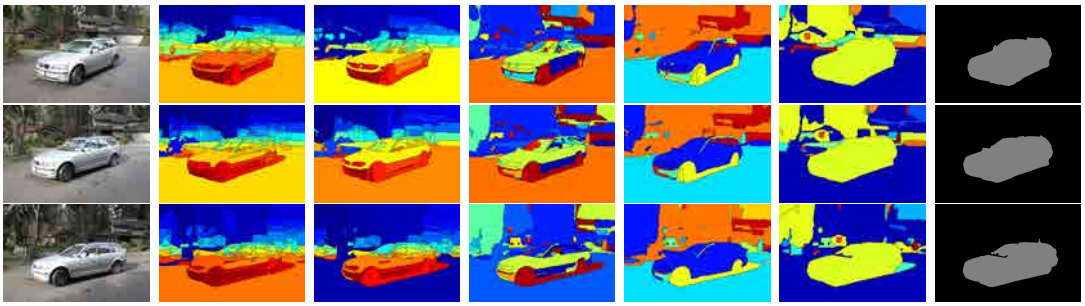


Figure 9.10: Sequence *BMW* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

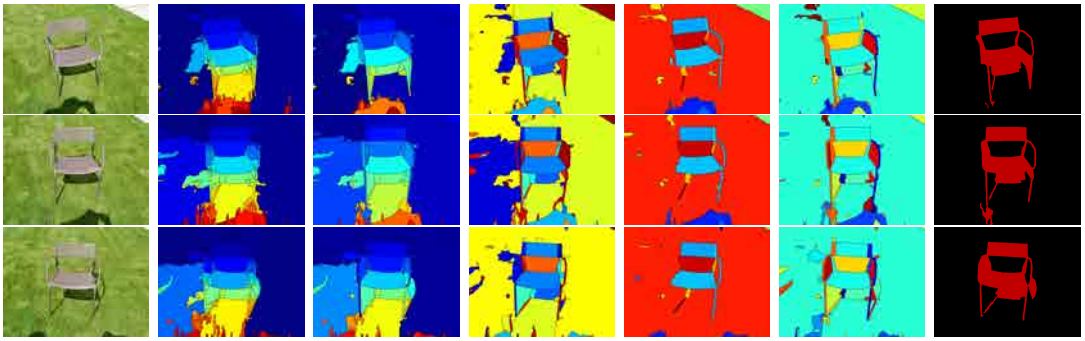


Figure 9.11: Sequence *Chair* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

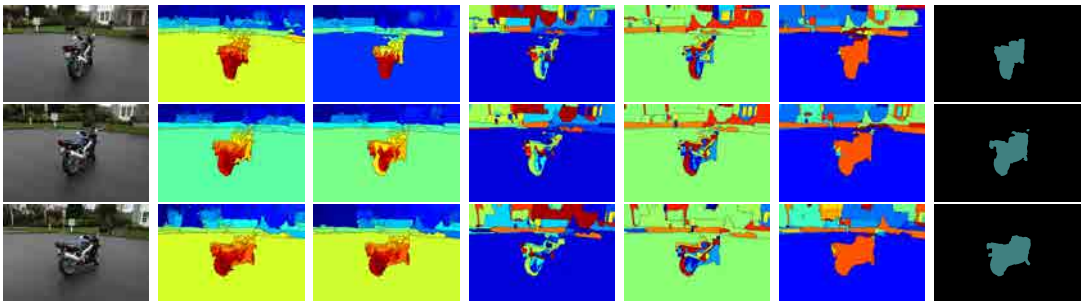


Figure 9.12: Sequence *Motorbike* [KSS12]. From left to right: original image, leaves partition, intra two-step iterative co-clustering (I-2S), inter two-step iterative co-clustering (I-2S), I-2S followed by a generic global optimization (I-2S+GG), I-2S followed by semantic global optimization (I-2S+SG) and semantic co-clustering based semantic segmentations (SCSS).

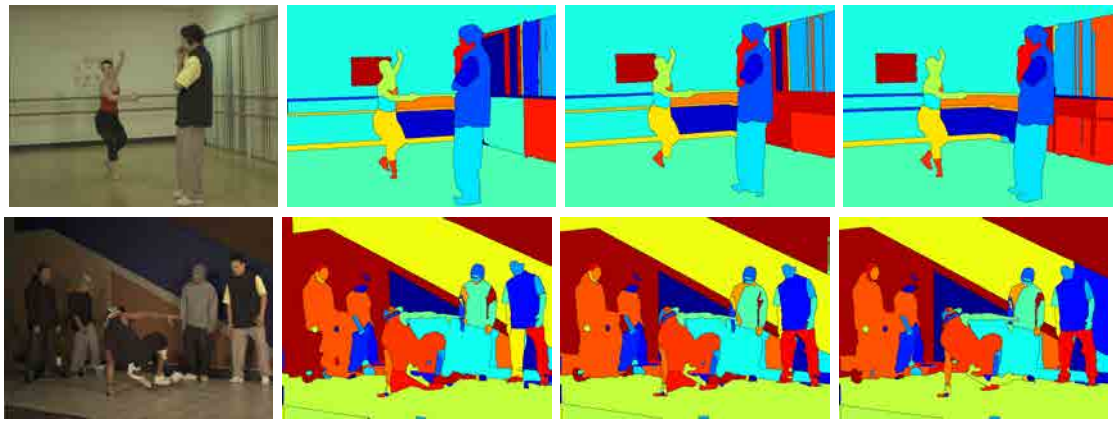


Figure 9.13: Qualitative assessment for generic co-clustering applied to ballet and breakdancers datasets[ZKU+04]

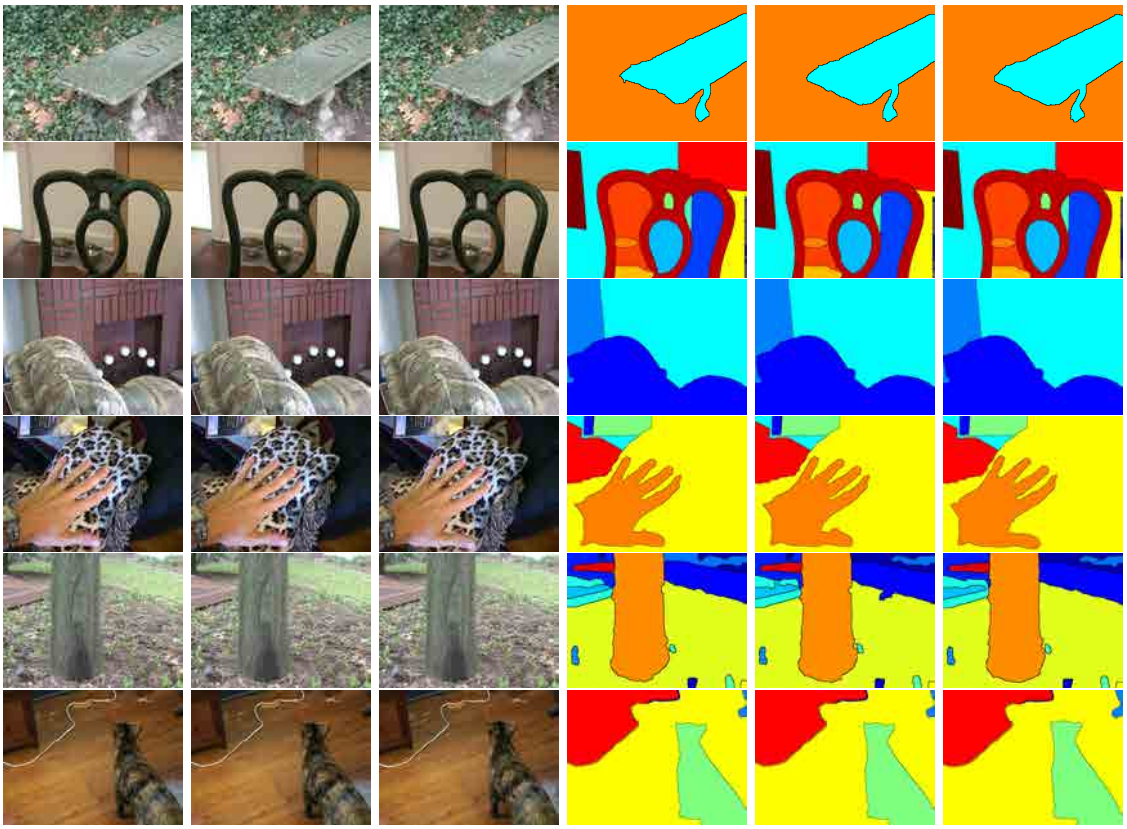


Figure 9.14: Illustrative results on the Occlusion/Object Boundary Detection Dataset [GVB11]

Conclusions and Future Work

10

In this work, a multiscale co-clustering framework for uncalibrated multiviews is proposed. Based on this framework, a generic two-pass co-clustering is presented to overcome the limitations imposed by the use of hierarchies in previous approaches. On top of this two-pass iterative algorithm, a global optimization process which exploits semantic information is described, having as a result a system where generic co-clustering and semantic segmentation benefits one from the other. Finally, an unsupervised scale selection technique that automatically obtains a single coherent labeling of the whole set of views has been presented. This part of the dissertation has been submitted to the European Conference on Computer Vision 2016 [VVGiN+16].

As future work, one of the ideas is using more powerful cues for intra similarity instead of color histograms and common boundary length. For instance, one possible approach could consist in taking advantage of UCMs, whose regions have been generated also considering texture and spectral clustering, to compute the intra similarity between regions of the same partition. When semantic information is considered, another future work line could be integrating such information as a cues in preliminary steps of the co-clustering, for instance to be used as both intra and inter similarity cues. Furthermore, such semantic information could be considered to build better initial hierarchies before applying the co-clustering techniques.

In another direction, more close to [KSS12], we would like to extend co-clustering techniques to calibrated scenarios, where the implicit geometry conveyed by the calibration information could be useful to compute a more accurate motion estimation by geometrically constraining the possible solutions.

Publications

11

The following publications are directly related with the content of this Thesis:

- C. Ventura, D. Varas, X. Giró-i-Nieto, V. Vilaplana, F. Marqués. Semantically driven multiresolution co-clustering for uncalibrated multiview segmentation. Submitted to the European Conference on Computer Vision (ECCV) 2016. In process of review.
- C. Ventura, X. Giró-i-Nieto, V. Vilaplana, K. McGuinness, F. Marqués, Noel E O'Connor. Improving spatial codification in semantic segmentation. Conference on Image Processing (ICIP) 2015.
- C. Ventura. Visual object analysis using regions and interest points. ACM international conference on Multimedia 2013.

Furthermore, work resulting from the collaboration with other researchers and with a content which is not directly related with this Thesis has also been published:

- K. McGuinness, E. Mohedano, Z. Zhang, F. Hu, R. Albatal, Cathal Gurrin, N. E O'Connor, A. F. Smeaton, A. Salvador, X. Giró-i-Nieto, C. Ventura. Insight Centre for Data Analytics (DCU) at TRECVid 2014: instance search and semantic indexing tasks. TRECVID Workshop 2014.
- C. Ventura, V. Vilaplana, X. Giró-i-Nieto, F. Marqués. Improving retrieval accuracy of Hierarchical Cellular Trees for generic metric spaces. Multimedia Tools and Applications, 2014.
- C. Ventura, X. Giró-i-Nieto, V. Vilaplana, D. Giribet, E. Carasusan. Automatic keyframe selection based on mutual reinforcement algorithm. International Workshop on Content-Based Multimedia Indexing (CBMI) 2013.
- C. Ventura, M. Tella-Amo, X. Giró-i-Nieto. UPC at MediaEval 2013 Hyperlinking Task. MediaEval 2013.
- C. Ventura, M. Martos, X. Giró-i-Nieto, V. Vilaplana, F. Marqués. Hierarchical navigation and visual search for video keyframe retrieval. International Conference on Multimedia Modeling 2012.

Bibliography

[AHG+12] P. Arbeláez, B. Hariharan, C. Gu, S. Gupta, L. Bourdev, and J. Malik, *Semantic segmentation using regions and parts*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 3378–3385. 15, 25, 26

[AMFM11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, *Contour detection and hierarchical image segmentation*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **33** (2011), no. 5, 898–916. 4, 5, 57, 58, 64, 65, 92, 93

[APTB+14] P. Arbeláez, J. Pont-Tuset, J. T. Barron, F. Marques, and J. Malik, *Multiscale combinatorial grouping*, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014. 5, 6, 16, 25, 26, 29, 45, 79, 81, 87

[ASS+12] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, *Slic superpixels compared to state-of-the-art superpixel methods*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012), no. 11, 2274–2282. 64

[BBM09] T. Brox, C. Bregler, and J. Malik, *Large displacement optical flow*, Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on, June 2009, pp. 41–48. 79, 93

[Bha46] A. Bhattacharyya, *On a measure of divergence between two multinomial populations*, Sankhyā: The Indian Journal of Statistics (1946), 401–406. 76

[BM92] S. Beucher and F. Meyer, *The morphological approach to segmentation: the watershed transformation*, OPTICAL ENGINEERING-NEW YORK-MARCEL DEKKER INCORPORATED- **34** (1992), 433–433. 4

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih, *Fast approximate energy minimization via graph cuts*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **23** (2001), no. 11, 1222–1239. 20

[CCBS12] J. Carreira, R. Caseiro, J. Batista, and C. Sminchisescu, *Semantic segmentation with second-order pooling*, Computer Vision ECCV 2012 (A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, and C. Schmid, eds.), Lecture Notes in Computer Science, vol. 7578, Springer Berlin Heidelberg, 2012, pp. 430–443 (English). 7, 8, 10, 11, 15, 16, 17, 22, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 36, 37, 39, 41, 42, 43, 44, 45, 48, 49

[CGW] M. Charikar, V. Guruswami, and A. Wirth, *Clustering with qualitative information*, Foundations of Computer Science, 2003., pp. 524–533. 59, 72

[CH07] D. J. Crandall and D. P. Huttenlocher, *Composite models of objects and scenes for category recognition*, Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on, IEEE, 2007, pp. 1–8. 16

[CLS12] J. Carreira, F. Li, and C. Sminchisescu, *Object recognition by sequential figure-ground ranking*, International journal of computer vision **98** (2012), no. 3, 243–262. 7, 15, 22, 37, 45

[CS12] J. Carreira and C. Sminchisescu, *Cpmc: Automatic object segmentation using constrained parametric min-cuts*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **34** (2012), no. 7, 1312–1328. 5, 6, 16, 25, 26, 29, 31

[DCYY14] J. Dong, Q. Chen, S. Yan, and A. Yuille, *Towards unified object detection and semantic segmentation*, Computer Vision–ECCV 2014, Springer, 2014, pp. 299–314. 16, 39

[DDS+09] J. Deng, W. Dong, R. Socher, L-J. Li, K. Li, and L. Fei-Fei, *Imagenet: A large-scale hierarchical image database*, Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on, IEEE, 2009, pp. 248–255. 31, 55

[DFB+13] A. Djelouah, J-S. Franco, E. Boyer, F. Le Clerc, and P. Perez, *Multi-view object segmentation in space and time*, Computer Vision (ICCV), 2013 IEEE International Conference on, IEEE, 2013, pp. 2640–2647. 56

[DHH+09] S. K. Divvala, D. Hoiem, J. H. Hays, A. Efros, and M. Hebert, *An empirical study of context in object detection*, Computer Vision and Pattern Recognition (CVPR), 2009 IEEE Conference on, IEEE, 2009, pp. 1271–1278. 15

[DHS16] J. Dai, K. He, and J. Sun, *Instance-aware Semantic Segmentation via Multi-task Network Cascades*, Computer Vision and Pattern Recognition (CVPR), 2016 IEEE Conference on (2016). 51

[DT05] N. Dalal and B. Triggs, *Histograms of oriented gradients for human detection*, Computer Vision and Pattern Recognition (CVPR), 2005 IEEE Conference on, vol. 1, IEEE, 2005, pp. 886–893. 8, 15, 19, 20, 26

[EVGW+] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, *The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results*, http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 25, 55

[EVGW+10] ———, *The pascal visual object classes (voc) challenge*, International Journal of Computer Vision **88** (2010), no. 2, 303–338. 15, 21, 29, 30, 86, 99

[EZWVG06] M. Everingham, A. Zisserman, C. K. Williams, and L. Van Gool, *The pascal visual object classes challenge 2006 (voc2006) results*, 2006. 20

[FCNL13] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, *Learning hierarchical features for scene labeling*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **35** (2013), no. 8, 1915–1929. 51, 55

[FGMR10] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, *Object detection with discriminatively trained part-based models*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010), no. 9, 1627–1645. 15, 19, 20, 26

[GAL+12] C. Gu, P. Arbeláez, Y. Lin, K. Yu, and J. Malik, *Multi-component models for object detection*, Computer Vision–ECCV 2012, Springer, 2012, pp. 445–458. 25, 26

[GCS13] F. Galasso, R. Cipolla, and B. Schiele, *Video segmentation with superpixels*, Computer Vision–ACCV 2012, Springer, 2013, pp. 760–774. 56, 57, 93, 94, 97

[GDDM14] R. Girshick, J. Donahue, T. Darrell, and J. Malik, *Rich feature hierarchies for accurate object detection and semantic segmentation*, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014. 15

[GKBS14] F. Galasso, M. Keuper, T. Brox, and B. Schiele, *Spectral graph reduction for efficient image and streaming video segmentation*, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 49–56. 76

[GKHE10] M. Grundmann, V. Kwatra, M. Han, and I. Essa, *Efficient hierarchical graph-based video segmentation*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 2141–2148. 56, 57, 76, 87, 93, 94, 97

[GSNJC+13] F. Galasso, N. Shankar-Nagaraja, T. Jimenez-Cardenas, T., and B. Schiele, *A unified video segmentation benchmark: Annotation, metrics and analysis*, Computer Vision (ICCV), 2013 IEEE International Conference on, 2013. 96

[GVB11] D. Glasner, S.N. Vitaladevuni, and R. Basri, *Contour-based joint clustering of multiple segmentations*, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, 2011. 56, 59, 63, 64, 71, 72, 73, 75, 76, 78, 79, 80, 91, 93, 99, 102

[HAB+11] B. Hariharan, P. Arbeláez, L. Bourdev, S. Maji, and J. Malik, *Semantic contours from inverse detectors*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 991–998. 30

[HAGM14] B. Hariharan, P. Arbeláez, R. Girshick, and J. Malik, *Simultaneous detection and segmentation*, Computer Vision–ECCV 2014, Springer, 2014, pp. 297–312. 15, 39, 55

[HJS09] H. Harzallah, F. Jurie, and C. Schmid, *Combining efficient object localization and image classification*, Computer Vision (ICCV), 2009 IEEE International Conference on, IEEE, 2009, pp. 237–244. 15, 19, 20, 26

[JBP12] A. Joulin, F. Bach, and J. Ponce, *Multi-class cosegmentation*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 542–549. 56, 58, 93, 94, 97

[KLH12] E. Kim, H. Li, and X. Huang, *A hierarchical image clustering cosegmentation framework*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, June 2012, pp. 686–693. 58, 84

[KSS12] A. Kowdle, S. N. Sinha, and R. Szeliski, *Multiple view object cosegmentation using appearance and stereo cues*, Computer Vision–ECCV 2012, Springer, 2012, pp. 789–803. 56, 91, 92, 94, 98, 99, 100, 101, 103

[KX12] G. Kim and E. P. Xing, *On multiple foreground cosegmentation*, Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE, 2012, pp. 837–844. 56, 58, 93, 94, 97

[LCLS13] F. Li, J. Carreira, G. Lebanon, and C. Sminchisescu, *Composite statistical inference for semantic segmentation*, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 3302–3309. 16, 37

[LCS10] F. Li, J. Carreira, and C. Sminchisescu, *Object recognition as ranking holistic figure-ground hypotheses*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1712–1719. 16

[LMB+14] T.Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, *Microsoft COCO: Common objects in context*, Computer Vision–ECCV 2014, 2014. 15, 31, 51, 55

[Low04] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, International journal of computer vision **60** (2004), no. 2, 91–110. 7, 8

[LSD15] J. Long, E. Shelhamer, and T. Darrell, *Fully convolutional networks for semantic segmentation*, Computer Vision and Pattern Recognition (CVPR), 2015 IEEE Conference on, 2015, pp. 3431–3440. 55

[LSP06] S. Lazebnik, C. Schmid, and J. Ponce, *Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories*, Computer Vision and Pattern Recognition (CVPR), 2006 IEEE Conference on, vol. 2, IEEE, 2006, pp. 2169–2178. 10, 11, 25

[LXL+15] Z. Liu, L. Xiaoxiao, P. Luo, C. C. Loy, and X. Tang, *Semantic image segmentation via deep parsing network*, Computer Vision (ICCV), 2015 IEEE International Conference on (2015). 51

[ME07] T. Malisiewicz and A. A. Efros, *Improving spatial support for objects via multiple segmentations*. 21, 22

[MFTM01] D. Martin, C. Fowlkes, D. Tal, and J. Malik, *A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics*, Computer Vision (ICCV), 2001 IEEE International Conference on, vol. 2, IEEE, 2001, pp. 416–423. 4

[OPM02] T. Ojala, M. Pietikainen, and T. Maenpaa, *Multiresolution gray-scale and rotation invariant texture classification with local binary patterns*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **24** (2002), no. 7, 971–987. 7, 9

[PCMY15] G. Papandreou, L-C. Chen, K. Murphy, and A. L. Yuille, *Weakly- and semi-supervised learning of a dcnn for semantic image segmentation*, Computer Vision (ICCV), 2015 IEEE International Conference on (2015). 51

[PTG15] J. Pont-Tuset and L. Van Gool, *Boosting object proposals: From pascal to coco*, Computer Vision (ICCV), 2015 IEEE International Conference on, 2015. 55

[PTM12] J. Pont-Tuset and F. Marques, *Supervised assessment of segmentation hierarchies*, Computer Vision–ECCV 2012, Springer, 2012, pp. 814–827. 92

[Ram07] D. Ramanan, *Using segmentation to verify object hypotheses*, Computer Vision and Pattern Recognition (CVPR), 2007 IEEE Conference on, IEEE, 2007, pp. 1–8. 20, 21

[RLYFF12] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei, *Object-centric spatial pooling for image classification*, Computer Vision–ECCV 2012, Springer, 2012, pp. 1–15. 15, 16

[RTMF08] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman, *Labelme: a database and web-based tool for image annotation*, International journal of computer vision **77** (2008), no. 1-3, 157–173. 20

[RVG+07] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie, *Objects in context*, Computer Vision (ICCV), 2007 IEEE International Conference on, IEEE, 2007, pp. 1–8. 22

[SF68] I. Sobel and G. Feldman, *A 3x3 isotropic gradient operator for image processing*, a talk at the Stanford Artificial Project in (1968), 271–272. 65

[SG00] P. Salembier and L. Garrido, *Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval*, Image Processing, IEEE Transactions on **9** (2000), no. 4, 561–576. 3, 60

[SZ03] J. Sivic and A. Zisserman, *Video google: A text retrieval approach to object matching in videos*, Computer Vision (ICCV), 2003 IEEE International Conference on, IEEE, 2003, pp. 1470–1477. 9, 10

[TE11] A. Torralba and A.A. Efros, *Unbiased look at dataset bias*, Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, June 2011, pp. 1521–1528. 55

[TKSMN13] E. Trulls, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, *Dense segmentation-aware descriptors*, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 2890–2897. 22, 23, 51

[TLF10] E. Tola, V. Lepetit, and P. Fua, *Daisy: An efficient dense descriptor applied to wide-baseline stereo*, Pattern Analysis and Machine Intelligence, IEEE Transactions on **32** (2010), no. 5, 815–830. 22, 23, 51

[TTK+14] E. Trulls, S. Tsogkas, I. Kokkinos, A. Sanfeliu, and F. Moreno-Noguer, *Segmentation-aware deformable part models*, Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on, 2014, pp. 168–175. 20

[USS12] J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha, *The visual extent of an object*, International Journal of Computer Vision **96** (2012), no. 1, 46–63 (English). 5, 15, 16, 22, 24, 25, 26, 51

[VAM15] D. Varas, M. Alfaro, and F. Marques, *Multiresolution hierarchy co-clustering for semantic segmentation in sequences with small variations*, Computer Vision (ICCV), 2015 IEEE International Conference on (2015). 56, 57, 58, 59, 63, 65, 72, 73, 76, 77, 78, 79, 80, 81, 82, 91, 93

[VB10] S.N. Vitaladevuni and R. Basri, *Co-clustering of image segments using convex optimization applied to em neuronal reconstruction*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, 2010. 56, 59, 63, 72, 78, 79

[VdBBR+12] M. Van den Bergh, X. Boix, G. Roig, B. de Capitani, and L. Van Gool, *Seeds: Superpixels extracted via energy-driven sampling*, Computer Vision–ECCV 2012, Springer, 2012, pp. 13–26. 64

[vdSUGS11] K. EA van de Sande, J. RR Uijlings, T. Gevers, and A. WM Smeulders, *Segmentation as selective search for object recognition*, Computer Vision (ICCV), 2011 IEEE International Conference on, IEEE, 2011, pp. 1879–1886. 20, 21

[VGiNV+15] C. Ventura, X. Giró-i Nieto, V. Vilaplana, K. McGuinness, F. Marqués, and N. E. O'Connor, *Improving spatial codification in semantic segmentation*, Image Processing (ICIP), 2015 IEEE International Conference on, IEEE, 2015, pp. 3605–3609. 15, 51

[VGNV+] C. Ventura, X. Giró-Nieto, V. Vilaplana, K. McGuinness, F. Marquesl, and N. O'Connor, *Improving spatial codification in semantic segmentation (supplementary material)*, https://imatge.upc.edu/web/sites/default/files/pub/cVentura.pdf, Accessed: 2016-03-29. 43, 45

[VJ01] P. Viola and M. Jones, *Rapid object detection using a boosted cascade of simple features*, Computer Vision and Pattern Recognition (CVPR), 2001 IEEE Conference on, vol. 1, IEEE, 2001, pp. I–511. 19, 20

[VVGiN+16] C. Ventura, D. Varas, X. Giró-i Nieto, V. Vilaplana, and F. Marqués, *Semantically driven multiresolution co-clustering for uncalibrated multiview segmentation*, Submitted to Computer Vision–ECCV 2016 (2016). 103

[XSF+12] W. Xia, Z. Song, J. Feng, L. Cheong, and S. Yan, *Segmentation over detection by coupled global and local sparse representations*, Computer Vision–ECCV 2012, Springer, 2012, pp. 662–675. 16, 37

[XXC12] C. Xu, C. Xiong, and J. J. Corso, *Streaming hierarchical video segmentation*, Computer Vision–ECCV 2012, Springer, 2012, pp. 626–639. 56, 57, 79, 84, 92, 93, 94, 97

[YBS13] P. Yadollahpour, D. Batra, and G. Shakhnarovich, *Discriminative re-ranking of diverse segmentations*, Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on, IEEE, 2013, pp. 1923–1930. 15, 37, 39

[ZCYF10] L. Zhu, Y. Chen, A. Yuille, and W. Freeman, *Latent hierarchical structural learning for object detection*, Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, IEEE, 2010, pp. 1062–1069. 19, 20

[ZJRP+15] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, *Conditional random fields as recurrent neural networks*, Computer Vision (ICCV), 2015 IEEE International Conference on (2015). 51, 55, 56, 86, 87, 88, 95, 96, 99

[ZKU+04] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, *High-quality video view interpolation using a layered representation*, ACM Transactions on Graphics (TOG), vol. 23, ACM, 2004, pp. 600–608. 93, 98, 101

[ZMLS07] J. Zhang, M. Marszałek, S. Lazebnik, and C. Schmid, *Local features and kernels for classification of texture and object categories: A comprehensive study*, International journal of computer vision **73** (2007), no. 2, 213–238. 21, 24