UNIVERSITAT POLITÈCNICA DE CATALUNYA

Ph.D. Thesis

Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences

Jaime Gallego Vila

Thesis supervisor:

Dra. Montse Pardàs Feliu

Department of Signal Theory and Communications
Universitat Politècnica de Catalunya, UPC

Barcelona, July 2013

Abstract

Foreground segmentation in video sequences is an important area of the image processing that attracts great interest among the scientist community, since it makes possible the detection of the objects that appear in the sequences under analysis, and allows us to achieve a correct performance of high level applications which use foreground segmentation as an initial step.

The current Ph.D. thesis entitled *Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences* details, in the following pages, the research work carried out within this field. In this investigation, we propose to use parametric probabilistic models at pixel-wise and region level in order to model the different classes that are involved in the classification process of the different regions of the image: foreground, background and, in some sequences, shadow. The development is presented in the following chapters as a generalization of the techniques proposed for objects segmentation in 2D planar sequences to 3D multi-view environment, where we establish a cooperative relationship between all the sensors that are recording the scene.

Hence, different scenarios have been analyzed in this thesis in order to improve the foreground segmentation techniques:

In the first part of this research, we present segmentation methods appropriate for 2D planar scenarios. We start dealing with foreground segmentation in static camera sequences, where a system that combines pixel-wise background model with region-based foreground and shadow models is proposed in a Bayesian classification framework. The research continues with the application of this method to moving camera scenarios, where the Bayesian framework is developed between foreground and background classes, both characterized with region-based models, in order to obtain a robust foreground segmentation for this kind of sequences.

The second stage of the research is devoted to apply these 2D techniques to multi-view acquisition setups, where several cameras are recording the scene at the same time. At the beginning of this section, we propose a foreground segmentation system for sequences recorded by means of color and depth sensors, which combines

different probabilistic models created for the background and foreground classes in each one of the views, by taking into account the reliability that each sensor presents. The investigation goes ahead by proposing foreground segregation methods for multi-view smart room scenarios. In these sections, we design two systems where foreground segmentation and 3D reconstruction are combined in order to improve the results of each process. The proposals end with the presentation of a multi-view segmentation system where a foreground probabilistic model is proposed in the 3D space to gather all the object information that appears in the views.

The results presented in each one of the proposals show that the foreground segmentation and also the 3D reconstruction can be improved, in these scenarios, by using parametric probabilistic models for modeling the objects to segment, thus introducing the information of the object in a Bayesian classification framework.

Resumen

La segmentación de objetos de primer plano en secuencias de vídeo es una importante área del procesado de imagen que despierta gran interés por parte de la comunidad científica, ya que posibilita la detección de objetos que aparecen en las diferentes secuencias en análisis, y permite el buen funcionamiento de aplicaciones de alto nivel que utilizan esta segmentación obtenida como parámetro de entrada. La presente tesis doctoral titulada *Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences* detalla, en las páginas que siguen, el trabajo de investigación desarrollado en este campo. En esta investigación se propone utilizar modelos probabilísticos paramétricos a nivel de píxel y a nivel de región para modelar las diferentes clases que participan en la clasificación de las regiones de la imagen: primer plano, fondo y en según que secuencias, las regiones de sombra. El desarrollo se presenta en los capítulos que siguen como una generalización de técnicas propuestas para la segmentación de objetos en secuencias 2D mono-cámara, al entorno 3D multi-cámara, donde se establece la cooperación de los diferentes sensores que participan en la grabación de la escena.

De esta manera, diferentes escenarios han sido estudiados con el objetivo de mejorar las técnicas de segmentación para cada uno de ellos: En la primera parte de la investigación, se presentan métodos de segmentación para escenarios mono-cámara. Concretamente, se comienza tratando la segmentación de primer plano para cámara estática, donde se propone un sistema completo basado en la clasificación Bayesiana entre el modelo a nivel de píxel definido para modelar el fondo, y los modelos a nivel de región creados para modelar los objetos de primer plano y la sombra que cada uno de ellos proyecta. La investigación prosigue con la aplicación de este método a secuencias grabadas mediante cámara en movimiento, donde la clasificación Bayesiana se plantea entre las clases de fondo y primer plano, ambas caracterizadas con modelos a nivel de región, con el objetivo de obtener una segmentación robusta para este tipo de secuencias.

La segunda parte de la investigación, se centra en la aplicación de estas técnicas mono-cámara a entornos multi-vista, donde varias cámaras graban conjuntamente la misma escena. Al inicio de dicho apartado, se propone una segmentación de primer

plano en secuencias donde se combina una cámara de color con una cámara de profundidad en una clasificación que combina los diferentes modelos probabilísticos creados para el fondo y el primer plano en cada cámara, a partir de la fiabilidad que presenta cada sensor. La investigación prosigue proponiendo métodos de segmentación de primer plano para entornos multi-vista en salas inteligentes. En estos apartados se diseñan dos sistemas donde la segmentación de primer plano y la reconstrucción 3D se combinan para mejorar los resultados de cada uno de estos procesos. Las propuestas finalizan con la presentación de un sistema de segmentación multicámara donde se centraliza la información del objeto a segmentar mediante el diseño de un modelo probabilístico 3D.

Los resultados presentados en cada uno de los sistemas, demuestran que la segmentación de primer plano y la reconstrucción 3D pueden verse mejorados en estos escenarios mediante el uso de modelos probabilisticos paramétricos para modelar los objetos a segmentar, introduciendo así la información disponible del objeto en un marco de clasificación Bayesiano.

Resum

La segmentació d'objectes de primer pla en seqüències de vídeo és una important àrea del processat d'imatge que acull gran interès per part de la comunitat científica, ja que possibilita la detecció d'objectes que apareixen en les diferents seqüències en anàlisi, i permet el bon funcionament d'aplicacions d'alt nivell que utilitzen aquesta segmentació obtinguda com a paràmetre d'entrada. Aquesta tesi doctoral titulada *Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences* detalla, en les pàgines que segueixen, el treball de recerca desenvolupat en aquest camp. En aquesta investigació es proposa utilitzar models probabilístics paramètrics a nivell de píxel i a nivell de regió per modelar les diferents classes que participen en la classificació de les regions de la imatge: primer pla, fons i depenent de les seqüències, les regions d'ombra. El desenvolupament es presenta als capítols que segueixen com una generalització de tècniques proposades per a la segmentació d'objectes en seqüències 2D mono-càmera, a l'entorn 3D multicàmera, on s'estableix la cooperació dels diferents sensors que participen en l'enregistrament de l'escena .

D'aquesta manera, s'han estudiat diferents escenaris amb l'objectiu de millorar les tècniques de segmentació per a cadascun d'ells: A la primera part de la investigació, es presenten mètodes de segmentació per escenaris mono-càmera. Concretament, es comença tractant la segmentació de primer pla per a càmera estàtica, on es proposa un sistema basat en la classificació Bayesiana entre el model a nivell de píxel per modelar el fons, i els models a nivell de regió creats per modelar els objectes de primer pla i l'ombra que cada un d'ells projecta. La investigació continua amb l'aplicació d'aquest mètode a seqüències gravades mitjançant càmera en moviment, on la classificació Bayesiana es planteja entre les classes de fons i primer pla, ambdues caracteritzades amb models a nivell de regió, amb l'objectiu d'obtenir una segmentació robusta per aquest tipus de seqüències.

La segona part de la investigació, es focalitza en l'aplicació d'aquestes tècniques mono-càmera a entorns multi-vista, on diverses càmeres graven conjuntament la mateixa escena. A l'inici d'aquest apartat, es proposa una segmentació de primer pla en seqüències on es combina una càmera de color amb una càmera de profunditat en una classificació que combina els diferents models probabilístics creats per al fons

i el primer pla a cada càmera, a partir de la fiabilitat que presenta cada sensor. La investigació continua proposant mètodes de segmentació de primer pla per a entorns multi-vista en sales intel·ligents. En aquests apartats es dissenyen dos sistemes on la segmentació de primer pla i la reconstrucció 3D es combinen per millorar els resultats de cada un d'aquests processos. Les propostes finalitzen amb la presentació d'un sistema de segmentació multicàmera on es centralitza la informació de l'objecte a segmentar mitjançant el disseny d'un model probabilístic 3D.

Els resultats presentats en cada un dels sistemes, demostren que la segmentació de primer pla i la reconstrucció 3D es poden veure millorats en aquests escenaris mitjançant l'ús de models probabilístics paramètrics per modelar els objectes a segmentar, introduint així la informació disponible de l'objecte en un marc de classificació Bayesià.

Agradecimientos

Quisiera empezar dando las gracias a Montse Pardás por dirigir y supervisar este trabajo de cuatro años de manera atenta y dedicada, con amabilidad y comprensión en todo momento.

Gracias a Gloria Haro por su ayuda en el desarrollo del Capítulo 4 y a Jordi Salvador por su colaboración en el Capítulo 8.

También agradecer a los demás profesores del grupo de procesado de imagen el soporte y el ambiente de trabajo tan enriquecedor del grupo: Ferrán Marqués, Philippe Salembier, Javier Ruiz, Ramon Morros, Josep R. Casas.

Gracias a todos los compañeros con los que he compartido buenos momentos de trabajo, viajes, conversaciones, cafés en el bar y mañanas de pastel casero: Omid, Cristian, Jordi Salvador, Jordi Pont, David Bernal, David Matas, David Varas, Marcel, Adolfo, Albert, Taras, Carlos, Marc, Guillem.

Finalmente, quiero dar las gracias de manera muy especial a Elena por su soporte incondicional y a mi familia por estar cerca de mí en todo momento.

A todos vosotros,

Gracias

Contents

List of Figures

List of Notations

c Color RGB domain

G$(\cdot)$ Gaussian likelihood

$\mathbf{s_{3D}}$ 3D Spatial XYZ domain

SfS Shape from Silhouette

s 2D Spatial XY domain

v $RGB\ XYZ$

x $RGB\ XY\ Z$ domain, where Z stands for the depth from the camera sensor

z $RGB\ XY$ domain

bg Background

EM Expectation Maximization algorithm

fg Foreground

GMM Gaussian Mixture Model

MAP Maximum a Posteriori

MRF Markov Random Field

pdf Probability density function

SCGMM Spatial Color Gaussian Mixture Model

SCGM Spatial Color Gaussian Model

seg. Segmentation

sh Shadow

VH Visual Hull

Chapter 1

Introduction

Foreground segmentation is the field of the image processing area that gathers all the techniques used to achieve a correct separation of the foreground objects from the background, for a certain video sequence under analysis. It is a fundamental first processing stage for vision systems which monitor real-world activity, where the output depends completely or partially on the visualization of the segmentation. For instance, in videoconferencing once the foreground and the background are separated, the background can be replaced by another image, which then beautifies the video and protects the user privacy. The extracted foreground objects can be compressed to facilitate efficient transmission using object-based video coding. As an advanced video editing tool, segmentation also allows people to combine multiple objects from different video and create new artistic results. In video surveillance tasks, foreground segmentation allows a correct object identification and tracking, while in 3D multi-camera environments, robust foreground segmentation makes possible a correct 3-dimensional reconstruction without background artifacts. The current Thesis is defined in this framework: *Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences* with the main objective of developing foreground segmentation methods based on the probabilistic modeling of the foreground objects and the background regions, for both, planar and multi-view video sequences.

1.1 Motivation

Nowadays, the society is presenting an increasing use of technological devices that interact with the users in order to make easier common tasks that appear in our life. The challenge that present all these new tools is related to how these computer systems can interact better with humans, allowing an intuitive communication be-

tween both by means of the human communication channels: image, sound and touch, to correctly detect and identify what is happening in the environment and extract the semantic information of any situation. The necessity of the foreground segmentation area to extract the information of the images recorded by camera sensors is motivated in this context.

Foreground segmentation is a complex issue inside the image processing area which has received a great deal of attention during the lasts years, mainly fostered by the necessity of high level applications to detect, interpret and imitate humans' actions and the technical possibility to carry out new systems in real time processing. This area has suffered a great change since some decades ago, when the scientists started with this research, trying to segment persons and objects that move over static elements of the environment in order to achieve an automatic detection. The constant increasing of the computational capacity, the improvement of the color camera sensors, the appearance of new devices suitable for capturing the depth of the scene and the reduction of the price in all these technical components, have created this new context on the foreground segmentation area towards precise and real-time detections.

In front of this scenario, there is a new trend to improve the reliability of the computer vision systems based not only on improving the segmentation technique used for single camera scenarios, but also, and central to the current foreground segmentation systems, on developing new techniques to combine properly several camera sensors, in order to take advantage of the data redundancy and improve the final decision. Hence, to find scalable foreground segmentation techniques that could be applied not only on a single planar camera, but also on a combination of several camera sensors is currently a very interesting challenge in computer vision. In this way, we propose this thesis as a foreground segmentation research from 2D scenarios, where just one color camera sensor is recording the scene under analysis, to a 3D framework, where several camera sensors are synchronized to record the same scene from different positions. In the middle, we will analyze different type of scenarios like static and moving camera sequences, as well as the combination of color and depth sensor and multi-view scenarios.

In the following chapters we will explain how the parametric region-based probabilistic models, used and proposed in this thesis, allow us to design a Bayesian classification between classes for single and multi-sensor foreground segmentation framework.

1.2 Summary of contributions

- **Foreground segmentation for 2D planar scenarios:**
  - **Foreground segmentation for monocular static sequences using pixel-wise background model with region-based foreground and shadow models**
    We have developed a robust 2D foreground segmentation for monocular static cameras where foreground and shadow classes are modeled in a region based level to achieve non-rigid probabilistic modeling along the scene.
  - **Foreground segmentation for 2D moving camera sequences using region-based foreground and background models**
    A foreground segmentation system for moving camera scenarios is proposed in this contribution. The principles of this system are based on the method designed for static cameras, but applied to two region-based models defined to model the foreground and background classes.

- **Foreground segmentation in multi-view sequences:**
  - **Foreground segmentation in color-depth multi-sensor framework**
    This approach combines two camera sensors that work in the color RGB and depth Z domains. Specific models for each sensor to characterize the foreground and background are defined. The probabilities obtained from the models are combined via logarithmic opinion pool decision, weighting the probabilities according to the reliability maps that each sensor presents.
  - **Multi-view Foreground segmentation in smart-room scenarios**
    Smart-room environments present a characteristic that make them suitable for an overall multi-view analysis of the scene: All the cameras are recording the scene at the same time from different points of view. Hence we propose to exploit this spatial redundancy in order to improve the segmentation obtained in each view:
    - **Reliability maps applied to robust SfS volumetric reconstruction between foreground and background/shadow models**
      We compute the reliability maps of each sensor by means of the similarity that the foreground model presents with respect to the background and shadow models. We obtain this similarity measure by computing the Hellinger distance between models and we use it in order to achieve a robust SfS reconstruction.

- **Joint Multi-view Foreground Segmentation and 3D Reconstruction with Tolerance Loop**
  A loop between cooperative foreground segmentation and 3D reconstruction is proposed in this research line by updating the foreground model, defined in each view, with the conservative 3D volume reconstruction of the object. We exploit here the possibilities that the 3D reconstruction with tolerance to errors presents, in order to reduce the misses presented in the 2D foreground masks and the 3D volumetric reconstruction.
- **3D Foreground probabilistic model**
  Our last approach consists in the design of a more robust and complete foreground model designed in the 3D space. In this way, we propose this object modeling to be shared by all the views, and used for monocular 2D segmentation. With this approach, we try to establish a novel method to compute the multi-view smart-room segmentation.

A complete list of contributions of this thesis, as well as a list of publications and collaborations in the image processing group of the UPC have been compiled in the last chapter.

1.3 Thesis Outline

The manuscript is organized as follows: in the next **Chapter 2**, we state the problem that we want to address regarding the foreground segmentation, and its dependence on the scenario characteristics and the acquisition setup utilized to record the sequences. **Chapter 3** is devoted to review the state of the art of the foreground segmentation area, necessary to establish the background concepts required to develop the proposals presented in this thesis. **Part I** and **Part II** gather the chapters intended to present our proposals: **Part I** deals with foreground segmentation approaches for 2D planar scenarios, where **Chapter 4** focuses on a foreground segmentation system appropriate for static camera sequences, which combines pixel-wise background model with parametric region-based foreground and shadow models, and **Chapter 5** utilizes the principles of **Chapter 4** to establish a foreground segmentation framework suitable for moving camera sequences. **Part II** of this thesis is devoted to explain the proposals for multi-view scenarios. In this part, **Chapter 6** deals with sequences recorded by means of color and depth sensors to develop a foreground segregation system which combines, in a Logarithmic Opinion Pool framework, the information provided by each sensor to determine the final foreground segmentation mask. In **Chapter 7** we propose a collaborative fore-

ground segmentation and 3D reconstruction process which achieves a robust 3D reconstruction of the object by defining and including the reliability maps of each sensor in the 3D reconstruction. Multi-view Foreground Segmentation and 3D Reconstruction with Tolerance Loop is presented in **Chapter 8** introducing a method to improve the foreground models of the 2D views, by using the conservative 3D volume of the object to update the 2D foreground models, thus improving the subsequent volumetric reconstruction. In **Chapter 9** we present the 3D foreground model to develop a multi-view foreground segmentation by creating a foreground model in the 3D space, and utilize the projection of this model to the 2D views, to perform the planar foreground segmentation. Finally, the conclusions, list of publications and future lines of work are presented in **Chapter 10**.

Chapter 2

Problem Statement

The segmentation of foreground objects in a video sequence, without having any prior information about the nature of the objects, entails a big number of challenges ranging from the camera sensor selected to record the scene, to achieve a precise segmentation of the objects avoiding as far as possible false detection errors. But, what is exactly a foreground object? One foreground object is an entity which appears in a region under analysis and presents enough interest to the observer to be classified and separated as a new detected object. This implies that foreground objects, are those which contribute to give new important information to the scene under analysis, and as a yin and yan they are always related to the concept of background, or what is equivalent, what we consider that does not give any additional semantic information about the sequence to the observer. In order to show the foreground segmentation concept, Figure 2.1 displays one example where the foreground detection of the person under analysis appears in white color and the background regions in black. As shown in the example, a correct foreground segmentation has to present low percentage of false positive and false negative detections allowing a precise segregation of the object under analysis.

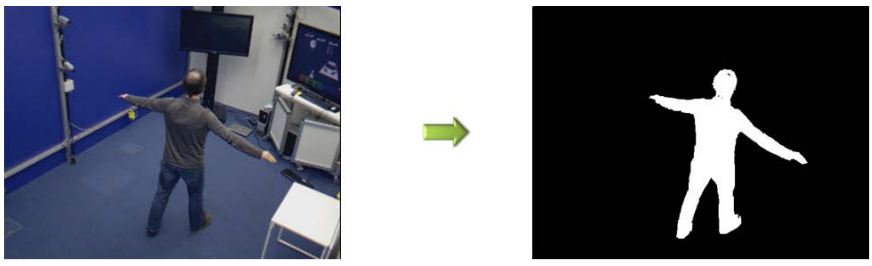


Figure 2.1: Example of foreground segmentation inside a room. In the left: original RGB image. In the right: foreground segmentation of the scene (white color represents the foreground pixels, black color the background ones.

Therefore, the segmentation of the foreground objects entails an initial learning about either the background of the sequence, or otherwise, which foreground object we are going to segment, allowing in any case the correct separation of the object from the background. We can now intuit that the quality of the foreground segmentation will depend on the difference that both, foreground and background classes present along the sequence, and this factor will be given, in a high manner, by the characteristics of the sequences that we need to segment.

Hence, in order to identify the challenges to solve when detecting the foreground, we can classify the sequences to analyze according to two criteria: the characteristics of the scenario under analysis, and the configuration of camera sensors that are recording the scene. It is obvious that these criteria follow a dependent relationship one another, such that the characteristics of the scenario will define the kind of sensors necessary to better analyze the scene, their number and position. All together will impose some constraints to the design that will be used to segment the foreground objects from the background regions, according to the difficulty that each one presents. The following sections deal with an in depth study of both aspects.

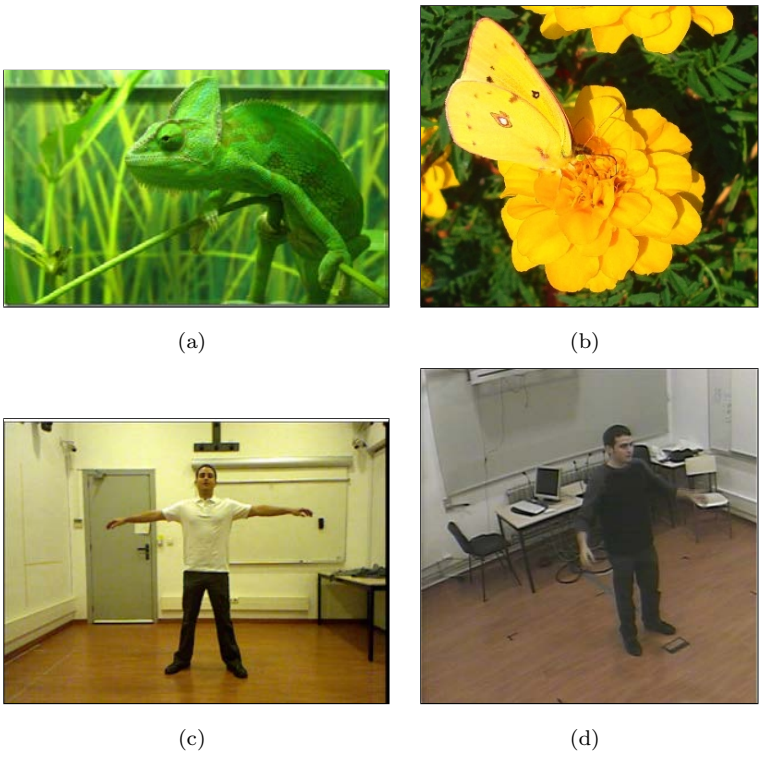
2.1 Scenario Characteristics

Several factors that affect the foreground segmentation are grouped within this point. One of the most important is whether the sensors are recording an inside or outside region, which will define the so important illumination and meteorological conditions (rainy, cloudy and windy situations) that can modify drastically the performance of the scenario under analysis. Moreover, the configuration of the scenario is central to the segmentation: is it a dynamic or static background, which objects/people we are going to analyze or if it is a crowded or non-crowded scenario among others.

Although there are many different situations that will influence in the foreground segmentation process, there are three main problems that can appear in the recordings, which difficult the foreground segregation process:

- **Camouflage situation between foreground and background.** This situation appears when foreground and background present regions with high similarity in the analysis domain. We have to consider that camouflage often appears in nature and real life, as we can see in the first row of Figure 2.2, and it is necessary to deal with this characteristic in any segmentation system. The video sequences that we are going to analyze, can present camouflage situations that affect the objects/people to segment, but in general, they will be less strong than the ones presented in Figures 2.2(a) and 2.2(b). Figures

2.2(c), 2.2(d) show examples of foreground-background camouflage situations in indoor video sequences. As we can observe in both pictures, the upper part of the person under analysis presents a RGB color very similar to the background. Hence, to maintain a correct segmentation in these complicated regions is a challenge in any image processing system.



(a) (b)

(c) (d)

Figure 2.2: Examples of camouflage situation. First row shows animal and insect camouflage in the nature. Second row shows examples of camouflage regions that will appear in the sequences under study.

- **Illumination setup.** When working with color camera sensors, the type of illumination will define the color tonality of the objects. Moreover, shadow and highlight phenomenons appear as a consequence of the illumination configuration and their incidence over the foreground objects and in general, over the scenario setup. Figure 2.3 shows an example of the illumination effects in indoor scenarios. As we can observe, the two people projects its shadow in the ground, while the highlights change the lightness of the regions affected by this effect. Figure 2.4 shows and example of outdoor scenario in sunny/cloudy conditions where the scenario changes drastically in few seconds due to the effects of the clouds occluding the sun light. In order to understand better the shadow and highlight effects, a brief explanation is written now:
  - Shadows: the intensity, position and direction of the illumination source

(a)

(b)

Figure 2.3: Examples of the illumination effect: Shadows and highlights in indoor scenarios.

can produce shadows over the scenario under analysis. Cast shadows are the source of several false positive detections in foreground segmentation tasks. It is well known that the detection of moving foreground objects generally includes their cast shadow, as a consequence of the background color and brightness modifications that the object produces when it occludes the light source. The undesirable consequences that shadow effect causes in the foreground segmentation are the distortion of the true shape and color properties of the object. Hence, to obtain a better segmentation quality, detection algorithms must correctly separate foreground objects from the shadows they cast.

- Highlights: they are areas of exceptional lightness in an image, and depend on the incidence angle of the light over the objects and the refractive index of the materials. Many false detections appear in the foreground segmentation process due to these effects. For instance, cluttered scenes in the background such as trees should not be detected as new objects when being illuminated by sun lights.

- **Dynamic background.** Preserving the background configuration is central to achieve a correct foreground segregation along the scene under analysis. Since foreground segmentation techniques are based on learning the background, all the modifications that appear in the scene, will impair the final segmentation results by increasing the false positive detections. Dynamic

(a)

(b)

Figure 2.4: Example of outdoor recording in a sudden change from sunny day to cloudy effect due to meteorological conditions.

(a) (b) (c)

Figure 2.5: Example of outdoor recording with dynamic background. The water of the fountain and the leaves of the trees give an special difficulty to this scenario, since the background is constantly changing along the scene.

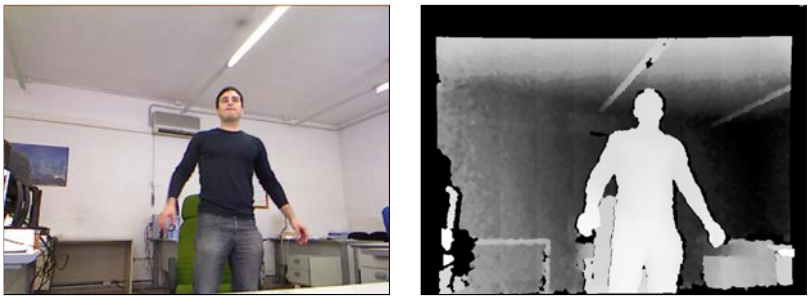
background appears when there are moving objects or surfaces behind the objects of interest. For instance, the tree leaves, one flag moving on the wind or the water of a fountain. Figure 2.5 shows an example where the water of the fountain and the leaves of the trees produce a noisy background that presents high difficulty to be modeled.

2.2 Configuration of the Camera Sensors

This point gathers some characteristics of the sensors that are central to the segmentation issue:

- **Type of sensors**. Currently, there are several kind of camera sensors that can be used to record the scene in different spaces like: color RGB, gray scale,

depth, Infra Red and Thermal cameras. The most common are the color RGB camera and the new depth sensors, being the Infra Red and Thermal cameras used for some specific applications. Figures 2.6 displays one example of color and depth images.



(a) (b)

Figure 2.6: Example of one image recorded with color camera and depth camera sensors. On the right hand, the darker the pixel, the deeper the region according to the distance from the sensor.

- **Movement of the sensors**. The movement of the cameras during the recording of the sequence will condition strongly the techniques to use for segmenting the foreground. The three possible situations are: static camera, where the camera is situated in a fixed position, moving camera with constrained motion, commonly used on surveillance scenarios where the camera performs a repeated movement to control a wide area, and moving camera with free movement, used when there is an object of interest which performs free movements along the sequence and the camera is focusing on it.

- **Position of the sensors**. This factor is mainly related to the place and position of the camera with respect to the objects that we want to segment. The distance to the foreground objects under analysis and the angle of analysis are the most important factors to take into account.

- **Number of sensors**. When using more than one sensor to record the same scene from different positions, the foreground segmentation can be widely improved by means of combining the information that all the sensors are giving us about the region under analysis. In this case, the redundancy of the data analyzed by each one of the sensors can results a more robust segmentation than the one obtained using just one sensor.

  – Single camera: can be either static or moving in indoor or outdoor scenarios. These characteristics, and the distance from the camera to the region of interest, are important factors in order to identify the challenges that will appear in the sequence. Far distances are typical from surveillance purposes. Close distances are commonly used for person segmentation and behavior analysis.

– Multi-camera: this framework is characterized for presenting more than one camera sensor recording partially or completely overlapped regions from different points of view. Multi-camera environment can be applied to smart-room scenarios, where the camera sensors are calibrated in order to obtain 3D reconstructions of the foreground objects.

2.3 Conclusions

The scenario characteristics and the type of sensors used to record the scene will condition the type of foreground segmentation technology necessary to carry out a correct foreground detection of the sequences.

Therefore, as we have seen, although the objective of the foreground segmentation challenge can be easily recognized and explained, there are many possible combinations of scene configurations and acquisition setups that make the foreground segmentation solutions divided according to the specific necessities of each situation. Hence, the solution to the foreground segregation problem is not unique for all the cases and must be understood as a group of techniques specific for certain setups.

In this thesis we deal with foreground segmentation techniques that improve the state of the art in some specific scenarios. We will start analyzing the use of parametric models in single color camera for indoor scenarios, and we will extrapolate the segmentation process to other acquisition setups and scenario characteristics from 2-dimensional scenarios to multi-view 3-dimensional framework. In the next chapter, we give an overview of the main state of the art methods devoted to the foreground segmentation analysis.

Chapter 3

Reference Work

Foreground segmentation implies the definition and identification of the background inside the image to achieve a correct foreground/background segregation. In such a way, most foreground detection methods of the literature are based in one way or another on learning the background of the scene under analysis in order to identify the variations that appear along the sequence and consider them as candidates to foreground objects. This is called *exception to background analysis*. Once the foreground objects have been detected, some techniques propose to take into account the objects information in order to improve the foreground detection, thus learning the characteristics of the foreground objects as well. Therefore, an initial classification of the foreground detection techniques is defined according to this criteria: foreground segmentation methods that only use background learning or methods that use both background and foreground learning.

The way that each system uses to represent or model each class (foreground and background) can be used to establish the second classification. In the literature we can recognize two big groups of proposals according to this:

- **Use of pixel-wise model:** these models consider each pixel as a separated entity of the image, thus proposing and independent analysis for each one. Pixel-wise modeling has been widely used to achieve a precise representation of the static background, since it works at pixel resolution. In this case, foreground pixels are detected by analyzing the differences between the input value and the pixel model. Usually, classes modeled at pixel-wise level are very sensitive to small variations that can appear due to illumination changes, shadows or dynamic background.

- **Use of region-based model:** this model is used to achieve the joint characterization of a group of pixels. Hence, the modeling of each pixel results less

precise than the pixel-wise modeling, but it is less sensitive to small variations of the image and it is more spatially flexible.

Table 3.1 displays this initial classification of the foreground segmentation systems.

In this chapter, we are going to analyze the state of the art methods according to these criteria. We have organized this overview first, considering different acquisition setups from 2D planar scenarios to multi-view 3D framework, and second, grouping the techniques according to the scenario and the application where each one is applied.

Table 3.1: General classification of the foreground segmentation methods

Kind of model	Class where the model is applied
Pixel wise	Background
Region Based	Foreground

3.1 Foreground Segmentation Using One Camera Sensor

Foreground segmentation using a single camera sensor (also called planar foreground detection) is the most studied area in the foreground detection challenge. All the techniques developed with this setup, can be used in many computer vision applications, such as automatic video surveillance (which could include tracking and activity understanding), human-computer interaction, object oriented encoding as in MPEG-4, etc. Moreover, they can be applied to other acquisition setups like stereo or multi-view sensors to obtain, for instance, depth information or volumetric foreground representations in the 3D space.

There are many different planar foreground segmentation approaches described in the literature. These techniques can be grouped according to the Table 3.1 and will be shown in the following subsections:

3.1.1 Foreground Segmentation Using Background Modeling

All the techniques grouped within this category are also called *exception to background segmentation systems*. They base the foreground detection process on obtaining an initial representation of the background and, for each frame of the sequence, analyze if the input pixel values belong or not to the background learned at

the beginning. [Pic05] gives an overview of some foreground segmentation methods based on background modeling.

The background modeling consists in creating statistical models of the background process of every pixel value, i.e., motion, color, gradient, luminance, etc. Then, the foreground segmentation is performed at each pixel as an exception to the modeled background [EHD00, HHD99, HHD02, SG00, WADP02].

Most of these techniques are thought for static camera devices since the staticity can ensure the correct learning of the background, at pixel-wise level, and its stability along the sequence under analysis. These methods usually share the following work-flow:

- Training period: N frames free of foreground objects used to learn the background.
- Process the sequence frame by frame:
  - Classify the pixels in foreground and background.
  - Update the background model according to the classification obtained.

The main techniques of the state of the art are explained below:

3.1.1.1 Temporal Median Filter

Pixel-wise method proposed by Lo and Velastin in [LV02] for foreground (fg) segmentation in static camera sequences. The approach consists in utilizing the median of the intensity value for each pixel of the image to perform the background model which in this approach, can be understood as a reference background (bg) image I_{bg}. The system uses the N last frames of the sequence to obtain the median intensity value of each pixel of the image $i \in I_{\text{bg}}$, hence, a FIFO (First In First Out) buffer for every pixel of the image is needed in order to save the corresponding N last color values $c_i = RGB$ where R=red, G=green and B=blue are the channels of the image. The work-flow of the system is as follows:

- Initialization: Training period of N frames free of foreground objects. The background reference image I_{bg} can be created by obtaining the median value in each pixel:

$$c_{\text{bg},i} = \text{median}(c_i, N), \tag{3.1}$$

- Process the sequence frame by frame:

– Classify the pixels in foreground and background. The pixels of the input frame at time t, $I_{t,i}$, will be considered as foreground if they accomplish the following criterion:

$$c_{\mathrm{bg},i} - \mathrm{th} < c_i < c_{\mathrm{bg},i} + \mathrm{th}, \tag{3.2}$$

where th denotes a threshold value defined by the user.

– Update the background model with the pixel value, only when the pixel has been considered as background. Hence, we will include the pixel value in the buffer in order to update the background image with progressive changes that can affect the background.

The main disadvantage of a median-based approach is that its computation requires a buffer with the recent pixel values. Moreover, the median filter does not accommodate for a rigorous statistical description and does not provide a deviation measure for adapting the subtraction threshold.

3.1.1.2 Running Gaussian Average

Foreground detection method proposed in [WADP02], appropriate for monocular static camera sequences in the gray scale images, color RGB or chroma YUV domain. In this approach, the authors propose to model the background independently at each pixel location i based on ideally fitting a multi-variate Gaussian probability density function (pdf) on the last n pixel values. Considering color images with $c = RGB$ channels, the likelihood of the background model for the pixel i is:

$$\begin{aligned} P(c_i|\mathrm{bg}) &= G(c_i, \mu_{c,i}, \sigma_{c,i}) = \\ &= \frac{1}{(2\pi)^{3/2}|\Sigma_{c,i}|^{1/2}} \quad \exp\left[-\frac{1}{2}(c_i - \mu_{c,i})^T \Sigma_{c,i}^{-1}(c_i - \mu_{c,i})\right], \end{aligned} \tag{3.3}$$

where $c_i \in \mathbb{R}^3$ is the input pixel value in the color $c \equiv \{RGB\}$ domain, $\mu_{c,i} \in \mathbb{R}^3$ denotes the mean value of the Gaussian, and $\Sigma_{c,i} \in \mathbb{R}^{3\times 3}$ is the covariance matrix. We introduce the subindex c in the formulation in order to denote that the model parameters are working in the color domain. This notation will be useful in next sections where probabilistic models will work in the spatial s and depth d domains as well.

The approach proposes to simplify the model and to speed up the foreground segmentation process by assuming uncorrelated RGB channels, thus defining Σ_c as:

$$\Sigma_c = \begin{pmatrix} \sigma_R^2 & \sigma_{RG} & \sigma_{RB} \\ \sigma_{GR} & \sigma_G^2 & \sigma_{GB} \\ \sigma_{BR} & \sigma_{BG} & \sigma_B^2 \end{pmatrix} = \begin{pmatrix} \sigma_R^2 & 0 & 0 \\ 0 & \sigma_G^2 & 0 \\ 0 & 0 & \sigma_B^2 \end{pmatrix}. \tag{3.4}$$

where σ_x^2 is the variance value for the x channel. Moreover, the background model is even more simplified by considering equal variances for the three channels so that, $\sigma_R^2 = \sigma_G^2 = \sigma_B^2 = \sigma^2$, thus avoiding specific updates for each channel.

Having simplified the background model in such a way, in order to avoid fitting the pdf from scratch at each new frame time, I_t, a running (or on-line cumulative) average is computed instead for each pixel i as:

$$\mu_{t,i} = \rho\ c_{t,i} + (1-\rho)\mu_{t-1,i}, \tag{3.5}$$

where ρ is an empirical weight often chosen as a trade-off between stability and quick update. Although not stated explicitly in [WADP02], the other parameter of the Gaussian pdf, the standard deviation σ can be computed similarly:

$$\sigma_{t,i} = \rho\ \sigma_{t,i} + (1-\rho)\sigma_{t-1,i}, \tag{3.6}$$

In addition to speed, the advantage of the running average is given by the low memory requirement: for each pixel, this consists of the two parameters ($\mu_{c,i}$, $\sigma_{c,i}$) instead of the buffer with the last n pixel values.

- Initialization: Training period of N frames free of foreground objects. The background model is initialized in the following way:
  - Frame $t = 0$: $\mu_{t=0,i} = c_{t=0,i}$; $\sigma_{t=0,i} = \sigma_{init}$, where σ_{init} is an initial value defined by the user.
  - Next training frames: the background model is updated according to the Equations 3.5, 3.6.
- Process the sequence frame by frame:
  - Classify the pixels in foreground and background. The pixels of the input frame at time t $I_{t,i}$ will be considered as foreground if the next inequality holds:

    $$||c_{t,i} - \mu_{t,i}||_2 > k\sigma_{t,i}, \tag{3.7}$$

    where $||.||_2$ is the euclidean distance. Considering that $\frac{||c_{t,i}-\mu_{t,i}||_2}{\sigma_{t,i}}$ is the Mahalanobis distance, we are normalizing the euclidean distance between the input pixel value $c_{t,i}$ and the mean value of the Gaussian that models the pixel $\mu_{t,i}$ by the variance $\sigma_{t,i}^2$ of the model. Hence, k is a factor which denotes the number of standard deviations tolerated in terms of distance, to consider a pixel belonging to the background.
  - Update the background model with the pixel value, just when the pixel has been considered as background. [KWH+02] remarked that the model

should be updated just in the case of background classification. For this reason, they propose the model update as:

$$\mu_{t,i} = M\mu_{t-1,i} + (1 - M)(\rho c_{t,i} + (1 - \rho)\mu_{t-1,i}), \tag{3.8}$$

$$\sigma_{t,i} = M\sigma_{t-1,i} + (1 - M)(\rho c_{t,i} + (1 - \rho)\sigma_{t-1,i}), \tag{3.9}$$

where the binary value M is 1 in correspondence of a foreground value, and 0 otherwise, and ρ is the adaptation learning rate used, which could be proportional to the probability $G(c_i, \mu_{c,i}, \sigma_{c,i})$ that the Gaussian presents or, as it is proposed in the paper, by defining $\rho = 0.01$. The equations work as a low-pass filter where past samples contribute more to the final value than the last one, and reduce the computation to provide the Gaussian updating. By updating the mean and the variance, the system is allowed to adapt to slow illumination changes.

If real-time requirements constrain the computational load, the update rate of either μ, or σ can be set to less than that of the sample (frame) rate. However, the lower the update rate of the background model, the less a system will be able to quickly respond to the actual background dynamic.

3.1.1.3 Mixture of Gaussians

Over time, different background values are likely to appear at the same pixel location. When this is due to a progressive change in the scene's properties, the models reviewed so far will, more or less promptly, adapt so as to reflect the value of the current background object. However, sometimes the changes in the background object are not permanent and appear at a rate faster than that of the background update. A typical example is that of an outdoor scene with trees partially covering a building: a same pixel location will show values from tree leaves, tree branches, and the building itself. Other examples can be easily drawn from snowing, raining, or watching sea waves from a beach. In these cases, a single-valued background is not an adequate model.

In [SG00], Stauffer and Grimson (S&G) raised the case for a multi-valued background model able to cope with multiple background values. In this method, different multi-variate Gaussians are assumed to characterize color RGB appearances in each pixel, and each one is weighted (ω) depending on how often the Gaussian has explained the same appearance. Mixtures of Gaussians (GMM) have been also used in the literature [HHD99, SG00] to ensure that repetitive moving background can be represented by different probabilistic functions.

Given the parameter set for each one of the pixels $\theta_{\text{bg},t} \equiv \{\omega_{t,k}, \mu_{t,k,c}, \Sigma_{t,k,c}\}$, the likelihood of the model for the pixel i is defined as follows:

$$P(c_{t,i}|\theta_{\text{bg},t}) = \sum_{k=1}^{K_{\text{bg}}} \omega_{t,k}\ G_{\text{bg}}(c_{t,i}, \mu_{t,k,c}, \sigma_{t,k,c}) = \\ = \sum_{k=1}^{K_{\text{bg}}} \omega_{t,k} \frac{1}{(2\pi)^{3/2}|\Sigma_{t,k,c}|^{1/2}} \exp\left[-\frac{1}{2}(c_{t,i} - \mu_{t,k,c})^T \Sigma_{t,k,c}^{-1}(c_{t,i} - \mu_{t,k,c})\right] \tag{3.10}$$

where K_{bg} is the total number of Gaussians used in each pixel, and ω_k is the prior probability (often referred as the weights of the Gaussians) that a background pixel is represented by a certain mode k of the mixture of Gaussians where $\sum_{k=1}^{K_{\text{bg}}} \omega_{t,k} = 1$. In practical cases, K_{bg} is set to be $K_{\text{bg}} = 3$ or $K_{\text{bg}} = 5$.

Analogously to 3.1.1.2, Gaussians are multi-variate to describe the color $c = RGB$ channels. These values are assumed independent, so that the co-variance matrix $\Sigma_{k,c}$ simplifies to diagonal. In addition, if the standard deviation for the three channels is assumed the same, it further reduces to a $\mathbf{I}\sigma_{k,c}$, where $\mathbf{I}$ is the identity matrix.

The probabilistic model defined for each pixel describes both, the background and the foreground classes. Hence, for each frame of the sequence, the pixels are analyzed independently, checking if the input color $c = RGB$ value of each pixel, c_i, matches any of the Gaussians of the model that represents the pixel. If so, the pixel will result foreground or background according to the class that the Gaussian is modeling. Otherwise, a new Gaussian is created and the least important Gaussian of the model is deleted.

The distributions are ranked in descending order based on the ratio between their weight ω_k and their standard deviation σ_k: $\eta_k = \frac{\omega_k}{\sigma_k}$. The assumption is that the higher and more compact the distribution, the more likely it is to represent the background, since the first few Gaussians in the list correspond to the ones with more supporting evidence (high weight imply more times explaining incoming pixels) at the lowest variance (explained incoming pixels are always very similar).

Then, the first B distributions in ranking order are accepted as background if they satisfy:

$$\sum_{i=1}^{B} \eta_i > T, \tag{3.11}$$

with T an assigned threshold usually fixed as $T = 0.6$.

The matching criterion for each one of the Gaussians of the model in every pixel i is defined analogously to the matching criterion proposed in the *Running Gaussian Average* system (Equation 3.7):

$$||c_{t,i} - \mu_{t,k,i}||_2 > 2.5\ \sigma_{t,k,i}, \tag{3.12}$$

where $||.||_2$ is the euclidean distance. The first in ranking order is accepted as a match for c_i. Furthermore, parameters $\theta_{\mathrm{bg},t} \equiv \{\omega_{t,k}, \mu_{t,k,c}, \Sigma_{t,k,c}\}$ are updated only for this matching distribution and by using simple on-line cumulative averages similar to those of Equation 3.8 and 3.9. The weighting factor ω is updated for the Gaussian that matches the input pixel as:

$$\omega_{t,k,i} = (1 - \alpha)\omega_{t-1,k,i} + \alpha(M_{t,k}), \tag{3.13}$$

where α stands for the updating factor, and $M_{k,t}$ is 1 for the Gaussian that has matched the input value, and 0 for the rest. $\alpha = 0.005$ is a common value. Thus, the more often a Gaussian explains an incoming pixel, the higher is its associated weight. Note that this is a low-pass filter average of the weights, where last samples have exponentially more relevance than older ones. The configuration of this updating produces the static foreground objects, which remain static for a certain period of time, to be integrated to the background model. Rather than a drawback, this is a design choice of the authors which has to be taken into consideration before employing the method without further modifications in any scenario.

If no match is found between the background Gaussians and the input value c_i, the last ranked distribution is replaced by a new one centered in c_i, with low weight and high variance.

Regarding the initialization of the model, a training period of N frames free of foreground objects is used while running the algorithm.

3.1.2 Foreground Segmentation Using Background and Foreground Modeling

As we have seen in the previous section, when there only exists a complete model of the background class, the foreground segmentation task is a problem of one-class classification [DR01, PR03] assuming the exception to background detection.

When a foreground model is also available, the foreground detection can be proposed as a Bayesian classification process between foreground and background classes. A Bayesian approach for foreground segmentation is important because it provides a natural classification framework supported on probabilistic models. In

[WADP02] and similar approaches, a Bayesian formulation is not possible since the foreground process is not modeled or it is only partially modeled.

3.1.2.1 Bayesian Classifiers

A Bayesian classifier performs the classification task by using the probability that a pixel sample belong to the foreground (fg) and background (bg) classes. If we use the pixel probabilities in order to achieve the final labeling of the pixels of the image, this can lead to a robust classification process since it utilizes the statistical information of the objects under analysis, thus improving the decision process.

In order to introduce the more general Bayesian classification approach, both foreground and background models (likelihoods) have to be available. Then, the probability that a pixel $i \in I_t$ belongs to one class $l \in \{\text{fg}, \text{bg}\}$, given the observation $I_{t,i}$, can be expressed as:

$$P(l|I_{t,i}) = \frac{P(I_{t,i}|l)P(l)}{P(I_{t,i})}, \tag{3.14}$$

where $P(l|I_{t,i})$ is called posterior probability, $P(I_{t,i}|l)$ is the likelihood of the model, $P(I_{t,i})$ is the probability to observe the input data and $P(l)$ is the prior probability, which depends on the application. However, approximated values for $P(l)$ can be easily obtained for each application by manually segmenting the foreground in some images, and averaging the number of segmented points over the total.

Once the posterior probabilities have been obtained, a simple pixel classification can be computed by comparing foreground and background probabilities. The pixel i will be labeled as foreground if the following inequality holds:

$$P(\text{fg}|I_{t,i}) > P(\text{bg}|I_{t,i}), \tag{3.15}$$

or what is equivalent, since $P(I_{t,i})$ is the same for both classes and thus, it can be disregarded:

$$P(I_{t,i}|\text{fg})P(\text{fg}) > P(I_{t,i}|\text{bg})P(\text{bg}). \tag{3.16}$$

If the inequality is not accomplished, the pixel will be classified as background.

As it has been previously mentioned, Bayesian classification [KS00a, MD03] can only be performed when there exist explicit models of the foreground entities in the scene. In order to create these models, an initial segmentation is usually performed as an exception to the modeled background, and once there is sufficient evidence that the foreground entities are in the scene, foreground models are created.

Similarly as with background models, foreground models are Gaussian-based in most of the cases. For instance, single-Gaussians have been used in [WADP02], MoGs have been used in [KS00a, MRG99] and nonparametric models with Gaussian kernels, in [EHD00, MD03]. On the other hand, foreground models can also be as simple as a uniform pdf. Simple models are useful when there does not exist any intention to model the foreground process or if the foreground is difficult to model for any reason.

Most of these methods propose a pixel-based foreground modeling, but since the foreground objects are constantly moving along the scene, some rotations and displacements of the object produce that new foreground regions appear along the scene, as well as other foreground regions can disappear due to occlusion situations. Because of that, pixel-wise foreground models are difficult to build an update, and region-based foreground models arise as a robust solution for these situations.

Foreground segmentation systems that work with uniform and region-based foreground models are presented in the following sections.

3.1.2.2 Pixel-Wise Foreground Segmentation by Means of Foreground Uniform Model and Background Gaussian Model

This is a pixel-wise foreground segmentation approach for monocular static sequences that combines background and foreground probabilistic modeling. In order to obtain an accurate 2D foreground segmentation using a Bayesian framework, [LP06a] proposes a pixel-wise Gaussian model to characterize the RGB color of the background pixels, and a uniform statistical model to model the foreground.

Hence, given observations of pixel color value across time c_i, a Gaussian probability density function is used to model the background color analogously to Section 3.1.1.2:

$$\begin{aligned} P(c_i|\text{bg}) &= G(c_i, \mu_{\text{bg},c,i}, \sigma_{\text{bg},c,i}) = \\ &= \frac{1}{(2\pi)^{3/2}|\Sigma_{\text{bg},c,i}|^{1/2}} \quad \exp\left[-\frac{1}{2}(c_i - \mu_{\text{bg},c,i})^T \Sigma^{-1}_{\text{bg},c,i}(c_i - \mu_{\text{bg},c,i})\right], \end{aligned} \tag{3.17}$$

where $c_i \in \mathbb{R}^3$ is the i-th input pixel value in the color $c = RGB$ domain, $\mu_{\text{bg},c,i} \in \mathbb{R}^3$ denotes the mean value of the background Gaussian that models the color c of pixel i, and $\Sigma_{\text{bg},c,i} \in \mathbb{R}^{3\times3}$ is the diagonal covariance matrix with RGB channels sharing the same variance value: $\Sigma_{\text{bg},c,i} = \mathbf{I}\sigma_{\text{bg},c,i}$.

The adaptation of the background model is the same proposed in [WADP02] explained in Section 3.1.1.2.

The foreground model is based on a uniform pdf to model the foreground process in each pixel, which is in fact the probabilistic extension of classifying a foreground pixel as an exception to the model. Since a pixel admits 256^3 colors in the RGB color space, its pdf is modeled as:

$$U(c_i) = \frac{1}{256^3} \tag{3.18}$$

Once the foreground and background likelihoods of a pixel are introduced, and assuming that we have some knowledge of foreground and background prior probabilities, $P(\text{fg})$ and $P(\text{bg})$ respectively (approximate values can be obtained by segmenting the foreground in some images, and averaging the number of segmented points over the total), the classification of a pixel as foreground can be done when the inequality presented in Equation 3.16 for color domain c is verified:

$$\begin{aligned} P(\text{fg}|c_i) &> P(\text{bg}|c_i), \\ P(c_i|\text{fg})P(\text{fg}) &> P(c_i|\text{bg})P(\text{bg}), \\ \frac{1}{256^3}P(\text{fg}) &> G(c_i, \mu_{\text{bg},c,i}, \sigma_{\text{bg},c,i})P(\text{bg}), \end{aligned} \tag{3.19}$$

In practice this is very similar to the approach defined in Section 3.1.1.2 consisting in determining background when a pixel value falls within 2.5 standard deviations of the mean of the Gaussian.

3.1.2.3 Region-based Foreground Segmentation Based on Spatial-Color Gaussians Mixture Models (SCGMM)

The system proposed in [YZC+07], is a good example of the SCGMM application to the foreground segmentation task. This method presents an approach to segment monocular videos recorded by static cameras, where both foreground and background classes are modeled using spatial-color Gaussian mixture models. Figure 3.1 shows the spatial representation of the foreground and background models. Hence, each pixel of the image is defined with five dimensional feature vector, i.e., $z = RGB\ XY$, representing the color $c = RGB$, and spatial $s = XY$ coordinates of the pixels. Then, the likelihood of a pixel $i \in I_t$ belonging to the foreground or background classes can be written as:

Figure 3.1: Spatial representation of the SCGMM models. Each ellipse is the spatial representation of each Gaussian of the models. Foreground SCGMM in red, background SCGMM in green.

$$\begin{aligned} P(z_i|l) &= \sum_{k=1}^{K_l} \omega_{l,k} G_l(z_i, \mu_{l,k}, \Sigma_{l,k}) \\ &= \sum_{k=1}^{K_l} \omega_{l,k} \frac{1}{(2\pi)^{5/2}|\Sigma_{l,k}|^{1/2}} \exp\left[-\frac{1}{2}(z_i - \mu_{l,k})^T \Sigma_{l,k}^{-1}(z_i - \mu_{l,k})\right], \end{aligned} \tag{3.20}$$

where $l \in \{\text{fg}, \text{bg}\}$ represents foreground or background, $\omega_{l,k}$ is the prior weight of the Gaussian component in the MoG. See Appendix A.3 for more information about GMM.

It is commonly assumed that the spatial and color components of the SCGMM models are decoupled, i.e., the covariance matrix of each Gaussian component takes the block diagonal form,

$$\Sigma_k = \begin{pmatrix} \Sigma_{k,s,} & 0 \\ 0 & \Sigma_{k,c} \end{pmatrix}$$

where s and c stand for the spatial and color features respectively. With such decomposition, each foreground Gaussian component has the following factorized form:

$$G_{fg}(z_i, \mu_k, \Sigma_k) = G(s_i, \mu_{k,s}, \Sigma_{k,s})\ G(c_i, \mu_{k,c}, \Sigma_{k,c}), \tag{3.21}$$

where $s_i \in \mathbb{R}^2$ is the pixel's spatial information and $c_i \in \mathbb{R}^3$ is its color value. The parameter estimation can be reached via Bayes' development, with the EM algorithm [DLR+77]. For this estimation an initialization frame is needed, containing a first segmentation of the foreground object. This initialization can be performed with an exception to the background scheme.

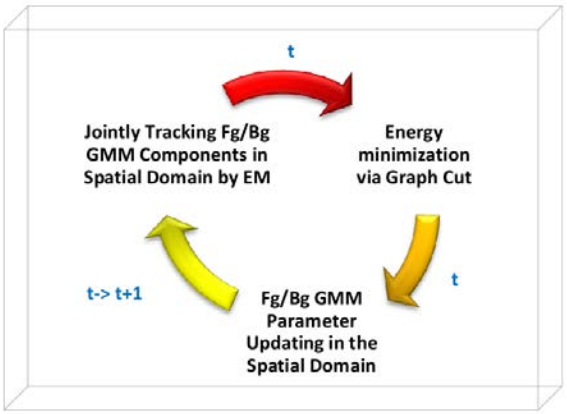


Figure 3.2: Work flow of the system proposed in [YZC+07].

The foreground segmentation using this model is obtained finding the evolution of the foreground-background five dimensional SCGMM models for each video frame, and deciding for each pixel, the one that maximizes the class probability.

3.1.2.3.1 Tracking Spatial Color Gaussian Mixture Models (SCGMM)

With this technique, the authors propose to combine the two SCGMMs into a generative model of the whole image, and maximize the joint data likelihood using a constrained Expectation-Maximization (EM) algorithm [DLR+77] (see Appendix A.4).

Using spatial and color information to model the scene, SCGMM has better discriminative power than color-only GMM widely used in pixel wise analysis.

The segmentation problem is solved by means of iterating the tracking-segmentation-updating process shown in Figure 3.2.

The first frame of the sequence is used to initialize the foreground and background models by means of the EM algorithm in both models. Hence, an initial classification into foreground and background pixels is needed. For each frame after the first one, first the SCGMM of the foreground and the background are combined and updated with the EM, thus performing a joint tracking of the foreground regions and the background. Afterwards the image SCGMM model is split back into two models, one describing the foreground, the other describing the background. Components belonging to the foreground before tracking are placed in the foreground SCGMM, and components belonging to the background before tracking are placed in the background SCGMM. The two SCGMM models are then used to perform graph cut segmentation (see Appendix B to extend the graph cuts information).

The segmentation results can be used for a post-updating of the models, where the foreground and background SCGMMs are trained separately with the seg-

mented pixels, which often provides better discriminative power for segmenting future frames.

Considering that the foreground and background colors stay the same across the sequence, a constrained update on the two models is performed. That is, apply Expectation Maximization (EM) algorithm on the foreground or background region to update the SCGMM models, forcing the color means and variances to be constant. In this way, propagation errors due to color updates are avoided. The joint tracking, energy minimization and updating steps are explained in the following sections.

3.1.2.3.2 SCGMM Joint Tracking

Given two SCGMM models, each one to characterize the $l \in \{\text{fg}, \text{bg}\}$ foreground and background classes, defined by a set of parameters $\theta_{l,t} \equiv \{\omega_{l,t,k}, \mu_{l,t,k}, \Sigma_{l,t,k}\}$. Both models are learned during the system initialization period, in the first frame $t = 0$, using the EM algorithm which maximizes the data likelihood (ML) of each segment:

$$\begin{aligned} \text{ML}\{\theta_{l,t=0}\} &= \arg\max_{\theta_{l,t=0}} L(\theta_{l,t=0}|I_{t=0,l}) = \\ &= \arg\max_{\theta_{l,t=0}} \prod_{z_{i,l} \in I_{t=0,l}} [P(z_{i,l}|\theta_{l,t=0})] = \\ &= \arg\max_{\theta_{l,t=0}} \prod_{z_{i,l} \in I_{t=0,l}} \left[\sum_{k=1}^{K_l} \omega_{l,k} G(z_{i,l}, \mu_{l,k}, \Sigma_{l,k}) \right], \end{aligned} \tag{3.22}$$

where $z_i \in \mathbb{R}^5$ is the input feature vector for pixel i in the $z = RGB\ XY$ domain.

An Expectation Maximization algorithm can be formulated to find the maximizer of the likelihood function. The aim of this part of the process is to propagate these SCGMM models over the rest of the sequence, since both foreground and background objects can be constantly moving. For this purpose, the algorithm looks for ways to obtain an approximate SCGMM model for the current frame before the graph cut segmentation. It is assumed that from time $t-1$ to t, the colors of the foreground and background objects do not change. Hence, the color parts of the SCGMM models remain identical:

$$G(c_i, \mu_{l,k,c,t}\Sigma_{l,k,c,t}) = G(c_i, \mu_{l,k,c,t-1}\Sigma_{l,k,c,t-1}) \tag{3.23}$$

The formulation of the updating scheme for the spatial parts $G(s_i, \mu_{l,k,s,t}\Sigma_{l,k,s,t})$ given the new input image It, is explained next:

Since we do not have a foreground/background segmentation on I_t, first a global SCGMM model of the whole image is formed by combining the foreground and back-

ground SCGMM models of the previous frame: $\theta^0_{I,t}$, where superscript 0 indicates that the parameter set is serving as the initialization value for the later update.

The probability of a pixel of the image $z_i = (r, g, b, x, y)$ given the global model $\theta^0_{I,t}$ can be expressed as the combination of both foreground and background models:

$$\begin{aligned} P(z_i|\theta^0_{I,t}) &= P(z_i|\theta_{\text{fg},t-1})\ P(\text{fg}) + P(z_i|\theta_{\text{bg},t-1})\ P(\text{bg}) = \\ &= \sum_{k=1}^{K_I} \omega^0_{k,t}\ G(s_i, \mu^0_{k,s,t}, \Sigma^0_{k,s,t})\ G(c_i, \mu_{k,c,t}, \Sigma_{k,c,t}), \end{aligned} \tag{3.24}$$

Denote $K_I = K_{\text{fg}} + K_{\text{bg}}$ as the number of Gaussian components in the combined image level SCGMM model, where the first K_{fg} Gaussian components are from the foreground SCGMM, and the last K_{fg} Gaussian components are from the background SCGMM.

The Gaussian term over the color dimension is defined in Equation 3.23 and remains fixed at this moment. The Gaussian component weights $\omega^0_{k,t}$, $k = 1, ..., K_I$, are different from their original values in their individual foreground or background SCGMMs due to $P(\text{fg})$ and $P(\text{bg})$:

$$\omega^0_{k,t} = \begin{cases} \omega^0_{\text{fg},k,t}\ P(\text{fg}) & \text{if } k \leq K_{\text{fg}} \\ \omega^0_{\text{bg},k-K_{\text{fg}},t}\ P(\text{bg}) & \text{if } K_{\text{fg}} < k \leq K_I \end{cases} \tag{3.25}$$

where $P(\text{fg})$ and $P(\text{bg})$ are the prior probabilities for each class, and are obtained by computing, in $t-1$, the area covered by each class, normalized by the total area of I_t. Thus, they satisfy $P(\text{fg}) + P(\text{bg}) = 1$.

Once foreground and background models have been combined, and for the current frame I_t, the objective is to obtain an updated parameter set over the spatial domain, which maximizes the joint data likelihood of the whole image, for all $k = 1, ..., K_I$, i.e.,

$$\{\omega_{k,t}, \mu_{k,s,t}, \Sigma_{k,s,t}\} = \arg \max_{\omega_{k,t},\ \mu_{k,s,t},\ \Sigma_{k,s,t}} \prod_{z_{i,t} \in I_t} P(z_{i,t}|\theta_{I,t}). \tag{3.26}$$

The EM algorithm is adopted here to iteratively update the model parameters from their initial values $\theta^0_{I,t}$. However, as it can be seen in Equation 3.26, unlike the traditional EM algorithm, where all model parameters are simultaneously updated, only the spatial parameters of the SCGMM models are updated in this phase, and the color parameters are kept unchanged. This can be implemented by constraining the color mean and variance to be fixed to their corresponding values in the previous frame (see Equation 3.23).

Such a restricted EM algorithm is shown below in Table 3.2. In the E-step, the posteriori of the pixels belonging to each Gaussian component is calculated, and in the M-step, the mean and variance of each Gaussian component in spatial domain are refined based on the updated posteriori probability of pixel assignment from E-step. In the literature this EM algorithm is called Expectation Conditional Maximization [MR93].

Table 3.2: Expectation Conditional Maximization.

1.st E-step, calculate the Gaussian component assignment probability for each pixel of the image i:

$$P^{(m)}(k|z_i) = \frac{\omega_k^{(m)}\ G(s_i,\mu_{k,s}^{(m)},\Sigma_{k,s}^{(m)})\ G(c_i,\mu_{k,c},\Sigma_{k,c})}{\sum_{k=1}^{K_I}\omega_k^{(m)}\ G(s_i,\mu_{k,s}^{(m)},\Sigma_{k,s}^{(m)})\ G(c_i,\mu_{k,c},\Sigma_{k,c})},$$

where m denotes the iteration and K is the number of mixture components involved in the process.

2.nd M-step, update the spatial mean and variance, and the weight component as:

$$\mu_{k,s}^{(m+1)} = \frac{\sum_{z_i\in I_t} P^{(m)}(k|z_i)\cdot s_i}{\sum_{z_i\in I_t} P^{(m)}(k|z_i)},$$

$$\Sigma_{k,s}^{(m+1)} = \frac{\sum_{z_i\in I_t} P^{(m)}(k|z_i)\cdot(s_i-\mu_{k,s}^{(m+1)})\cdot(s_i-\mu_{k,s}^{(m+1)})^T}{\sum_{z_i\in I_t} P^{(m)}(k|z_i)},$$

$$\omega_k^{(m+1)} = \frac{\sum_{z_i\in I_t} P^{(m)}(k|z_i)}{\sum_{k=1}^{K_I}\sum_{z_i\in I_t} P^{(m)}(k|z_i)},$$

3.1.2.3.3 Energy Minimization

After the joint foreground/background model have been combined into a generative model of the image, the model has been updated using EM, and split back into foreground and background models, the segmentation problem is solved using energy minimization. At any time instant t, let the feature vectors extracted from the video pixels be $z_{t,i}$, $i = 1, ..., N$ where N is the number of pixels in each frame. Denote the unknown label of each pixel as $l_{t,i} \in \{\text{fg}(=1), \text{bg}(=0)\}$, and the labeling of the all the pixels of the image as $l_{I_t} = \{l_{t,1}, l_{t,2}, ...l_{t,i}...l_{t,N}\}$. In the following discussions, we may ignore subscript t when it causes no confusion.

The energy-based function is formulated over the unknown labeling variables of every pixel l_i, in the form of a fist-order Markov Random Field (MRF) energy function:

$$E(l_{I_t}) = E_{\text{data}}(l_{I_t}) + E_{\text{smooth}}(l_{I_t}) = \sum_{i\in I_t} D_i(l_i) + \lambda \sum_{\{i,j\}\in\psi} V_{i,j}(l_i, l_j), \tag{3.27}$$

where ψ denotes the set of 8-connected pair-wise neighboring pixels, $i \in I_t$ are the set of pixels of the image under analysis. The role of λ is to balance the data $D_i(l_i)$ and smooth cost $V_{i,j}(l_i, l_j)$. The above energy function can be efficiently minimized by a two-way graph cut algorithm (See Appendix B), where the two terminal nodes represent foreground and background labels.

The pair-wise smoothness energy term is modeled as:

$$E_{\text{smooth}}(l_{I_t}) = \sum_{\{i,j\}\in\psi} V_{i,j}(l_i, l_j) = \sum_{\{i,j\}\in\psi} \frac{1}{d(i,j)} \exp\left[-\frac{||c_i - c_j||^2}{2\sigma^2}\right], \tag{3.28}$$

where $c_i \in \mathbb{R}^3$ stands for the input color $c = RGB$ of pixel i, $||c_i - c_j||$ is the euclidean distance between input c_i, c_j RGB values, σ is the average distance $||c_i - c_j||$ between neighboring pixels in the image, and $d(i,j)$ is the spatial $s = XY$ distance between two pixels i and j.

This favors the segmentation boundary along regions where strong edges are detected.

The data energy term $E_{\text{data}}(l_{I_t})$ evaluates the posterior probability of each pixel belonging to the foreground or background. Given the SCGMM models, the data cost $E_{\text{data}}(l_{I_t})$ is defined as:

$$E_{\text{data}}(l_{I_t}) = \sum_{i\in I_t} D_i(l_i) = \sum_{i\in I_t} -\log P(l_i|z_i), \tag{3.29}$$

The posterior $P(l_i|z_i)$ can be calculated according to Bayes development (Equation 3.14).

3.1.2.3.4 Fg/bg GMM Parameter Updating in the Spatial Domain

Given foreground and background pixels $I_{t,\text{fg}}$, $I_{t,\text{bg}}$ obtained from the Energy Minimization step (Section 3.1.2.3.3), the objective is to obtain the updated parameter sets over the spatial domain $\theta_{\text{fg},s,t} \equiv \{\omega_{\text{fg},k,t}, \mu_{\text{fg},k,s,t}, \Sigma_{\text{fg},k,s,t}\}$ and $\theta_{\text{bg},s,t} \equiv \{\omega_{\text{bg},k,t}, \mu_{\text{bg},k,s,t}, \Sigma_{\text{bg},k,s,t}\}$, which maximizes the data likelihood of each image region $I_{t,\text{fg}}$, $I_{t,\text{bg}}$:

$$\theta_{l,s,t} \equiv \{\omega_{l,k,t}, \mu_{l,k,s,t}, \Sigma_{l,k,s,t}\} = \arg \max_{\omega_{l,k,t},\ \mu_{l,k,s,t},\ \Sigma_{l,k,s,t}} \prod_{z_{t,i}\in I_t} P(z_{t,i}|\theta_{l,t}), \tag{3.30}$$

where $l \in \{\text{fg}, \text{bg}\}$.

The spatial domain, mean and variances, are updated applying Expectation Conditional Maximization algorithm (Table 3.2) for each foreground and background models separately, forcing the color means and variances to be constant and using for each model $I_{t,\mathrm{fg}}$, $I_{t,\mathrm{bg}}$ respectively instead of all pixels. After the updating process, the work-flow shown in Figure 3.2 is executed again for each frame, obtaining as a result the foreground segmentation of each frame of the sequence.

3.1.3 Shadows and Highlights Removal Techniques

The foreground segmentation techniques explained before have to deal with shadow and highlight phenomenons in order to reduce the false positive and false negative detections that these illumination effects produce. It is well known that the detection of foreground objects generally includes their cast shadow, as a consequence of the background color and brightness modifications that the object produces when it occludes the light source. The undesirable consequences that shadow effect causes in the foreground segmentation are the distortion of the true shape and color properties of the object. On the other hand, highlights can also affect the scene depending on the background materials and the illumination source, producing false detection errors. Hence, in order to obtain a better segmentation quality, foreground segmentation techniques usually adds a post-processing step to remove shadow and highlight detections from the resultant foreground mask.

3.1.3.1 Brightness and Color Distortion Domain

One of the most exploited properties in shadow removal task is the consideration that shadow regions reduce the luminance background values while maintaining the chromaticity ones. Highlights removal algorithms are based on the same chromaticity principle but, on the contrary, these regions increase the luminance background values. A shadow is normally an area that is only partially irradiated or illuminated because of the interception of radiation by an opaque object between the area and the source of radiation. If we assume that the irradiation consists only of a white light, the chromaticity in a shadowed region should be the same as when it is directly illuminated. Hence, a normalized chromatic color space, e.g. r = R /(R + G + B), g = G /(R +G + B), is immune to shadows. However, lightness information is unfortunately lost. Thus, the analysis of the color and brightness distortion between foreground and background pixels will be useful in order to localize the shadow regions.

Brightness distortion (BD) can be defined as a scalar value that brings expected background close to the observed chromaticity line. Similarly, color distortion (CD) can be defined as the orthogonal distance between the expected color and the ob-

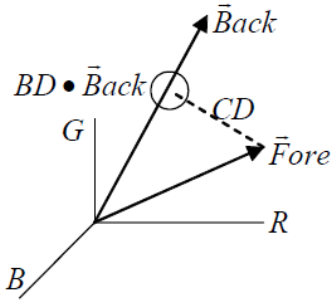


Figure 3.3: Distortion measurements in the *RGB* color space: Fore denotes the RGB value of a pixel in the incoming frame that has been classified as foreground. Back is that of its counterpart in the background.

Figure 3.4: Example of brightness distortion BD and color distortion CD domains. Values have been normalized to allow their representation in a gray scale domain. The darker the BD and CD, the smaller their values.

served chromaticity line. Both measures are shown in Figure 3.3 and formulated in (3.31).

$$\begin{aligned} \text{BD} &= \arg\min_{\beta} \|c_{i,\text{in}} - \beta c_{i,\text{bg}}\|^2, \\ \text{CD} &= \|c_{i,\text{in}} - \beta c_{i,\text{bg}}\|. \end{aligned} \tag{3.31}$$

Where $c_{i,\text{in}} \in \mathbb{R}^3$ is the i-th input pixel's value ($i = 1, ..., N$) in the *RGB* space. $c_{i,\text{bg}}$ is that of its counterpart in the background.

Figure 3.4 shows a representation of the BD and CD domains in an indoor scenario, where both values have been normalized to allow their representation in a gray scale domain. The darker the BD and CD, the smaller are their values. As we can observe, the shadow projected by the person on the ground, presents a BD and CD values that make possible their detection.

3.1.3.2 Shadow/Highlight Detection Based on BD and CD Analysis Applied in Foreground Segmentation

Many shadow/highlight detection methods like [HHD99], are based on the color and brightness distortion analysis. These shadow/highlight removal techniques are

applied after the foreground segmentation process proposed in [WADP02] (in practice, any foreground detection system is valid). Then, they analyze the foreground pixels and detect those that have similar chromaticity but lower brightness to the corresponding region when it is directly illuminated, by computing Equations 3.31. In order to do that the adaptive background reference image provides the desired information.

Hence, brightness distortion values over 1.0 correspond to lighter foreground. On the other hand, the foreground is darker when BD is below 1.0. The analysis is done for each pixel $i \in I_{t,\text{fg}}$, and a set of thresholds are defined to assist the classification into foreground, highlight or shadow pixel as shown in Algorithm 1

Algorithm 1 Thresholds for shadow and highlight detection

```
if CD < 10.0 then
    if 0.5 < BD < 1.0 then SHADOW
    else if 1.0 < BD < 1.25 then HIGHTLIGHT
    end if
else FOREGROUND
end if
```

Other methods of the state of the art are also based on this principle: In [XLP05] the authors try to avoid wrongly diagnosed foreground regions proposing the hybrid shadow removal method that combines the shadow detection proposed in [HHD99], with mathematical morphology reconstruction, which improves the false negative ratio, although increasing false positive foreground detections.

In a statistical parametric framework, [PT05] proposes a pixel-wise multivariate Gaussian model system. [HHCC03] uses a region model using a statistical parametric method via Spatial and Color Gaussian Shadow Model, and a pixel decision based on threshold comparison, because a foreground model is not available, while [LPAM+09] utilizes a bidimensional Gaussian distribution to model the Brightness and Color distortion of each shadow pixel classifying each pixel using belief propagation.

3.2 Multi-Sensor Foreground Segmentation

Multi-sensor foreground segmentation is becoming an important area in foreground detection, since it allows a better segregation of the foreground objects with respect to the background regions than the one obtained from a single camera sensor. The reason of this improvement in the final results is based on the concept that when using multiple (more than one) sensors to record the scene, the data redundancy

that appears in their combination usually allows a better segmentation in difficult regions that can appear in any camera *i.e.* camouflage, shadows, highlights... Hence, we take a decision based on all the cameras that perform the acquisition setup, thus allowing a better discrimination of the foreground from the background regions.

One of the possibilities when working on these scenarios consists in reconstructing the 3D shapes of the foreground elements that appear in the multi-view sequences captured by the multi-camera setting. This approach appears in opposition to work in an image-based manner, using geometric relationships between pairs of images described by epipolar constraints. Therefore, multi-view approaches can be classified in:

- **Image-based multi-view.** The analysis is performed directly on the captured images and multi-view cues are exploited by considering epipolar constraints that can be computed from fundamental matrices.
- **3-dimensional reconstruction.** Three-dimensional shapes are computed from the multi-view data with several possibilities for their representation, each providing different advantages and drawbacks, as will be seen below.

Both groups can be implemented by using sensors of the same type (*i.e.* color RGB, depth Z...) or combining different ones in a *multimodal analysis* in order to improve the final results by working on different domains of the scene under analysis.

In order to combine the different camera sensors used to record the scene, a process of camera calibration is necessary for each one of the cameras. This calibration will be useful to achieve the 3D-2D correspondence.

In the following, we describe the camera model, which will be used in the rest of the manuscript to get the 3D-2D correspondence between the pixels of the views, and then we review the camera calibration method. Later on, since this thesis deals with multi-view foreground segmentation combining color and depth sensors, and smart-room 3D scenarios, we will explore both areas.

3.2.1 Pinhole Camera Model and Camera Calibration

A camera can be seen as an optical device which performs the projection from the 3-dimensional real world to the 2-dimensional image plane. In a simple model, the camera center is behind the image plane, and 3D points are mapped to 2D where the line joining the camera center and the 3D point meets with the image plane. This model, which is called the *pinhole camera model*, is one of the most common models used in color cameras.

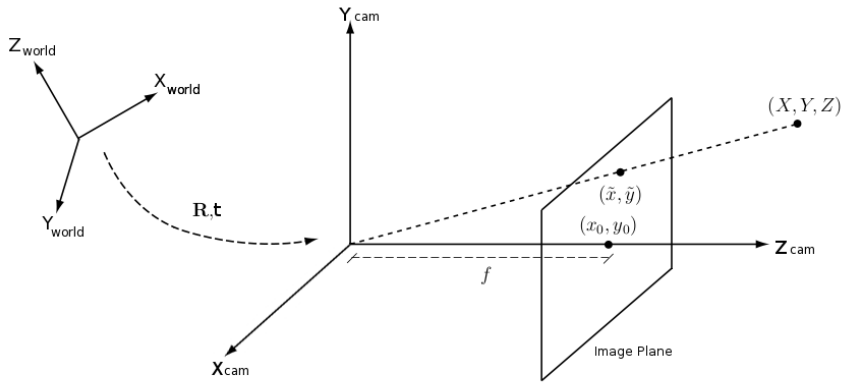


Figure 3.5: Pinhole projection model. A point $s_{3D} = (XYZ)$ in the real world coordinate system $(X_{\text{world}}, Y_{\text{world}}, Z_{\text{world}})$ is first referred to the camera coordinate system $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$ and then projected into the image plane thus resulting in the $s_{2D} = s = (\tilde{x}, \tilde{y})$ pixel coordinates. Focal length is noted as f.

Therefore, the conversion necessary to obtain the correspondence to 2-dimensional coordinates (pixel positions) of the camera images from a 3-dimensional magnitude (a 3D location) is the projection process where one dimension of the 3-dimensional space is lost. Hence, this projection process, transforms 3-dimensional Euclidean coordinates in the world reference frame into 2-dimensional coordinates in the camera reference frame: $\mathbb{R}^3 \rightarrow \mathbb{R}^2$.

Given a certain 3-dimensional point $s_{3D} \in \mathbb{R}^3$ and the 2-dimensional spatial pixel coordinates $s_{2D,i} = s_i \in \mathbb{R}^2$, it is possible to establish the 2D-3D correspondence by means of the *Projection Matrix:* $\mathbf{P}$:

$$s_i = \mathbf{P}_i \; s_{3D} \tag{3.32}$$

From Figure 3.5, we can express the projection model in homogeneous coordinates ($\tilde{x} = \frac{fX}{Z}, \tilde{y} = \frac{fY}{Z}$) in Equation 3.32 as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.33}$$

where f is the focal length. The model may be generalized if the image coordinates are not centered at the intersection of the optical axis with the retinal plane, and if the scaling of each axis is different:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \left(\begin{array}{ccc|c} fm_x & 0 & x_0 & 0 \\ 0 & fm_y & y_0 & 0 \\ 0 & 0 & 1 & 0 \end{array} \right) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.34}$$

where m_x and m_y are the scaling factors of the focal length in each dimension, and x_0 and y_0 are offsets in each dimension. The matrix containing all the information regarding the projective device (i.e. the camera sensor) is usually denoted as the intrinsic parameters matrix $\mathbf{K}$.

Usually, the coordinate system of the real world $(X_{\text{world}}, Y_{\text{world}}, Z_{\text{world}})$ does not coincide with the coordinate system associated with the camera $(X_{\text{cam}}, Y_{\text{cam}}, Z_{\text{cam}})$ thus an affine transformation relating this two systems is required:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \underbrace{\underbrace{\begin{pmatrix} fm_x & 0 & x_0 \\ 0 & fm_y & y_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} [\mathbf{R}|\mathbf{t}]}_{\mathbf{P}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{3.35}$$

where $\mathbf{R}$ and $\mathbf{t}$, are the 3×3 rotation matrix and 3×1 translation vector respectively with respect to the real world coordinate system; and where $\mathbf{P} = \mathbf{K}(\mathbf{R}|\mathbf{t})$ is the camera projection matrix.

Since real lens introduce non linear distortion effects, radial distortion r_d is going to be introduced in the formulation as the most noticeable distortion effect [HZ03]. The radial distortion model is expressed by the following equation:

$$\frac{r_d}{r} = \frac{\tilde{x}_d - x_0}{\tilde{x} - x_0} = \frac{\tilde{y}_d - y_0}{\tilde{y} - y_0}, \tag{3.36}$$

where $(\tilde{x}_d, \tilde{y}_d)$ are the coordinates of a distorted image point.

Since the Taylor series expansion of Equation 3.36 with respect to r is $1 + k_1 r^2 + k_2 r^4 + \cdots$, then $k_1, k_2, \cdots$ are the unique values which are needed to obtain the real image distorted points. Usually, a couple of terms are enough to achieve a good approximation. Hence, the pixel coordinates of the distorted image can be computed as:

$$\tilde{x}_d = x_0 + L(r)(\tilde{x} - x_0), \tag{3.37}$$

$$\tilde{y}_d = y_0 + L(r)(\tilde{y} - y_0), \tag{3.38}$$

$$r = \sqrt{\left(\frac{\tilde{x}_d - x_0}{fm_x}\right)^2 + \left(\frac{\tilde{y}_d - y_0}{fm_y}\right)^2}, \tag{3.39}$$

where r is the radius of distortion and $L(r) = 1 + k_1 r^2 + k_2 r^4$.

3.2.1.1 Camera Calibration

Once we have defined the parameters necessary to characterize the camera sensors, they can be obtained in practice by the calibration process. This process is based on estimating the intrinsic $(\mathbf{K}, k_1, k_2)$ and extrinsic $(\mathbf{R}, \mathbf{t})$ parameters of the camera.

$\mathbf{P}$ has 12 entries, and (ignoring scale) only eleven degrees of freedom in homogeneous coordinates. Hence, it is necessary to have at least 11 equations (i.e., 11 3D/2D pairs of points) to solve $\mathbf{P}$. In practice, more points are used, to minimize a function of the error [HZ03]. All these calibration points may be obtained using special calibration devices, such as a chessboard panel.

3.2.2 Image-Based Multi-View Foreground Segmentation

One of the most extended techniques to improve the foreground segmentation results in a certain scene, consists in combining information of several sensors that are recording it from different positions. In this case, the improvement comes from the different perspective that the camera sensors present. Otherwise, if we combine different type of sensors in a *multi-modal* framework, for instance, combining color *RGB* and depth camera sensors, the improvement will appear thanks to the analysis of different domains. In these applications, to obtain the correspondence of the pixels among views is necessary in order to correctly combine the information between 2D images. In this section, we will focus on the multi-view analysis combining color and depth sensors.

3.2.2.1 Foreground Segmentation Combining Color and Depth Sensors

Color *RGB* and depth sensors work with different technologies that can be used together at the same time without suffering any interference between them, thus presenting non-correlated errors each other:

-Color cameras are based on sensors like CCD or CMOS among others, which allow us a more reliable representation of the scene with high resolution. The segmentation using this kind of sensors results in more precise separation between foreground and background if there are no color camouflage problems between both classes.

- Depth cameras are based on IR transmitter/receiver sensors. Despite new precise sensors based on laser technologies are appearing nowadays, so far, the resultant depth maps obtained using ToF (device that computes Time of Flight using Infra Red light) and kinect (low cost sensor sold by Microsoft that uses structured Infra Red light) are images with lack of precision on the definition of the objects ([KBKL09] gives an in-depth technical analysis of ToF limitations). The segmentation using these devices is a more robust segmentation against color problems, though errors with depth camouflage will be present.

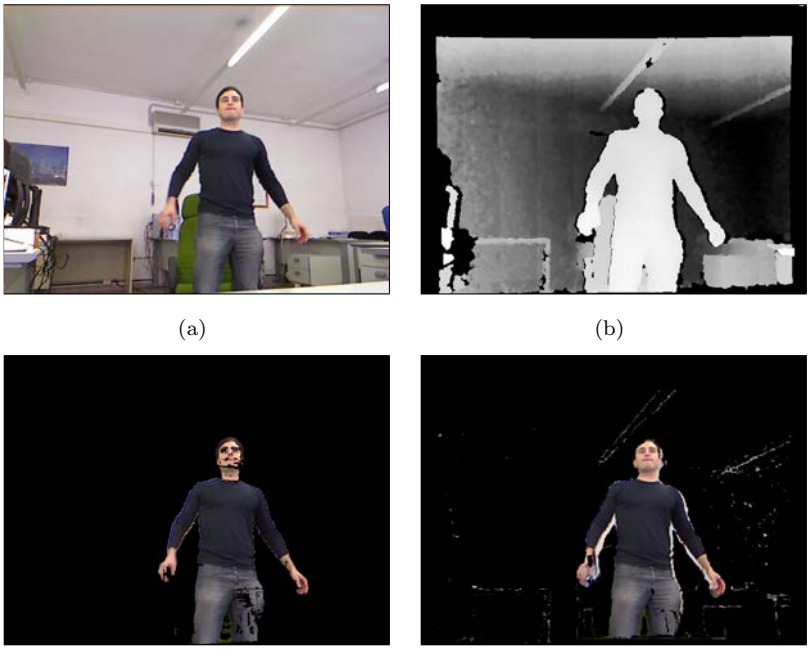
Therefore, a correct combination of these sensors allow to improve the overall performance of the system. For that, both camera sensors must be calibrated and registered projecting the depth map onto the color image, allowing a color-depth pixel correspondence. It should be noted that some problems of miss association can appear due to:

- Camera centers are different and some blind regions appear for each one of the sensors because of the projection process and the parallax computation between cameras.
- The low resolution of the depth measurements produces that several color pixels are associated to only one depth map value.
- The lack of precision of the depth sensor is more pronounced in the borders of the objects, and produces many depth-color association errors in these regions.

Both color and depth sensors can be segmented separately by means of planar foreground segmentation methods (Section 3.1.2). In order to show the limitations of each sensor, Figure 3.6 shows an example of segmentation with a simple exception to background analysis presented in [WADP02] and explained in section 3.1.1.2. We can observe how color segmentation (Figure 3.6(c)) gives us a reduced false positive detections in the segmentation although some false negative errors appear due to the foreground-background color similarity. When using depth segmentation (Figure 3.6(d)), robustness against color similarity is present, but some false positive detections appear in the borders of the object due to the lack of precision of the depth sensor. In the following section some methods of the state of the art which combine the color and depth sensor information are explained in detail.

3.2.2.1.1 Combining Color and Depth Sensors by means of Trimap Analysis

Since depth sensors present an important problem of precision in the borders of the objects, these methods propose to analyze in a different manner the foreground border regions, which are prone to errors, from the rest of the image. Some proposals like [CTPD08, FFK11] are based on this idea. Hence, these methods require



(a) (b)

(c) (d)

Figure 3.6: Foreground segmentation applied to color and depth sequences. From left to right. a) Original color image. b) Original depth image in gray scale. The whiter the pixel, the closer to the camera. c) Color segmentation via exception to background analysis. d) Depth segmentation via exception to background analysis. Black pixels in foreground segmentation correspond to background regions.

the approximate location of the edges of the objects to segment, which are defined by a trimap.

In a trimap, the image under analysis is divided into three different regions: foreground, background and unknown decision, and can be obtained from different processes. A basic approach is explained next:

- An initial foreground segmentation using the depth information is applied based on a thresholding plane defined by the user.
- After that, morphological operations, erosion and dilation, are performed over the foreground silhouette F.
- Final trimap is defined as: Let F be the number of pixels detected as foreground in the previous foreground detection, and B their counterpart in the background, E_F the binary image after erosion and D_F the binary image after dilation. The definitive foreground region is then defined as $T_F = E_F$. The definitive background region is defined as $T_B = B - D_F$ and the uncertainty region as $T_U = D_F \cap E_F$

Figure 3.7 shows an example of the resultant trimap.

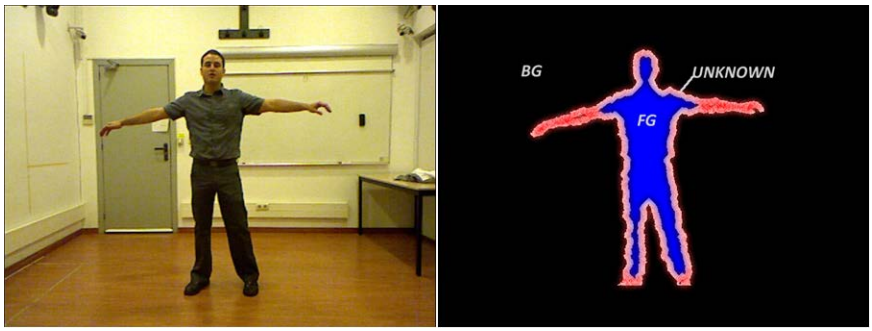


Figure 3.7: Example of trimap segmentation among foreground (fg), background (bg) and unknown regions.

Once the trimap is computed, a special analysis for the foreground segmentation is applied in the uncertainty area to correctly define the foreground and the background pixels.

In [CTPD08], an alpha matting analysis is used. When assigning each pixel to the foreground and background, its depth is compared to the threshold, and it is assigned an alpha value of 1 or 0, which is recorded into what is called the *alpha-matte*. These values are based on each pixel alone. A cross bilateral filter is then applied to the sparse alpha-matte, using the color image as the guide for the range filter (the bilateral filter was introduced to the computer vision field in [TM98] as a method for smoothing grayscale images, we refer the reader to this publication to have an in depth explanation). The idea is to preserve edges by taking a weighted average of local pixels. In this system, the authors filter the alpha values and base these weights on the distance in the grid lattice and the color space.

Hence, the refined estimate for the alpha value of each pixel, A_i, belonging to the trimap uncertainty region $i \in I_{T_U}$ is:

$$A_i = \frac{1}{K_i} \sum_{j \in N, \alpha_j} \alpha_j f(\|i-j\|)\ g(\|c_i - c_j\|), \tag{3.40}$$

where α_j is the alpha value from the alpha-matte, f is the spatial kernel (a Gaussian centered at i), g is the range filter kernel (also a Gaussian), $c_i = RGB$ is the color value of pixel i, N is the neighborhood surrounding pixel I_i and K is a normalizing factor, the sum of the product of filter weights defined as $K_J = \sum_{j \in N, \alpha_j} f(\|i-j\|)\ g(\|c_i - c_j\|)$. The distance between colors is measured as a Euclidean distance.

Other approaches like [FFK11], propose to construct a graph in the uncertainty area T_U in the color domain, and use the graph-cut segmentation technique to classify all pixel in the unknown regions as foreground or background. The workflow of this system is:

1. Create trimap of the image.

2. Using the definitive foreground and definitive background from all trimaps in the batch two Gaussian mixture color models are created, one for the foreground $\mathrm{GMM}_{\mathrm{fg}}$ and one for the background $\mathrm{GMM}_{\mathrm{bg}}$.

3. Create the graph and apply the graph-cut algorithm to classify all pixels in the unknown regions in foreground and background.

4. The color models are updated based on the pixel classification.

5. Steps 3 and 4 are repeated until a maximum number of iterations is reached.

3.2.2.1.2 Combining Color and Depth Sensors Using Probabilistic Models for Depth and Color Data

These approaches propose to create probabilistic models for each one of the sensors. The influence of each sensor to the final labeling decision in foreground or background is evaluated according to the reliability that each camera sensor presents. One example of this kind of segmentation methods is proposed in [SK11].

In [SK11], the authors propose to model the background by means of a GMM in a four dimensional domain based of the color $c = RGB \in \mathbb{R}^3$ and depth $D = d \in \mathbb{R}$ domains. Then, the reliability of each sensor is evaluated for each one of the pixels according to the detection of discontinuities in the depth image. These discontinuities are detected by analyzing the variance in the original depth image $v(x)$. Moreover, normalized color and depth differences $\hat{c}$ and $\hat{d}$ are computed for each pixel as:

$$\hat{d}(x) = \frac{d(x) - d_{\min}}{d_{\max} - d_{\min}} \qquad \hat{c}(x) = \frac{c(x) - c_{\min}}{c_{\max} - c_{\min}}, \tag{3.41}$$

where $d(x) = d_i - d_{\mathrm{bg}}$ and $c(x) = c_i - c_{\mathrm{bg}}$ are the depth and color differences between the input value, and the mean value of the background Gaussian for the pixel i. $c_{\max}$, $c_{\min}$ and $d_{\max}$, $d_{\min}$ are the minimum and maximum color and depth distances of the image under analysis.

The variance in the depth image is also normalized ($\hat{v}(x)$) between 0 and 1 to be comparable to $\hat{d}(x)$ and $\hat{c}(x)$. In areas in which the depth uncertainty is high, the depth measurement will be considered unreliable. Therefore the normalized depth difference $\hat{d}(x)$ is weighted with the uncertainty $\hat{v}(x)$, resulting in an uncertainty filtered depth $\hat{dv}(x)$, which is scaled between zero and one depending on the uncertainty. In contrast to that the color is more reliable if the depth uncertainty is high. Therefore the color weight $\hat{cs}(x)$ is multiplied with the depth uncertainty $\hat{dv}(x)$ and added to the color weight. The result is that if the depth uncertainty is high the

color weight is weighted even higher while at the same time the uncertainty filtered depth is weighted lower. The following equations detail this:

$$\begin{aligned} \hat{dv}(x) &= (1 - \hat{v}(x))\ \hat{d}(x), \\ \hat{cv}(x) &= (1 + \hat{v}(x))\ \hat{c}(x), \\ \hat{s}(x) &= \frac{1}{2}(\hat{dv}(x) + \hat{cv}(x)), \end{aligned} \tag{3.42}$$

where $\hat{s}(x)$ is the resultant combined weighting image.

3.2.3 3-Dimensional Reconstruction

When we are interested in obtaining the final 3-dimensional volumetric representation of one foreground object, the more convenient strategy consists in using multiple cameras sensors surrounding the object under study. This will allow us to achieve enough information belonging to the foreground object, from all the points of view, to correctly define the foreground object in the 3-dimensional space. Such reconstruction will be more precise the more cameras observe the space where the scene is located. In this way, it is necessary to achieve a correct calibration and synchronization of the camera sensors to appropriately process the 2-dimensional data flows recorded by each one of the sensors. Figure 3.8 displays an example of a smart room, which is a common set-up used to record the multi-view sequences.

There are different approaches to obtain the volumetric reconstruction of the foreground object in the literature. In all of them it is usually assumed that the scene of interest is inside the convex hull of the cameras, *i.e.*, it is visible by all the cameras. From least to most accurate, the volumetric estimates are:

- **Convex Hull** (CH) The Convex Hull of an object in the 3D space is the intersection of all the convex sets containing all the points of the object. In the three-dimensional space, the Convex Hull is a convex polyhedron. Given a number of 3D points, there have been several implementations to obtain the Convex Hull. A review of some of these techniques can be found in [FS77] and another proposal taking care of the technical aspects of a practical implementation can be found in [Day90].

- **Visual Hull** (VH) A more refined object estimate is the Visual Hull [BL03, Lau91, Lau94, Lau95]. The Visual Hull is obtained with the technique known as Shape from Silhouette: For each frame of the muti-view sequence and each one of the sensor frames, the foreground object is segmented, thus obtaining binary foreground masks with the silhouette of the object of interest for each

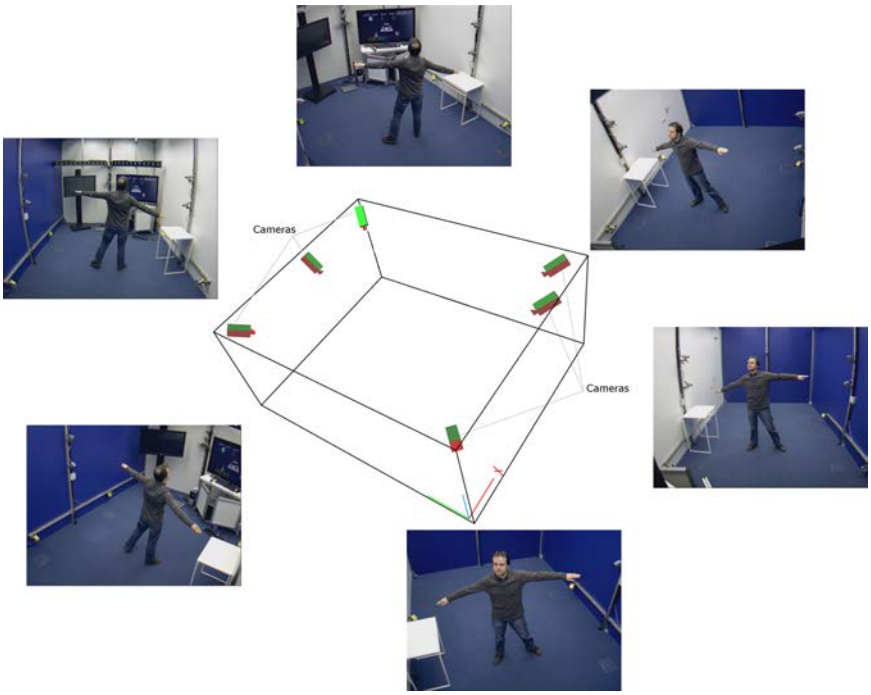


Figure 3.8: Example of smart-room setup. The cameras are placed surrounding the object of interest so that, we achieve information of the object from all the points of view.

view. The volume estimate known as Visual Hull is obtained as the maximal volume which could explain the observed silhouettes.

- **Photo Hull** (PH) It is the most accurate estimate of the real object. When instead of binary silhouettes, color images captured by multi-camera settings are used for reconstructing a scene, the photo hull is obtained. The process is performed as a photo-consistency test of visible volumetric points with respect to each image. The Photo Hull is defined as the maximum volume that is photo-consistent, and Voxel coloring [SD97], Space Carving [KS00b] and Energy minimization [SP05] are the methods used to obtain it.

If we assume that the different volume estimates, obtained by means of each method, are free of errors, we can define a precision chain of the 3D reconstruction of a real object volume Ψ as:

$$\Psi \subseteq \mathrm{PH}(\Psi) \subseteq \mathrm{VH}(\Psi) \subseteq \mathrm{CH}(\Psi), \tag{3.43}$$

Figure 3.9 shows a comparison between visual hull and photo hull. As we can observe, both reconstruction methods have problems to reconstruct the exact shape of the object due to the limitations of precision that the acquisition setup impose. In spite of this, using the color information in the reconstruction process (in PH), helps to improve the precision of the resultant volume.

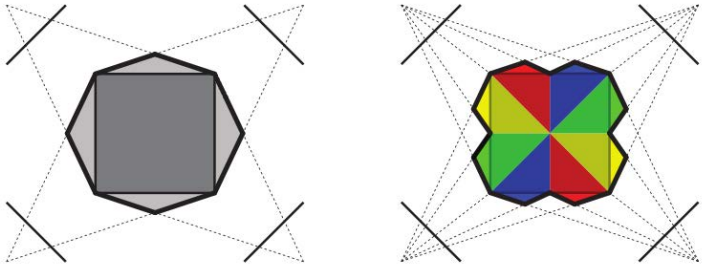


Figure 3.9: Tightness of the photo hull compared to the visual hull. On the left, the bold line represents the visual hull of a 2D scene (in this case, the central occupied square) reconstructed from a set of silhouettes (segments) available in a set of 1D cameras. On the right, the photo hull is computed and represented by the bold line, resulting in a tighter reconstruction of the actual shape, which is the colored square.

The choice of one method or another depends on the application and the computational load that can be processed. Although Photo Hull obtains the most accurate volume estimate, it is not suitable for real-time operation. On the contrary, Visual Hull can be computed by using Shape from silhouette techniques, which are suitable for real-time processing while maintaining a correct precision of the volume. The only requirement is the necessity of computing the foreground segmentation of each one of the views, which can be obtained using the techniques presented in Section 3.1.

This thesis utilizes Shape from Silhouette (SfS) techniques in foreground detection methods for multi-view scenario. Therefore, in the following section, SfS reconstruction is explained in detail.

3.2.3.1 Shape from Silhouette

3-dimensional reconstruction based on SfS approach presents two main steps:

- Foreground segmentation in each one of the views in order to obtain the foreground silhouettes of the object that we want to reconstruct.
- Intersection test. It is the main step of the SfS. Each point in the 2-dimensional foreground silhouettes defines a ray in the 3D space that intersects the foreground volume somewhere along the ray. The union of all the visual rays for all the foreground points defines a conic ray where the entity is guaranteed to lie. In SfS, the intersection of the visual cones associated with a set of cameras defines the volume in which the object is guaranteed to lie. Figure 3.10 shows an example of the intersection of the conic rays, while in Figure 3.11 a graphical representation of the SfS operation is displayed. We can see how SFS-based algorithms are not able to perform an accurate reconstruction of

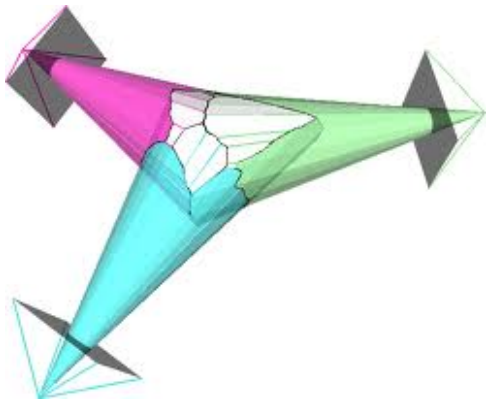


Figure 3.10: Example of visual hull with three views. The visual cones intersect further constraining the volume estimates of the foreground object.

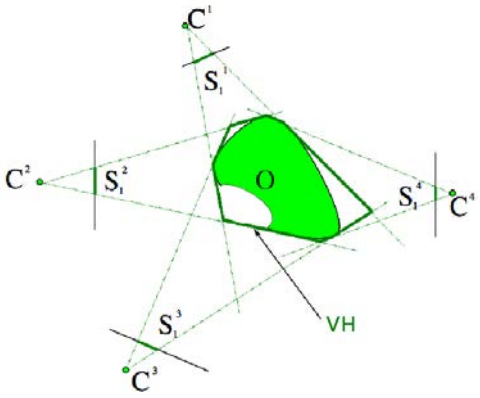


Figure 3.11: Example of visual hull with four camera views C reconstructing a concave object O from the silhouettes segmented in each view S^c. The resultant Visual Hull VH can not represent the concavities of the objects.

concave objects, if the concavity shape is not detected by any camera sensor.

One of the main approaches to obtain the intersected volume is the voxel-based SfS. Next section is devoted to explain it.

3.2.3.1.1 Voxelized Shape from Silhouette These techniques divide the space into voxels, which is the volume elements representing values in the 3-dimensional space (the pixel equivalents for 3D volume data) [CKBH00, LP06b, LP06a, MKKJ96, MTG97]. Then, the system projects each voxel to the views belonging to the acquisition setup, in order to detect if they are contained in every silhouette. This process is carried out by using a projection test.

There are many possible projection tests. Some are faster, and others more robust to noise. A simple Projection Test is the One Pixel Projection Test, which is passed if the pixel corresponding to the projection of the center of the voxel belongs

to a silhouette. Algorithm 2 shows the voxelized SfS considering any projection test.

Algorithm 2 Voxelized SfS

Require: : Silhouettes: S(c), Projection Test: PT (voxel, silhouette)

```
1: for all voxel do
2:     voxel ← Foreground
3:     for all cameras do
4:         if PT (voxel, S(c)) is false then
5:             voxel ← Background
6:         end if
7:     end for
8: end for
```

As we can observe in Algorithm 2, Visual Hull is highly dependent on the 2D foreground segmentation that is performed in each view. Just one false negative error in one of the views is propagated to the 3-dimensional reconstruction, resulting a false negative error in all the voxels projecting to false negative pixels.

Hence, the concept of Visual Hull (VH) is strongly linked to the one of silhouettes' consistency: A set of silhouettes is consistent if there exists at least one volume which exactly explains the complete set of silhouettes, and the VH is the maximal volume among the possible ones. If the silhouettes are not consistent, then it does not exist an object silhouette-equivalent, so that the VH does not exist.

Total consistency hardly ever happens in realistic scenarios due to inaccurate calibration or wrong silhouettes caused by errors during the 2D detection process. Because of that, some SfS methods have been designed in the past assuming that the silhouettes can not be consistent, thus adding a tolerance to error (τ) in the number of views necessary to consider a voxel as occupied. Hence, adding error tolerance to the 3-dimensional reconstruction, the estimate of the visual hull is conservative in the sense of assuming that τ foreground under-segmentation errors can occur.

Considering the tolerance to errors τ as the maximum number of cameras that can detect background in the projection test and still consider the voxel as foreground in the reconstruction process, the 3-dimensional reconstruction algorithm is modified as appears in Algorithm 3, where num_bg is the number of projection tests detecting background.

This approach will lead to reduce the number of false negative errors although losing precision in the final reconstructed volume. Figure 3.12 shows an example of this effect, where one dancer in a 8-cam smart-room sequence is reconstructed using the Visual Hull method (without tolerance) and the conservative Visual Hull with tolerance $\tau = 2$. As we can appreciate, normal VH presents some false neg-

Algorithm 3 Voxelized SfS with tolorenace to errors

Require: : Silhouettes: S(c), Projection Test: PT (voxel, silhouette), Tolerance: τ

```
1: for all voxel do
2:     voxel ← Foreground
3:     num_bg ← 0
4:     for all cameras do
5:         if PT (voxel, S(c)) is false then
6:             num_bg ← num_bg +1
7:         end if
8:         if num_bg > τ then
9:             voxel ← Background
10:            break
11:        end if
12:    end for
13: end for
```

ative errors in the resultant volume, since just one false negative error in the 2D foreground detection of one of the views, is propagated to the final volume. When using conservative VH with $\tau = 2$, the resultant volume estimates presents less false negative errors, since for each pixel, we are allowing up to two background detections in the silhouettes to decide that the voxel is occupied. The drawback of conservative Visual Hull is that both true background detections and false negative errors are treated the same way thus increasing the false positives errors in the resultant volume, since inconsistencies in voxel occupancies increase drastically, and consequently, the precision of the volume is reduced.

3.2.4 Multi-view Cooperative Foreground Segmentation Using 3-dimensional Reconstruction Methods

Classical 3-dimensional reconstruction methods are based on the Visual Hull computed with the foreground segmentation masks obtained in a separated step for each view. Similarly to the octree-based voxelization, [EBBN05] uses a finer resolution in those regions where it is needed, accompanied by a post-processing aiming at obtaining crack-free polygonal surfaces, using marching cubes [LC87].

Many authors have been working in 3-dimensional reconstruction techniques that deal with the inconsistency of the silhouettes proposing SfS techniques with enhance robustness. In these proposals, consistency tests between views and further processing is applied in order to overcome the limitations in the silhouette extraction.

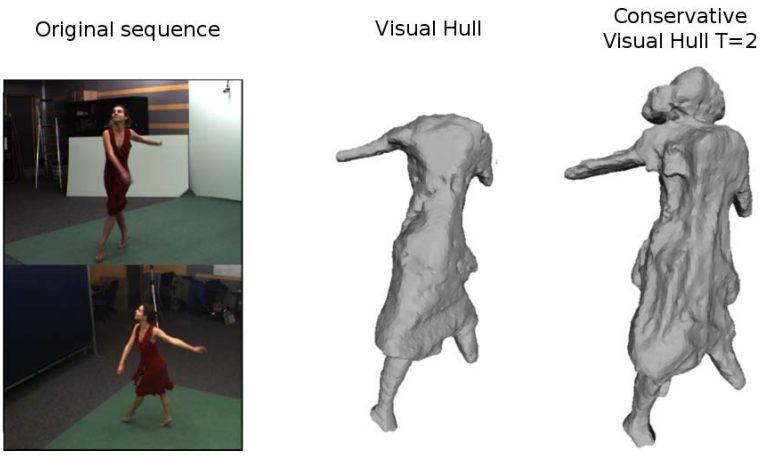


Figure 3.12: SfS of an 8-cam smart-room sequence. First column displays two of the eight-view frames sequence. Second row shows the Visual Hull reconstruction without considering any tolerance to errors. Third column depicts the conservative Visual Hull with $\tau = 2$.

3.2.4.1 Correcting Shape from Silhouette Inconsistencies

The method proposed by [LP06a] is an example of this kind of techniques. It detects inconsistencies in 2D silhouettes regions that can be detected by reconstructing the VH using SfS methods and projecting it back to examine how the projections match with the generative silhouettes. Then the shape can be reconstructed using a different criterion when there are parts of the volume (Inconsistent Hull:IH) which project to inconsistent regions in the silhouettes (Inconsistent Silhouettes:IS).

The IH is introduced as the volume where does not exist a shape which could possibly explain the observed silhouettes. In order to estimate the IH, we need to determine the unions of the inconsistent cones, similarly as SfS methods determine the inter- sections of the visual cones. The concept of Shape from Inconsistent Silhouette is introduced by using a voxel-based approach. The detailed process for the IH voxelization is shown in Algorithm 4.

Therefore, IH contains all the volumetric points which cannot explain the silhouettes where they project. In terms of consistency, these points are candidates of not having been classified as Shape by error, while all the points in the VH are error-free.

3.2.4.2 Fusion 2D Probability Maps for 3D Reconstruction

There are several approaches of the literature where the final 3D reconstruction is obtained by fusing not only the silhouettes of the foreground objects, but also the probabilities that each pixel presents to belong to the background and the

Algorithm 4 Voxelization of the IH

Require: : Silhouettes: S(c), Projection Test: PT (voxel, silhouette),

for all voxel **do**
    VH(voxel) ← true
    for all cameras **do**
        if PT (voxel, S(c)) is false **then**
            VH(voxel) ← false
        end if
    end for
end for
Project the VH to all the camera views: $\mathrm{VH}_{\mathrm{proj}}(c)$
for all voxel **do**
    IH(voxel) ← false
    for all cameras so that PT(voxel,S(c)) is true **do**
        if PT (voxel, S(c)) $\neq$ PT (voxel, $\mathrm{VH}_{\mathrm{proj}}(c)$) **then**
            IH(voxel) ← true
        end if
    end for
end for

foreground classes, in the case of having a 2D probabilistic framework. In [FB05] the authors propose a reference of the multi-view probability fusion.

In the paper ([FB05]), the authors use a space occupancy grid as a probabilistic 3D representation of scene contents, while considering each camera as a statistical occupancy sensor. The goal of this framework is to infer the voxel occupancy V in the position $S_{3D} = XYZ$: $V_{S_{3D}}$, given the set of input images from the camera sensors I', the background models defined for each camera B, the foreground silhouettes for each sensor F and the prior knowledge introduced to the model τ. Where $V_{S_{3D}} \in \{\mathrm{fg}, \mathrm{bg}\}$ fg $\equiv 1$ and bg $\equiv 0$ (free or occupied) respectively.

Modeling the relationships between the variables requires computing the joint probability of these variables, $P(V_{S_{3D}}, I', B, F, \tau)$ based on the statistical dependencies expressed in Figure 3.13:

$$P(V_{S_{3D}}, I', B, F, \tau) = P(\tau)P(B|\tau)P(V_{S_{3D}}|\tau)P(F|V_{S_{3D}}, \tau)P(I'|F, B, \tau), \quad (3.44)$$

Each component is defined as:

- $P(\tau)$, $P(B|\tau)$ are the prior probabilities of the parameter and of background image parameters. These terms are set to uniform distribution since there is not *a priori* reason to favor any parameter values.

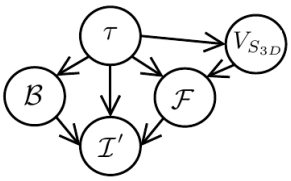


Figure 3.13: Dependency graph of the variables of the system. I': observed images. B: background models. F foreground silhouettes. τ prior knowledge introduced to the model. $V_{S_{3D}}$: occupancy at voxel S_{3D}.

- $P(V_{S_{3D}}|\tau)$ is the prior likelihood for occupancy. This term is also set to uniform distribution, since the authors choose not to favor any voxel location.
- $P(F|V_{S_{3D}}, \tau)$ is the silhouette likelihood term. The dependencies considered reflect that voxel occupancy in the scene explains object detection in images.
- $P(I'|F, B, \tau)$ is the image likelihood term. Image colors are conditioned by object detections in images, and the knowledge of the background color model.

Equation 3.44 can be developed considering a pixel-wise background model based on a Gaussian distribution in the $c = RGB$ domain, and a uniformly distributed foreground model with no further assumptions of objects of interests. Once the joint probability distribution has been fully determined, it is possible to use the Bayes' rule to infer the probability distributions of the variable $V_{S_{3D}}$ given the value of Known variables I', B, τ and marginalizing over unknown variable F:

$$\begin{aligned} P(V_{S_{3D}}|I', B, \tau) &= \frac{\sum_F P(V_{S_{3D}}, I', B, F, \tau)}{\sum_{V_{S_{3D}}, F} P(V_{S_{3D}}, I', B, F, \tau)} = \\ &= \frac{\prod_{v,i} \sum_{F_i^v} P(F_i^v|V_{S_{3D}}, \tau) P(I_i^v|F_i^v, B_i^v \tau)}{\sum_{V_{S_{3D}}} \prod_{v,i} \sum_{F_i^v} P(F_i^v|V_{S_{3D}}, \tau) P(I_i^v|F_i^v, B_i^v \tau)}, \end{aligned} \tag{3.45}$$

where v stands for the view under analysis and i is the index of pixel under analysis.

The final expression (Equation 3.45) relates the voxel occupancy to all the pixel observations. In practice, the inference product can then be computed over $k \times k$ window of pixels centered at the image projection of voxel S_{3D} in each image.

3.3 Conclusions

In this chapter we have reviewed the main techniques of the state of the art that use the foreground segmentation process as a central step to achieve correct application results. The references have been presented according to the acquisition setup

utilized to record the scene under analysis as well as the application under study. As we have seen, the strategy to follow to segment the foreground objects from the scene, is not unique, and must be designed according to the characteristics of the scenario, in order to avoid the influence of shadows, highlights and the so common camouflage problem between foreground and background. Moreover, we have also observed that redundancy between cameras can be applied to foreground segmentation in multi-view scenarios, and can be useful to reduce the detection errors that appear in some views.

In the following chapters we will explain the proposals developed in this thesis, referencing this chapter when commenting some aspects and strategies of the state of the art.

Part I

Proposals. Foreground Segmentation in 2D Planar Sequences

When considering a sequence where the foreground objects are close to the camera sensor so that the color and shape of the object is clear enough to characterize it along the frames, we can design a foreground segmentation framework which exploits all this information by means of probabilistic modeling. Hence, according to the characteristics of the sequence, a Bayesian classification process can be defined if we are able to correctly model the foreground (fg) and background (bg) data of the sequence in order to establish a probabilistic processing.

As we have seen in the previous chapters, when we design a foreground segmentation system for a certain sequence under study, the characteristics of the scenario will determine the strategy to follow in order to reduce the false positive and false negative detections. In this line, previous work has shown us that pixel-wise models present more accuracy to model static regions, like the background in static cameras sequences, since they allow us to represent them at pixel resolution. On the other hand are the region-based models, which although modeling the regions with less precision than the pixel-wise models, are more appropriate for modeling non-static regions, since this kind of models adapts better to the motion changes that appear, for example, in foreground objects, or in moving background scenarios.

Therefore, these principles are the bases of the proposals that we are going to explain in this part of the dissertation, devoted to introduce two foreground segmentation techniques suitable for 2-dimensional planar scenarios. We first explain the foreground segmentation system that we have designed for static camera sequences, which combines in a Bayesian framework pixel-wise background modeling with region-based foreground and shadow models. Next, we present a foreground segmentation technique appropriate for moving camera sequences, which applies the bases of the first approach to achieve a Bayesian classification between foreground and background region-based models, in order to obtain a robust foreground detection system for these complicated scenarios.

Chapter 4

Bayesian Foreground Segmentation in Static Sequences

4.1 Introduction

In this chapter we present a foreground segmentation system for monocular static camera sequences and indoor scenarios that achieves correct foreground segmentation results also in those complicated scenes where foreground and background present similar color. In this system, we propose to combine pixel-wise probabilistic modeling for the background class, with region-based foreground and shadow probabilistic models, thus taking the most of each one to improve the foreground segmentation results.

As we have seen in previous sections, pixel-wise modeling gives a precise representation of the static background but it cannot be used to characterize moving regions like the ones belonging to the foreground or shadow classes, since both are constantly changing and moving along the scene and a probabilistic model at a pixel-wise level is difficult to build and update. For them, region-based models are the best option to achieve its probabilistic representation because they allow us to obtain a correct adaptation to the shapes and new regions that can appear along the sequence, while maintaining the performance of the probabilistic modeling.

Knowing this, this approach has to deal with two main aspects:

- Combine the region-based foreground and shadow models with the pixel-wise background model in order to achieve a correct classification of the pixels of

the image in foreground, background and shadow classes.

- Since we are going to use region-based probabilistic models to characterize the foreground objects and their casts shadow regions, we need a logic system to correctly deal with the foreground objects and their associated models like, for example: accept a foreground detection as a foreground object, create the foreground and shadow models of the object, remove objects that disappear from the scene, etc.

The foreground segmentation system presented in this chapter solves both aspects by following a three steps work-flow: An initial foreground detection performs a simple segmentation via Gaussian pixel color modeling and shadows removal. Next, a tracking step uses the foreground segmentation for identifying the objects, and tracks them using a modified Mean Shift algorithm [GPL08, CR03]. At the end, an enhanced foreground segmentation step is formulated into a Bayesian Maximum a Posteriori - Markov Random Fields (MAP-MRF) framework, which combines the parametric models defined for each one of the classes.

This proposal is explained in detail in the following sections.

4.1.1 State of the Art

A brief overview of foreground segmentation techniques devoted to planar static scenarios is presented in this section, with the objective to extend the survey presented in Section 3.1 and establish the context for the development presented in this Chapter.

4.1.1.1 Techniques Based on Background Modeling

Over the recent years there have been extensive research activities in proposing new ideas, solutions and systems for robust object segmentation and tracking to address the foreground segmentation in indoor static sequences. Most of them adopt the background subtraction as a common approach for detecting foreground moving pixels, whereby the background scene structures are modeled pixel-wise by various statistically-based learning techniques on features such as intensities, colours, edges, textures, etc. A pixel is classified as background when its value is not correctly modeled by the background model, in the so called exception to background analysis. The models employed include mono-modal Gaussians ([JDWR00]), Gaussian Mixture Model (GMM) ([SG00]), nonparametric Kernel density estimation ([EHD00]), or simply temporal median filtering ([ZA01, LV02]).

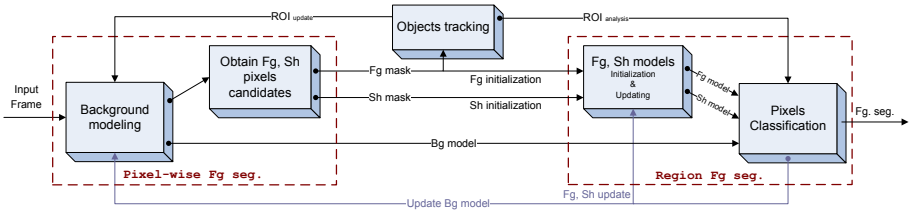


Figure 4.1: Work Flow. Bg stands for background, Fg for foreground and Sh for shadow.

Optionally, shadow removal techniques can be incorporated in the background subtraction/modeling step to improve the segmentation removing the false positives detections that illumination problems produce. [PMTC03] have presented an in-depth survey of these algorithms while [XLP05] propose the hybrid shadow removal method that we have used as the initial step for shadow modeling.

After foreground detection, a connected component analysis (CCA) is usually performed in order to cluster and label the foreground pixels into meaningful object blobs, from which some inherent appearance and motion features can be extracted. Finally, there is a blob-based tracking process aiming to find persistent blob correspondences between consecutive frames. Several authors employ this kind of solution ([PT03, GVPG03, CHB+05, XLL04]).

The main problem of these algorithms is that false negatives appear when foreground and background present color similarities. False positives can also be observed when an external agent modifies the configuration of the scene (illumination changes, shadow effects or spatial alterations of the background objects configuration). The trade-off between false positive and false negative detections, makes it difficult to solve this problem using only the techniques explained above. Furthermore, none of these proposals uses feedback between the foreground detection and the tracking process, to improve the updating of the models in order to avoid the propagation of wrong detections along the sequence.

4.1.1.2 Techniques Based on Foreground Modeling

Background subtraction techniques only require the construction of a background model. However, if a foreground model is available, a Bayesian approach for foreground segmentation and tracking can be performed with the objective to improve the segmentation of the foreground object. In order to create the models, an initial segmentation is usually performed using an exception to background method, and once there is sufficient evidence that the foreground entities are in the scene, foreground models are created.

Several foreground models have been proposed in the past for different purposes including the foreground segmentation task ([KS00a, MD03, LHGT04]), or object and person trackers where the foreground has been previously segmented ([MRG99, EHD00]). As with background models, foreground models are Gaussian-based in most of the cases. Different alternatives are: single-Gaussians ([WADP02]), GMM ([MRG99, KS00a]), and nonparametric models with Gaussian kernels ([MD03, SS05]). In [KS00a] people are first segmented with the exception to background approach and tracked by segmenting them into classes of similar color (initialized by Expectation Maximization (EM) ([DLR$^+$77])). Each pixel is assigned in the following frames to the class that maximizes the probability of the pixel to belong to that class (including a class for the background). Means and variances of the classes are updated after classification. However, the partition of the object in regions modeled by independent Gaussians is too rigid and prone to errors. The work in [MRG99] uses a GMM to model the color distribution of the objects to track and EM to update its distribution. Since the objective is to track a single object, a background model is not used and thus a complete segmentation is not achieved. In the proposal presented in Section 3.1.2.3 ([YZC$^+$07]), a GMM for modeling both the foreground and background, in spatial and color domains, is used. The models are first initialized using a reference frame and the background and foreground models are adjusted using the EM algorithm. This kind of algorithms, with iterative processes, present a high computational cost that doesn't allow a real time sequence analysis. Moreover, in case of a complex background, and even using a GMM with a very high number of Gaussians, the foreground can occupy background regions of similar color which become close to its position as the object moves along the scene.

In the literature, none of these systems propose to combine this approach with tracking methods, because it is assumed that foreground modeling allows a good segmentation and tracking for itself. However, as it has been said above, there is certain difficulty to correctly maintain a good foreground model in some scenarios where foreground and background present color similarities.

Moreover, a specific model for the shadow of each object can be constructed using the tracking information and an initial shadow detection. This allows to make the foreground/background segmentation within a Bayesian framework, using a background model and specific foreground and shadow models for each object and its shadow.

The segmentation system that we propose, combining foreground detection with an object tracking algorithm, follows the work flow of Figure 4.1. It consists of three main blocks: Pixel-wise Foreground Segmentation, Objects Tracking, and Foreground segmentation based on Spatial-Color models.

4.1.2 Proposed Method

We propose a system that runs as a complex implementation of the simple concept of surveillance: be aware for external changes, detect and track objects and refine the object detection improving the knowledge about it and focusing the attention in its region.

In this way, the goal of this system consists in taking the most of each block, using the information available to facilitate the updating of the models and processes. Hence, according to Figure 4.1 main blocks of the system are:

- **Pixel-wise Foreground Segmentation:** This initial step is used as a first glance at the foreground objects that appear in the scene. It also segments shadow pixels to create a shadow model for each detected object.

- **Objects Tracking:** It is used to detect and track those objects that appear in the scene, matching the blobs detected in the first segmentation with the objects that are being tracked. It assigns the detected blobs to objects with a label that characterizes them along the time and brings us the valuable spatial information about the position and size of the object in the scene. A Region Of Interest (ROI) is obtained for each object to track, and it is used for appropriate background and foreground models updating and for associating, in the next step, each foreground model with its corresponding object. The method proposed uses a classical Mean-Shift tracking method with the following improvements: several connected components association to each object (it avoids false positive detections when an object is segmented in several connected components), detection and solving of objects occlusion (analyzing the connected components detected in each frame), focus the position estimation in those regions that belong to the foreground and incorporation to the background of all the foreground detections, not belonging to the objects in analysis, which appear outside the defined ROI.

- **Foreground segmentation based on Spatial-Color model:** Here a final enhanced foreground segmentation of each object is obtained, combining in a Bayesian framework spatial-color models of the foreground and shadows regions with the pixel-wise color model of the background. The foreground and shadow models are obtained using preliminary shadows and foreground masks, the position of each object, and the background model, all obtained in the previous two steps. The novelty of this approach resides in the combination of a pixel-wise background model with foreground and shadow spatial models within a MAP-MRF framework. We associate a spatial-color GMM foreground and shadow models to each object that is being tracked in the scene, assuming that the shadow effect that each object produces is an attribute

of the object that produces the background color change. Novel updating techniques for spatial-color GMM are also proposed in the color and spatial domains, for correct evolution of the models along the scene, thus achieving a precise final foreground segmentation.

These processes will be explained in the following sections:

Section 4.2 describes the initial pixel-wise foreground segmentation via color Gaussian modeling. The Modified Mean-Shift based tracking system is explained in Section 4.3. Section 4.4 is devoted to the enhanced foreground segmentation proposed method, focusing on the foreground, shadow and background probabilistic models and the final pixel classification. Finally, some results and conclusions are presented in Section 4.5 and Section 4.6 respectively.

4.2 Initial Pixel-Wise Foreground Segmentation

As we can observe in Figure 4.1, an initial foreground segmentation is performed in the first block. The aim of this initial segmentation is to obtain an initial estimation of the foreground and shadow regions and a robust background model that is going to be used for classification in the last foreground segmentation block of the system. This initial foreground constitutes the input of the tracking system, and will be used to initialize the foreground model for each object. The shadow pixel candidates will also be used for creating or updating the shadow model of each object.

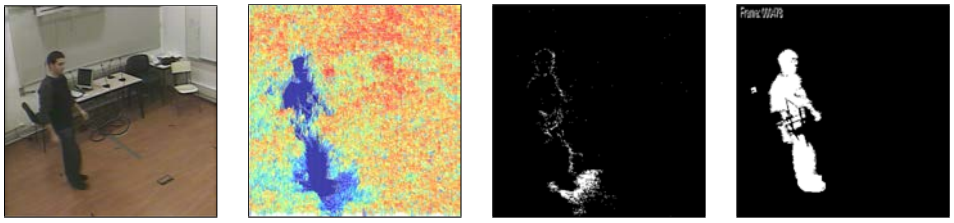
4.2.1 Background Model

For static backgrounds applications, a precise pixel model can be learned. Although more complex models for each pixel could be used, a Gaussian distribution in the RGB color space has proved to work efficiently in most of the considered scenarios [WADP02].

$$P(c_i|\text{bg}) = \frac{1}{(2\pi)^{3/2}|\Sigma_{c,i}|^{1/2}} \quad \exp\left[-\frac{1}{2}(c_i - \mu_{c,i})^T \Sigma_{c,i}^{-1}(c_i - \mu_{c,i})\right], \tag{4.1}$$

where $c_i \in \mathbb{R}^3$ is the i-th input pixel's value ($i = 1, ..., N$) in the $c = RGB$ domain, $\mu_{c,i} \in \mathbb{R}^3$ is the pixel mean value, $\Sigma_{c,i} \in \mathbb{R}^{3\times3}$ is the covariance matrix and $|\Sigma_{c,i}|$ is its determinant. We first initialize each background Gaussian ($\mu_{c,i}$ and $\Sigma_{c,i}$) with initial training values learned from a set of frames with no foreground. To simplify the model, we assume non-correlated components (see Appendix A.2).

As we can observe in Figure 4.1, this model is updated at each frame with the segmentation mask obtained after the final pixel classification. Background pixels



(a) (b) (c) (d)

Figure 4.2: From left to right. a) Original image. b) Probabilistic image with $P(c_i|\text{bg})$. Red color denotes high probability, blue color denotes low probability. c) Shadow mask obtained via Hybrid Shadow removal method [XLP05]. d) Foreground mask obtained via pixel-wise method [XLP05]

are updated in order to adapt the model to progressive image variations, according to the following equations:

$$\begin{aligned} \mu_{i,j,t} &= (1-\rho)\mu_{i,j,t-1} + \rho c_{i,j,t}, \\ \sigma^2_{i,j,t} &= (1-\rho)\sigma^2_{i,j,t-1} + \rho(c_{i,j,t} - \mu_{i,j,t})^2. \end{aligned} \tag{4.2}$$

Where j denotes the RGB color component, ρ is the update rate $0 \leq \rho \leq 1$. The resulting background model is used in the next step to obtain the foreground and shadow candidates, and it is also used in the final Bayesian classification step.

Figure 4.2(b) depicts the background probabilistic image of a certain frame of a smart room sequence. Pixels with high probability are depicted in red colors and those with low probabilities with blue colors. We can realize that the region occupied by the foreground object presents a low probability of being background. Regions with shadows present also low probability and are those that produce false foreground detections. Regions with foreground-background color similarity present high background probability, and are those that produce false background detections.

4.2.2 Selection of Foreground and Shadow Candidates

In order to obtain probabilistic shadow and foreground models, we look for a group of pixel candidates to initialize and update each one of the models before the final pixel classification. Foreground candidates are used to initialize the foreground model (the updating is done with the final pixel classification), and shadow candidates are used to initialize and also update the shadow model.

The pixel-wise background model of the previous step is used to obtain the initial group of foreground pixel candidates by means of exception to background analysis. The $s_i = XY \in \mathbb{R}^2$ pixel spatial information, and the $c = RGB$ color value $c_i \in \mathbb{R}^3$ of these foreground candidates are used next to find the shadow candidates.

Foreground pixel candidates through exception to background

A pixel is classified as foreground candidate if it doesn't match the background model. This classification is done according to the following equation ([WADP02]):

$$\|c_{i,j,t} - \mu_{i,j,t}\|^2 \quad > \quad k^2\sigma^2_{i,j,t}, \tag{4.3}$$

where $\|\cdot\|$ is the Euclidian distance, i stands for the pixel under analysis, j denotes the RGB color component and k is the decision constant (usually fixed to 2.5).

Color based shadow candidates detection

A shadow is normally an area that is only partially irradiated or illuminated because of the interception of radiation by an opaque object between the area and the source of radiation. To assist the classification into foreground or shadow, the Brightness Distortion (BD) and Color Distortion (CD) of each pixel are analyzed according to the shadow removal method explained in the Section 3.1.3. Hence, a set of thresholds on the Brightnes Distortion (BD) and Color Distortion (CD), as defined in [XLP05] are applied. With this procedure we obtain an initial classification of the pixels in foreground, background and shadow. The resulting shadows and foreground masks are obtained with those pixels belonging to each class, respectively. Figure 4.2(c) and 4.2(d) show examples of shadow segmentation mask and foreground mask after shadow removal process.

4.3 Modified Mean-Shift Based Tracking System

This block of the system is in charge of managing the objects (detect and remove) and obtaining certain ROIs for pixel-wise and spatial-color foreground segmentation blocks:

- $\text{ROI}_{\text{analysis}}$ to limit the next foreground segmentation into a specific region for each object.
- $\text{ROI}_{\text{update}}$ for background model feedback.

To obtain these ROIs, we propose to use the tracking algorithm presented in [GPL08]. This algorithm tracks the objects of the scene matching detected foreground blobs and tracked objects using a modified Mean Shift tracking algorithm ([CR03]). The main modifications are the following: the foreground segmentation from the previous block is used into the Mean Shift algorithm and the association of several blobs to an object is allowed.

The necessary inputs for this system are the original image of the sequence we are analyzing, and the foreground segmentation mask obtained in the previous

block. This exception to background segmentation is suitable for this tracking system, despite it presents a high number of false negatives when color similarities between background and foreground appear. However, it allows a high speed segmentation and reduces the false positives. This guarantees that the initialization of the foreground model (explained in Section 4.4) will not model background or shadow regions and that the $\mathrm{ROI}_{\mathrm{analysis}}$ will be the minimum area needed to enclose the object.

The main steps of this algorithm are:

Foreground objects detection. The foreground mask produced in the pixel-wise foreground segmentation is filtered with a morphological area opening in order to eliminate those connected components (denoted by CC) with an area smaller than a given threshold. This threshold depends on the size of the objects we want to detect. In order to label the connected components as objects, a correspondence between the detected CCs at time t and the objects at time $t-1$ is needed. Hence, a register for the objects ($\Theta = \{\theta_{j,t}\}_{j=1,\ldots,\#\mathrm{objects}}$) that maintains the updated information for any detected Object (centroid position, size, color histogram and counter of appearance and occlusion state) is used. Those CC that have not been associated to any Objects, and have been tracked for a period of time previously defined, are introduced in the corresponding registers as new Objects. Let us note that an object might not be correctly detected by the simple exception to background algorithm due to its similarity with the foreground (for instance, if the detected size is smaller than the area threshold applied to the CC's). However, the object can be recovered in the successive frames if it moves into a different area of the scene, because the detection of new objects is continuously applied at each frame.

Mean shift tracking of foreground Objects. The temporal correspondence of the objects is performed using the adapted Mean Shift algorithm ([GPL08]). This system proposes to restrict the information used by this algorithm to the pixels belonging to the foreground. In this way, possible errors on the background area are avoided. As a result of this algorithm we obtain an estimation for the centroid of the object j at time t ($\theta_{j,t}$), with the warranty that within the area of $\theta_{j,t}$ at this position there are one or more CC.

The system also takes into account that in the foreground detection the objects are often detected in more than one connected components, due for instance to the similarity between the color of some parts of the object and the background that breaks the connectivity of the foreground regions. Hence, this system associates to an object all the foreground connected components that are included (totally or partially) in a rectangle of the size of the object and centered in the Mean-Shift position estimation. This prevents the appearance of new Objects due to small errors in the foreground detection, which is common in connected components

based tracking systems ([GVPG03]). The size, centroid position and histogram of the Object $\theta_{j,t}$ is then updated in its corresponding register. If two or more objects share the same connected components, we will enter an occlusion situation. In this case, only the centroid position and the counter are updated, using the result of the Mean Shift algorithm to estimate the position.

If an Object $\theta_{j,t}$ has no CC associated, a Lost Object counter will be increased. When it reaches a given threshold , the Object $\theta_{j,t}$ is removed from the register.

After the tracking process the two Regions of Interest are created as input for the other two blocks:

- $\text{ROI}_{\text{analysis}}$ for the last segmentation block: If there is no occlusion in the scene, this ROI is defined as all the pixels inside the bounding box of the foreground objects. If there is occlusion, the tracking algorithm cannot ensure the correct position of the bounding boxes of the occluded objects. To avoid errors in these situations, this ROI will be the bounding box that contains the different objects that take part in the occlusion.

- $\text{ROI}_{\text{update}}$ for background updating: It is the complementary of the $\text{ROI}_{\text{analysis}}$. It is used to indicate background regions free of foreground objects in order to update the background with all the progressive changes and foreground detections that do not belong to the foreground objects.

4.4 Bayesian Foreground Segmentation Using Pixel-Based Background Model and Region Based Foreground and Shadows Models

This block of the system (identified as Region Fg segmentation in Figure 4.1) tries to enhance the segmentation of each object that is being tracked. For this purpose, we propose to follow the work-flow used in other works like [SS05] but with some modifications added to reduce the computational cost, and remove the so common false detections due to shadow effects.

That is, the classification is made in a Bayesian framework, introducing a prior that contains neighborhood information. A graph cut is used to make the classification in this context. For every frame I_t, the foreground, shadow and background models are combined to achieve the segmentation. We propose to use the more complete Gaussian Mixture Model in the joint color-space domain (SCGMM) for the foreground regions ([SS05]), the Gaussian Model also in the color-space domain (SCGM) for the shadow regions, and to use the Gaussian pixel-wise color

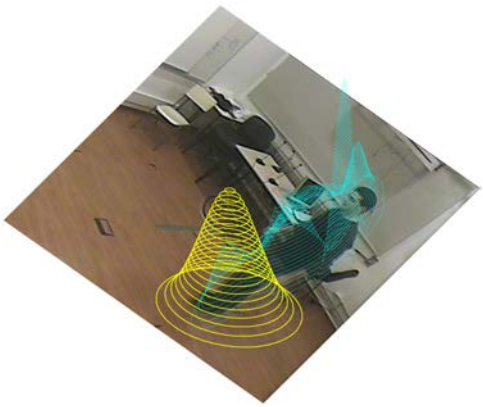


Figure 4.3: Spatial representation of foreground and shadow spatial models. In blue the foreground SCGMM. In yellow, the shadow SCGM.

model (CGM), from the pixel-wise segmentation block, for background modeling, which allows a very precise description of it and it is easy and computationally less expensive to update.

We thus combine a pixel-wise background model with region based models. The foreground and shadow models of each object are initialized when a new object is detected and both models are updated based on the classification performed on the current frame and in the $\text{ROI}_{\text{analysis}}$ obtained from the tracking block. The Gaussian model of every pixel assigned to background is updated recursively as it is explained in Section 4.2. The updated models are then used for the classification of the next frame I_{t+1}, which is performed by comparing the probabilities of foreground, shadow and background of every pixel within the graph cut algorithm. Figure 4.3 shows a graphical representation of the foreground and shadow spatial models when modeling a person (blue model) and the shadow projected to the ground (yellow color).

4.4.1 Foreground Model

Once the tracking process detects a new object to track, a foreground model is created and associated to it using the spatial and color information.

As commented before, since the foreground is constantly moving and changing, an accurate model at a pixel level is difficult to build and update. For this reason, we propose to use a Spatial Color Gaussian Mixture Model (SCGMM), as in [YZC$^+$07], because foreground objects are better characterized by color and position, and GMM is a parametric model that describes accurately multi-modal probability density functions. Thus, the foreground pixels are represented in a five dimensional space. The feature vector for pixel i, $zi \in \mathbb{R}^5$, is a joint domain-range representation, where the space of the image lattice is the domain, (XY), and the color space, (RGB), is

the range ([SS05]).

The likelihood of pixel i is then,

$$\begin{aligned} P(z_i|\text{fg}) &= \sum_{k=1}^{K_{\text{fg}}} \omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k) \\ &= \sum_{k=1}^{K_{\text{fg}}} \omega_k \frac{1}{(2\pi)^{5/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(z_i - \mu_k)^T \Sigma_k^{-1}(z_i - \mu_k)\right] \end{aligned} \tag{4.4}$$

where w_k is the mixture coefficient, $\mu_k \in \mathbb{R}^5$ and $\Sigma_k \in \mathbb{R}^{5\times5}$ are, respectively, the mean and covariance matrix of the k-th Gaussian distribution, $|\Sigma_k|$ is the determinant of matrix Σ_k. It is commonly assumed that the spatial and color components of the SCGMM models are decoupled, i.e., the covariance matrix of each Gaussian component takes the block diagonal form,

$$\Sigma_k = \begin{pmatrix} \Sigma_{k,s} & 0 \\ 0 & \Sigma_{k,c} \end{pmatrix}$$

where s and c stand for the spatial and color features respectively. With such decomposition, each foreground Gaussian component has the following factorized form:

$$G_{\text{fg}}(z_i, \mu_k, \sigma_k) = G_{\text{fg}}(s_i, \mu_{k,s}, \Sigma_{k,s})\ G_{\text{fg}}(c_i, \mu_{k,c}, \Sigma_{k,c}), \tag{4.5}$$

where $s_i \in \mathbb{R}^2$ is the pixel's spatial information and $c_i \in \mathbb{R}^3$ is its color value.

4.4.1.1 Initialization

The initialization of the foreground model is done via Expectation Maximization (EM) algorithm ([DLR+77]) in the overall five dimensional domain with the color-spatial information obtained from all the pixels detected as foreground by the pixel-wise foreground segmentation block, and located inside the object's ROI obtained from the tracking block. The number of Gaussians that will compound the model should be slightly higher than the number of color-spatial regions of the object to ensure that the object is correctly modeled with at least one Gaussian per region. There are several manners to obtain this number of regions. In our case, we choose to analyze the object's RGB-histogram in the following way: Once the foreground and background histograms are calculated, the number of bins used to define them are examined to detect the N first bins with higher probability which gather together the 70% of the color appearance probability. In each class, for each one of these bins, a Gaussian will be added to the model.

In the next frames, after initialization, the object will be segmented via the proposed Bayesian foreground segmentation analyzing only the $\text{ROI}_{\text{analysis}}$ region until its disappearance from the scene.

Figure 4.4: Example of GMM initialization. From left to right: Input color frame, corresponding mask of this frame, and the initial foreground SCGMM, where each ellipse corresponds to one Gaussian of the foreground model, filled with the mean color that each one is modeling.

Figure 4.4 displays an example of the Gaussian's initialization for a certain frame. In this Figure, third image shows the representation of the foreground model, where each ellipse is the spatial representation of one Gaussian of the model, and each one is filled with the mean color that each it is modeling in the color $c = RGB$ domain. The axis of the ellipses are defined according to the eigenvalues of the spatial covariance matrix ($\lambda = \lambda_1, \lambda_2$) as: $\text{axis}_i = 2\sqrt{\lambda_i}$. This consideration will be used in all the Gaussian's spatial representations that will appear throughout this Thesis.

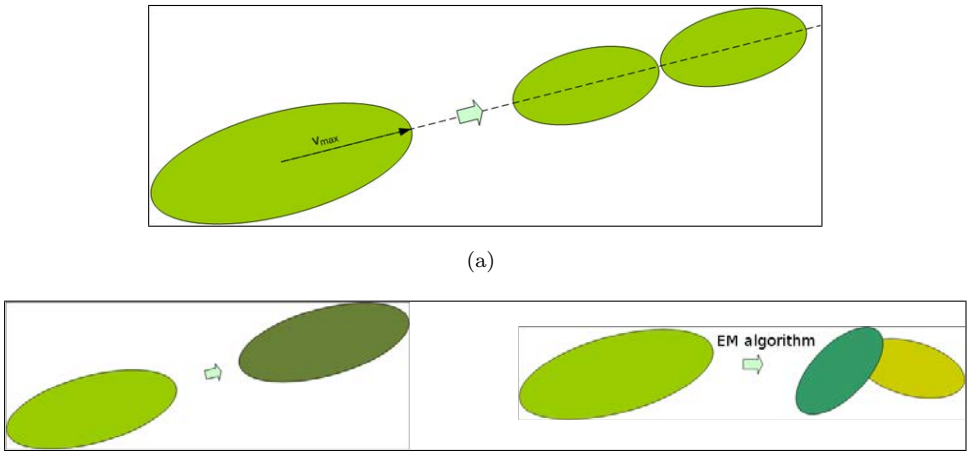
4.4.1.2 Updating

While we assume a static background, the foreground objects usually perform a displacement within the scene. In a normal situation, this displacement can be accompanied by an object rotation, which could produce the appearance of new color-spatial foreground regions belonging to the part of the object that was occluded to the camera until this moment. Thus, the spatial components of the Gaussian Mixture and also, the color ones, need to be updated after the classification in foreground, background or shadow of each frame.

The complete foreground updating in the spatial and color domains could lead to False Positives error propagation if the foreground regions present similar colors to the background and shadow ones.

Thus, we propose a two-steps updating for the foreground model. This updating allows a correct spatial domain updating and a conditional color updating which introduces new foreground color regions to the foreground model depending on the degree of similarity between the foreground model and the background and shadow ones. The two steps updating is as follows:

Spatial domain updating: the pixels classified as foreground form a mask that is used for the updating. In this step, only the spatial components of the Gaussian Mixture are updated. As it is proposed in [KS00a], we assign each foreground pixel



(a)

(b)

Figure 4.5: Graphical representation of the Gaussian split criterion. a) shows the split in the spatial domain where v_{max} is the eigenvector associated to the largest eigenvalue. b) depicts the color updating; Gaussian Color updating on the left; on the right, color updating by means of the creation of two Gaussians.

to the Gaussian k that maximizes:

$$P(k|z_i, \text{fg}) = \frac{\omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k)}{\sum_k \omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k)} \tag{4.6}$$

the denominator is the same for all the classes and can be disregarded.

Once each pixel has been assigned to a Gaussian, the spatial mean and covariance matrix of each Gaussian are updated with the spatial mean and variances of the region that it is modeling.

Also, in order to achieve a better adaptation of the model into the foreground object shape, we propose a Gaussian split criterion according to the spatial size of the Gaussian. The Gaussians that accomplish the following expression are split into two smaller Gaussians in the direction of the eigenvector associated to the largest eigenvalue, $\lambda_{\max}$: $\lambda_{\max} > \chi$, where χ is a size threshold. In our tests, $\chi = \max(\text{object}_{\text{height}}, \text{object}_{\text{width}})/4$ yields correct results.

Figure 4.5(a) displays a graphical example of the spatial updating.

Color domain updating: once the spatial components of the Gaussians have been updated, we update the foreground model according to the color domain. For each foreground Gaussian, we check if the data that it is modeling (according to the pixels assigned to this Gaussian) follows a Gaussian distribution. The multidimensional Kolmogorov-Smirnov test ([FF87]) can be used for this aim. Otherwise, simple tests based on distances analogous to (Equation 4.3) can be applied to the pixels assigned to a Gaussian in order to compute the percentage of these pixels that are well described by the Gaussian.

- If the data follows a Gaussian distribution, only one Gaussian is needed to model these pixels. In this situation, we first analyze whether a color updating is needed, comparing the Gaussian distribution in analysis with the Gaussian distribution that better models the data. This comparison can be made via Kullback-Leibler divergence ([Kul87]) or with simple tests that compare, each component $c = RGB$ of the mean values (μ_1 and μ_2) of the two distributions in relation with their variances (σ_1^2 and σ_2^2),

$$\|\mu_{1,c} - \mu_{2,c}\|^2 < min(k^2\sigma_{1,c}^2, k^2\sigma_{2,c}^2), \tag{4.7}$$

  where k is a decision constant (we use $k = 2.5$). Index 1 and index 2 denote the Gaussian distributions that we want to compare. In this case, index 1 denotes the Gaussian distribution of the foreground model and index 2 denotes the Gaussian distribution that better models the data. If the Gaussian in analysis models correctly the data, no updating is necessary. Otherwise, the color domain parameters of the Gaussian are replaced by the data ones.

- If not, it means that more than one Gaussian is needed to model these pixels. Another Gaussian distribution is created, and we use the EM algorithm to maximize the likelihood of the data in the spatial and color domains.

Figure 4.5(b) displays a graphical example of both color updating possibilities.

In order to increase the robustness of the system, color updating of the foreground model is only performed if the Gaussian of the foreground model is different enough in the color domain from the Gaussians of the background and shadow models that correspond to the same spatial positions. Again, we can apply Kullback-Leibler divergence or compare the mean value of the distributions. For instance, we consider that the foreground model can be updated if at least 70% of the pixels that the new Gaussian represents have a background model that does not accomplish (4.7).

Figure 4.6 displays an example of nine updating iterations starting from the initialization presented on Figure 4.4. As we can see, if the model is correctly initialized, the spatial and color updating can split the Gaussians to obtain a correct modeling of the foreground object that we want to segment. If we don't have any spatial restriction defined by χ, the spatial updating can obtain a perfect modeling of the object by using an elevated number of Gaussians, which is, in fact non practical, due to the computational burden. Realize that at each iteration we are doubling the number of Gaussians, increasing the likelihood of the overall model as well as the computational cost of this processing. Figure 4.7 displays the likelihood evolution associated to the foreground model at each iteration. In these graphs, we can observe how doubling the number of the Gaussians of the model, the Log-

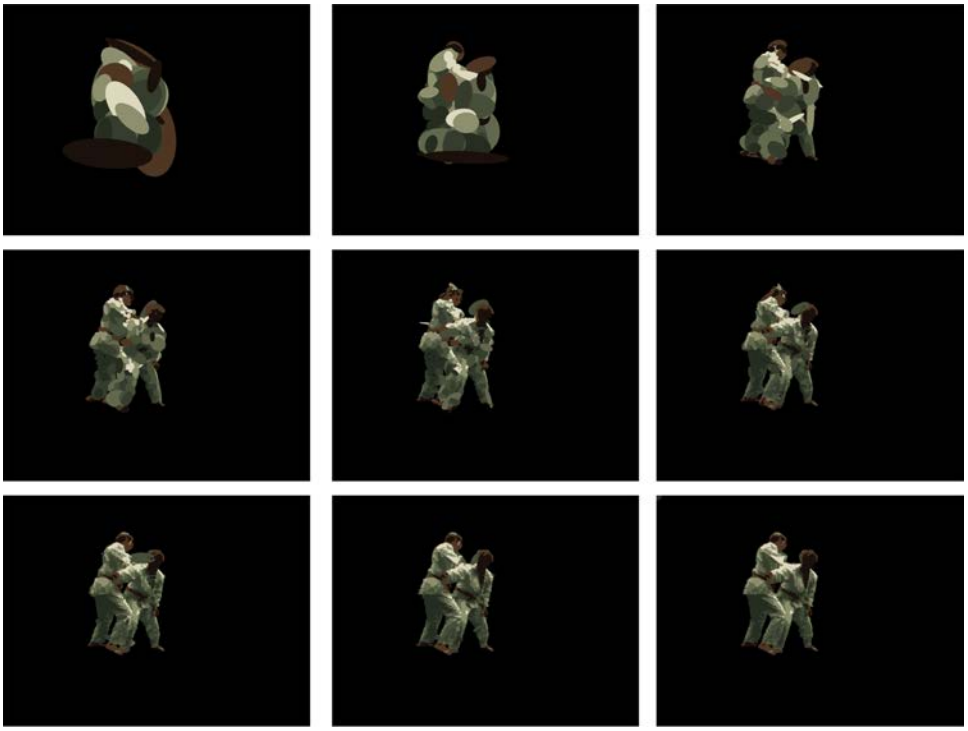


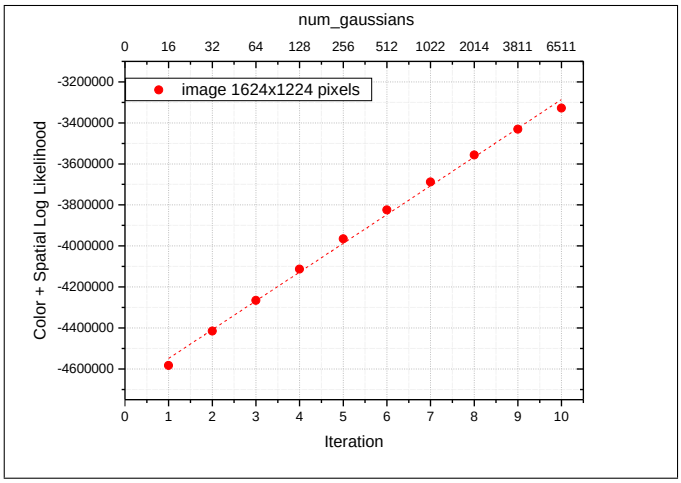
Figure 4.6: Example of foreground model updating. From left to right and from up to down, foreground model updating iterations over the same frame presented in Figure 4.4. Each ellipse corresponds to the spatial representation of the Gaussians of the model, colored with the mean color that each one is modeling. The Gaussians adapt correctly to the real shape of the object while increasing the number of Gaussians of the model at each iteration.

likelihood presents a linear improvement, which means an exponential evolution of the foreground likelihood when doubling the number of Gaussians of the model.

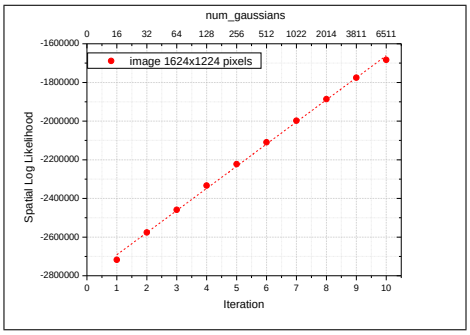
4.4.2 Shadow Model

We propose a new system to remove the so common false positive detections that shadow effects generate in foreground segmentation. As we have said in the introduction, most segmentation methods that use foreground modeling do not take into account the shadow effect despite it is a common source of errors. Our experiments confirm that foreground modeling is not enough to avoid shadow effects in some scenarios. We can observe it in Figure 4.8.

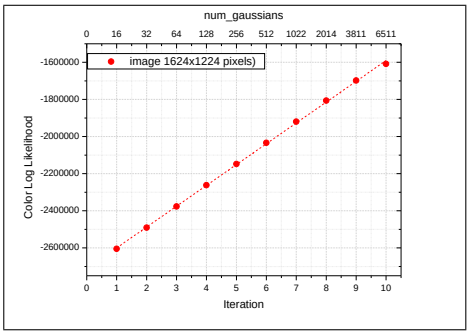
The fact that we consider a Bayesian framework between foreground and background models, like [SS05] leads us to incorporate a probabilistic model of the shadow within the same framework. Hence, the use of a shadow model for each detected object is proposed with the aim of including probabilistic information about the kind of shadow effect that each object is generating. Therefore, we propose to associate to each object a shadow model, together with the foreground model. Since



(a)



(b)



(c)

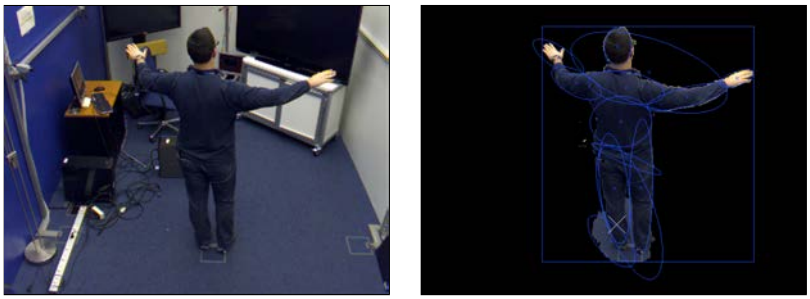
Figure 4.7: Log likelihood graphs of the foreground model for each iteration displayed in Figure 4.6 with an image size of 1624x1224 pixels. a) depicts the color+spatial likelihood, b) and c) show the spatial and color likelihood respectively.

the shadows are constantly moving and changing like the foreground, and in most of the cases they can be described with only one spatial-color region, we propose to use a Spatial Color Gaussian Model (SCGM), which presents similar benefits than a more complex SCGMM (as verified in our tests), but significantly reducing the computational cost.

The initialization of the shadow model is done analyzing the shadow pixels (obtained from the pixel-wise segmentation block) that appear inside the ROI of the object, obtaining its color-spatial mean and covariance matrix, and considering, as in the foreground model, that space and color dimensions are decoupled.

For the next frames, spatial and color mean and variance are updated with the detected shadow pixels. Mean is updated according to the following causal low-pass filter equation:

$$\mu_t = \alpha \sum_i Z_{sh,i} + (1-\alpha)\mu_{t-1}, \tag{4.8}$$



(a) (b)

Figure 4.8: Example of foreground segmentation with false positive detections due to shadow effects. a) Original image. b) Foreground segmentation using SCGMM fg model and pixel-wise Gaussian bg model.

where α is a small time constant (we use a value of 0.2) and Z_{sh} denotes all pixels detected as shadow in the shadow detection step. Covariance matrixes Σ_s, Σ_c are recalculated taking into account the new mean μ_t and the new pixels classified as shadow.

As foreground regions normally overlap the shadow ones, these shadow regions usually present different non-Gaussian real shapes along the scene in analysis. Hence, the shadow Gaussian model can reach a high probability in those pixels located close to the spatial mean with similar color to the shadow, despite they could belong to the foreground.

To adapt the Gaussian spatial modeling to the real shadow area and improve the shadow detection in those scenes where foreground and shadow have similar color, the dependence between color and spatial domains is used via Bayes formulation. Thus, we achieve a better representation of the shadow shape and avoid errors in the pixels classification. Therefore, we define the likelihood of pixel i given shadow as:

$$\begin{aligned} P(z_i|\mathrm{sh}) &= P(x_i|c_i,\mathrm{sh}) \cdot P(c_i|\mathrm{sh}) \\ &\simeq \varphi(x_i,c_i,\mathrm{sh})\; G_{\mathrm{sh},s} \cdot G_{\mathrm{sh},c}, \end{aligned} \tag{4.9}$$

where $G_{\mathrm{sh},s} \equiv G_{\mathrm{sh}}(x_i,\mu_s,\Sigma_s)$, $G_{\mathrm{sh},c} \equiv G_{\mathrm{sh}}(v_i,\mu_c,\Sigma_c)$ and $\varphi(x_i,c_i,\mathrm{sh})$ gathers the dependence between spatial and color domains:

$$\varphi = \begin{cases} 0 < \eta < 1 & \text{if } G_{\mathrm{sh},c} < G_{\mathrm{sh},s} \\ \gamma & \text{otherwise.} \end{cases} \tag{4.10}$$

In this way, we penalize the shadow model in all those pixels where $G_{\mathrm{sh},c} < G_{\mathrm{sh},s}$, i.e., we penalize the likelihood of those pixels that are closer, in the spatial sense than in the color sense, to the shadow model. The scale factor η satisfies: $0 < \eta < 1$ (in our experiments $\eta = 0.2$ yields correct results). Also, to maintain the p.d.f. property $\sum_{i\in\Omega_d} P(x_i|c_i,\mathrm{sh}) = 1$, where Ω_d denotes the discrete image domain, an

(a)

(b)

Figure 4.9: Shadow spatial models reducing the probability in foreground regions.

increase of the probability of the rest of the pixels is proposed. Hence, a likelihood scale factor γ is used in the shadow group of pixels that fulfill $G_{\text{sh},c} \geq G_{\text{sh},s}$:

$$\begin{aligned} \gamma &= \frac{1 - \sum_K \eta P(x_i|\text{sh})}{\sum_M P(x_i|\text{sh})} \\ &= \frac{1 - \sum_K \eta G_{\text{sh},s}}{\sum_M G_{\text{sh},s}}, \end{aligned} \tag{4.11}$$

where K is the set of pixels index where $G_{\text{sh},c} < G_{\text{sh},s}$ and M is the set of remaining pixels. As we can observe in Figure 4.9, this likelihood adapts better the shadow model to the shadow region, reducing the spatial probability in those pixels belonging to the foreground we want to segment.

4.4.3 Background Model

Since we want to combine the range background model with the joint range-domain foreground model, we need to extend the pixel-based model (Equation 4.1), obtained from the first block of the system, to a five dimensional model by using a SCGMM, analogously to the foreground model. For that, we use a mixture of N five dimensional Gaussians, one representing each pixel in the image and thus having equal mixture proportions,

$$P(zi|bg) = \sum_{k=1}^{N} \frac{1}{N} G_{bg}(z_i, \mu_k, \Sigma_k) \tag{4.12}$$

where

$$G_{bg}(z_i, \mu_k, \Sigma_k) = \delta(x_i - \mu_{k,s}) P(c_i|bg).$$

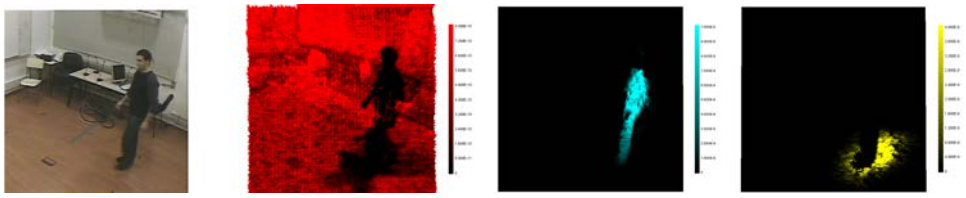


Figure 4.10: Spatial color pixel probabilities. From right to left: Original frame. In red background probability. Foreground probabilities are represented in cyan color. Yellow shows shadow probabilities.

Thus, we are using N Gaussians, each one centered (in space) at each pixel position with a zero spatial variance. This is sufficient for indoor scenarios with a static camera, although a small spatial variance can be used in order to allow for small outdoor background motions or camera shaking.

4.4.4 Classification

Once the foreground, shadow and background models have been computed, at frame t, the labeling can be done, assuming that we have some knowledge of foreground, shadow and background prior probabilities, $P(\text{fg})$, $P(\text{shadow})$ and $P(\text{bg})$ respectively, using a Maximum A Posteriori (MAP) decision. The priors can be approximated by using the foreground, background and shadow areas, computed as number of pixels, in the previous frame, $t-1$,

$$P(\text{fg}) = \frac{Area_{\text{fg}}|_{t-1}}{N}; \quad P(\text{bg}) = \frac{Area_{\text{bg}}|_{t-1}}{N};$$

$$P(\text{sh}) = \frac{Area_{\text{sh}}|_{t-1}}{N}.$$

Figure 4.10 shows a graphical example of final pixel probability for each one of the pixels of the image. As it can be seen, having a model for background, foreground and shadow classes, pixels can be correctly modeled.

A pixel i may be assigned to the class $l \in \{\text{foreground, background, shadow}\}$ that maximizes $P(l_i|z_i) \propto P(z_i|l_i)P(l_i)$ (since $P(z_i)$ is the same for all classes and thus can be disregarded).

To simplify the classification, and assuming that shadow and background pixels will be treated in the same way for the final segmentation mask, we combine shadow results into the background ones according to the following criterion:

$$P(\text{bg}|z_i) = max(P(\text{bg}|z_i), P(\text{sh}|z_i)) \tag{4.13}$$

Analogously to [SS05, YZC$^+$07], we choose to additionally consider the spatial context when taking the segmentation decisions, instead of making an individual classification of the pixels. We consider for this aim a MAP-MRF framework in

order to take into account neighborhood information defining a prior $P(l)$ with two terms: the class prior for each one of the pixels $P(l_i)$ and the regularization term that is computed using the neighborhood information. Then, if we denote by l the labeling of all the pixels of the image: $l = \{l_1, l_2, ..., l_N\}$, and by Nb_i the four connected neighborhood of pixel i we have:

$$\begin{aligned} P(l|z) &\propto P(z|l)P(l) \\ &= \prod_{i=1}^{N} P(z_i|l_i) \cdot \prod_{i=1}^{N} P(l_i) \cdot \exp\left(\sum_{i=1}^{N}\sum_{j\in Nb_i} \lambda(l_i l_j + (1-l_i)(1-l_j))\right) \\ &= \left(\prod_{i=1}^{N} P(z_i|l_i)\ P(l_i)\right) \cdot \exp\left(\sum_{i=1}^{N}\sum_{j\in Nb_i} \lambda(l_i l_j + (1-l_i)(1-l_j))\right), \end{aligned} \tag{4.14}$$

Taking logarithms in the above expression leads to a standard form of the energy function that is solved for global optimum using a standard graph-cut algorithm ([BVZ01]). (See Appendix B for more information about energy minimization).

4.5 Results

We performed both qualitative and quantitative evaluation of our system. Quantitative results are obtained analyzing the MUHAVI public Data Base ([Vo09]), which is compound by a set of twelve sequences where one person performs some actions (run, punch and kick) inside a smart room. The ground truth of each frame of the sequences is available by means of manual segmentation, and it is used in order to make the numerical evaluation. Qualitative results are obtained analyzing another two different smart room settings in Figure 4.11 and Figure 4.13 to show a wide range of possible scenarios. We compared the proposed method to three state of the art pixel based background segmentation methods:

- The parametric Running Gaussian Average method ([WADP02]) (RGA) that has proved to work efficiently in controlled indoor environments like smart rooms.
- The combination of RGA method with the shadow removal method ([XLP05]) (RGA + sh.rem.), which shows an improvement of the RGA method, using shadow removal techniques proposed in [HHD99] complemented with a morphological analysis.
- The nonparametric background subtraction method Kernel Density Estimation ([EHD00]) (KDE), which is also a well known and widely used foreground segmentation technique.

(a) Sequence 1 (b) Sequence 2

Figure 4.11: Qualitative results. Rows, from top to bottom: original sequence, pixel-based fg detection ([XLL04]), region based fg detection ([YZC$^+$07]) (green ellipses represent the spatial domain of the Gaussians belonging to the bg model. Red ellipses are their counterpart in the fg model), our results (red ellipses represent the Spatial fg model, white ellipse represents the Spatial shadow model).

The technique proposed in [YZC$^+$07] has been considered only in qualitative results, because it is suitable for scenes where the object remains more static than in the evaluation sequences. In this comparison, a complete analysis of our system is performed testing it without shadow removal (Bayes.), and with the shadow removal technique presented in this section (Bayes.+sh.rem.). In this way, we will be able to see the positive effect of including the shadow model into the Bayesian framework.

In Table 4.1 quantitative results of MUHAVI database can be observed. The metrics used in the evaluation are: Precision (P), Recall (R) and f_{measure} metrics, formulated as follows:

$$P = \frac{\text{TP}}{\text{TP} + \text{FP}}; \qquad R = \frac{\text{TP}}{\text{TP} + \text{FN}};$$

$$f_{\text{measure}} = \frac{2RP}{R + P}.$$

where TP, FP and FN are $TruePositive$, $FalsePositive$ and $FalseNegative$ pixels detected in the evaluation: frame, sequence or set of sequences.

As it can be observed in Table 4.1, the basic Running Gaussian Average and Kernel Density Estimation methods are those that achieve the lowest f_{measure}, in part due to the shadow effects and the vulnerability in front of foreground-background color similarities. When RGA method is combined with an efficient Shadow removal system, foreground segmentation quality improves in a wide range reaching better precision and recall rates that allow an f_{measure} of 0.82, but problems for

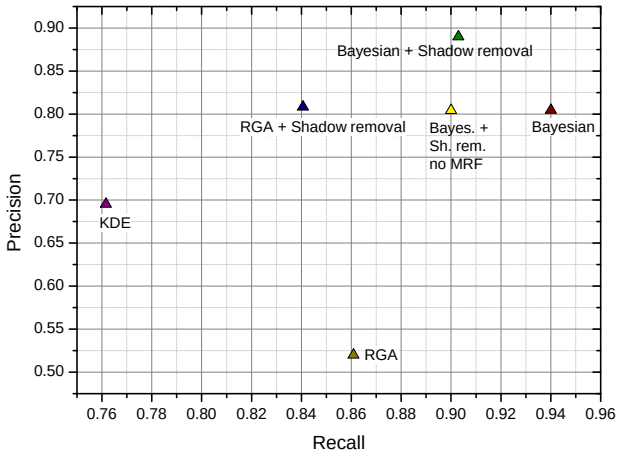


Figure 4.12: Precision vs Recall Graph.

background-foreground color similarities still remain. Our Bayesian+shadow removal system improves all these rates achieving an f_{measure} of 0.90 solving segmentation problems that shadow effects and foreground-background color similarities produce. Bayesian system shows the results of our method without using the shadow removal. As it can be observed, the proposed shadow removal system achieves a Precision improvement of 8% that denotes how FP detections are reduced thanks to the SCGM shadow modeling method proposed. Only a Recall decrease of 3% is obtained because FN detections increase due to the shadow removal algorithm. The results improvement that the Markov Random Field framework adds to the overall proposal can be observed by comparing the Bayesian+shadow removal column with the Bayesian+shadow removal no MRF column. In this column, we show the results obtained by our proposal using a simple $P(\text{fg}|z_i)$, $P(\text{bg}|z_i)$ and $P(\text{sh}|z_i)$ pixel-wise comparison instead of the MRF framework. As it can be observed, using the MRF classification, the system presents an f_{measure} improvement of 5%, an 8% in precision and a 1% in recall. Table 4.1 also shows the f_{measure} increase percentage ($\Delta\%$ f_{measure}) of each method with respect to the RGA segmentation.

In Figure 4.12 we can observe a graphical Precision-Recall comparison between all the methods tested in this evaluation where we can appreciate that our system (Bayesian+shadow removal) is the best option according to precision-recall ratio. Table 4.2 shows the f_{measure} calculated for each one of the sequences. It is important to highlight that the Bayesian method and the Bayesian+Shadow removal method present similar values in some scenes. This occurs in those scenes without shadow effects.

In Figures 4.11 and 4.13, a qualitative comparison can be observed. Smartroom sequences not belonging to the MUHAVI data base, which performed poorly

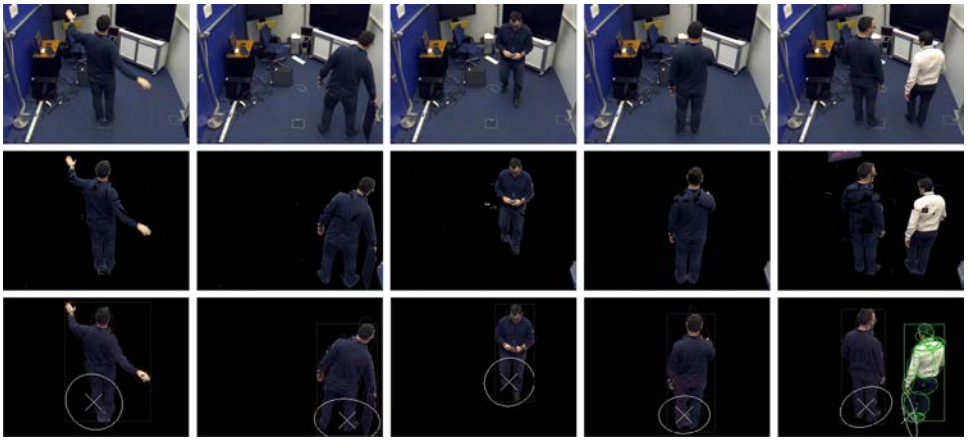


Figure 4.13: Qualitative results. Rows, from top to bottom: original sequence, pixel-based fg detection proposed in [XLL04], our results.

using pixel-based methods ([WADP02, SG00]) have been selected. In particular, two sequences are shown in this paper where the colors of the foreground objects are in the same range than a part of the background. This generates many misses in the foreground detection when only the background model is used.

Results can improve using a region based model ([YZC$^+$07]). However, a high computational load is required and some errors still appear. As it can be observed, in the results of our system the segmentation is more robust, reducing the false positive detections thanks to the proposed updating of the background model, and also the false negatives are reduced thanks to the use of the foreground and shadow spatial-color modeling. In the results some ellipses can be seen: the colored ones correspond to the spatial domain of the GMM foreground model, and the white one corresponds to the spatial domain of the Gaussian Shadow model (that appears when shadows are present).

In Figure 4.11, we can see comparison results in two different smart room scenarios where the problem of color similarity between fg and bg is present. In this figure, our method is compared with a pixel-wise method ([XLL04]) and a region-based method proposed in [YZC$^+$07].

Figure 4.13 shows a sequence result where two people interact inside a smart room. As well as the problem of color similarity, some false detections appear due to the interaction with background objects and the dynamic background regions like the one that is created by the TV screen.

In Figure 4.14, we can observe the segmentation of one person. The segmentation obtained by our method (second row, second column), presents less false positive and false negative detections than the segmentation obtained by using the pixel-wise method proposed in [XLL04] (first row, second column), thanks to the correct

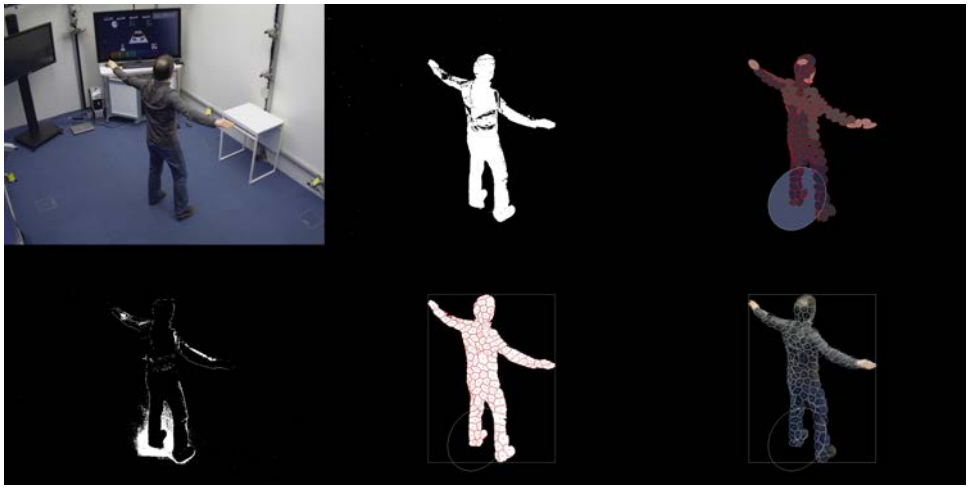


Figure 4.14: Qualitative results. From top to bottom and from left to right: original sequence, pixel-based fg detection ([XLL04]), foreground and shadow models (Gaussians are filled with the mean RGB value that are modeling), shadows detected by [XLL04], our segmentation results with the foreground and shadow models (fg Gaussians in red, sh Gaussian in white), our segmentation showing foreground pixels with original colors.

probabilistic modeling achieved by foreground and shadow models (depicted in the third column).

Figure 4.15, depicts one example of the MUHAVI data base results, where a comparison with the method proposed by [XLL04] can be observed. Thanks to the Bayesian approach between shadow, foreground and background models, our method achieves a correct shadow removal avoiding most false positive and false negative foreground detection that other methods present. Finally, Figure 4.14 shows a different smart-room scenario.

4.5.1 Computational Cost

There are two main processes that spend almost all the amount of time devoted to the foreground segmentation: First, the evaluation of each Gaussian of the foreground model over the pixels of the image, and second, the updating process to adapt the model to the changes that appear in the object. Hence, the computational cost of the overall system depends on the size of the image that we want to analyze, and the number of Gaussians utilized to model the foreground object. As we have seen, if we increase the number of Gaussians of the foreground model, we achieve a better characterization of the object, but the computational cost will be also increased. Therefore, there is a trade-off between the processing time, the size of the images and the number of Gaussians of the foreground model.

Figure 4.16, Figure 4.17 and Figure 4.18 display some graphs that show how the

Figure 4.15: Qualitative results. Rows, from top to bottom are: original frames, foreground segmentation obtained with [XLL04], foreground segmentation obtained with our method. Note that gray regions that appear in the borders of the person, in the foreground segmentation mask, are due to the interlacing effect present in the sequence, which produces white foreground lines interlaced with black background lines.

time devoted to the decision and updating steps vary according to the resolution of the images and the number of Gaussians used in the sequence presented at Figure 4.4. Three image resolutions are evaluated in these graphs: 406×306, 812×612 and 1624×1224, by using an Intel Xeon X5450 3.0GHz processor. As we can observe, the processing time increases linearly as we increase the Gaussians of the foreground model. In order to work at real-time, resolutions around 406x306 pixels, and a foreground model performed by 10 to 100 Gaussian distributions have to be chosen. With that framework, the system allows a speed of 0.44 frames/second, for a video sequence of 406x306 pixels with one object in scene.

Table 4.1: Overall MUHAVI Data Base Comparison Results. In bold type the results corresponding to the best scores.

Metrics	**Foreground Segmentation Technique**					
	RGA	KDE	RGA+ sh. rem.	Bayes.	**Bayes.+ sh. rem.**	Bayes.+ sh. rem no MRF
precision	0.52	0.69	0.80	0.80	**0.88**	0.80
recall	0.86	0.76	0.84	**0.94**	0.91	0.90
f_{measure}	0.65	0.72	0.82	0.87	**0.90**	0.85
$\Delta\%$ f_{measure}	-	11.77	26.72	33.93	**38.83**	31.41
f.p.s	**11.02**	0.24	7.50	0.50	0.44	0.46

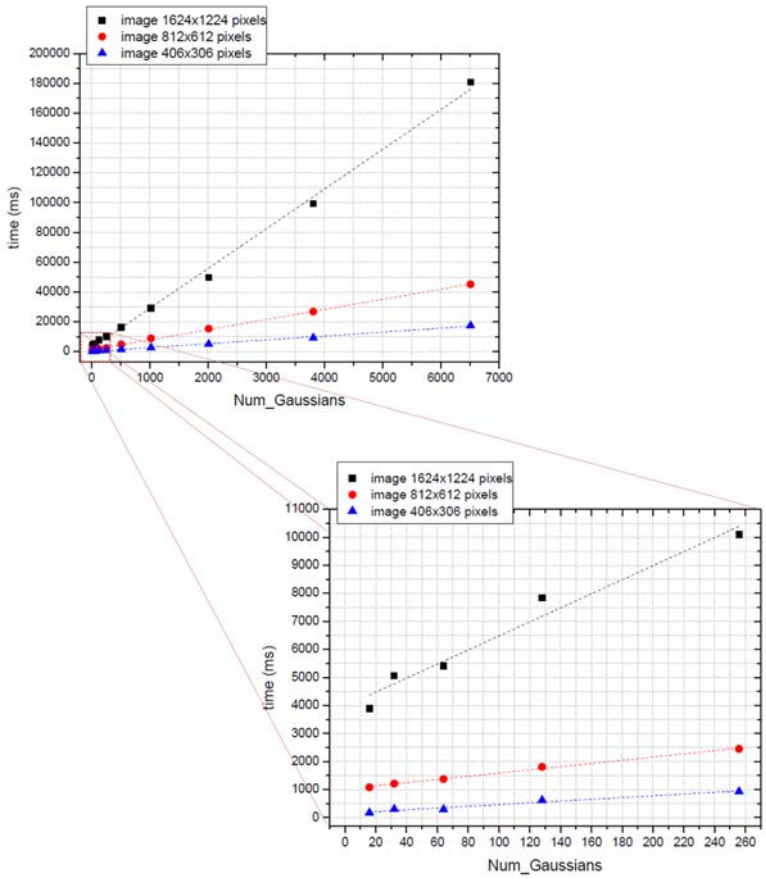


Figure 4.16: Computational cost of the probability computation and the classification steps. Processing time is analyzed according to the number of Gaussians of the foreground model and the resolution of the image.

4.6 Conclusions

This chapter presents a system for enhanced foreground objects segmentation purpose. In this system we combine successfully three techniques: initial pixel-wise foreground segmentation, tracking system based on MeanShift and final foreground segmentation based on Bayesian framework via pixel-wise background modeling and foreground spatial-color modeling. Each of these blocks has a specific function, and has been configured to implement the surveillance concept: be aware for external changes, detect and track the objective, and refine the detection. Also, a new technique for shadow removal into the specific Bayesian framework has been presented and used into the overall system to avoid the so common errors that shadow effects produce. The results show that the proposed system improves the foreground segmentation obtained with other pixel-wise methods, reducing the false positives, and false negatives detections also in those complicated scenes where similarity between foreground and background colors appears. In future work we will consider to improve the computational cost under the assumption that it can be easily reduced

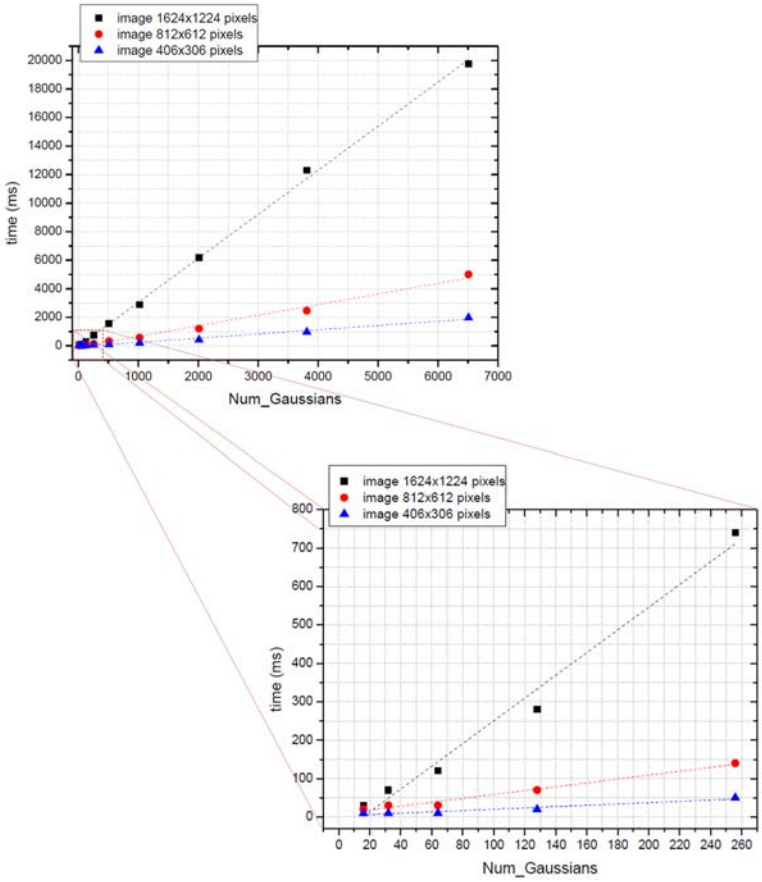


Figure 4.17: Computational cost of the updating step. Processing time is analyzed according to the number of Gaussians of the foreground model and the resolution of the image.

via parallel processing using multi-threading, and programming some algorithms under CUDA GPU programming.

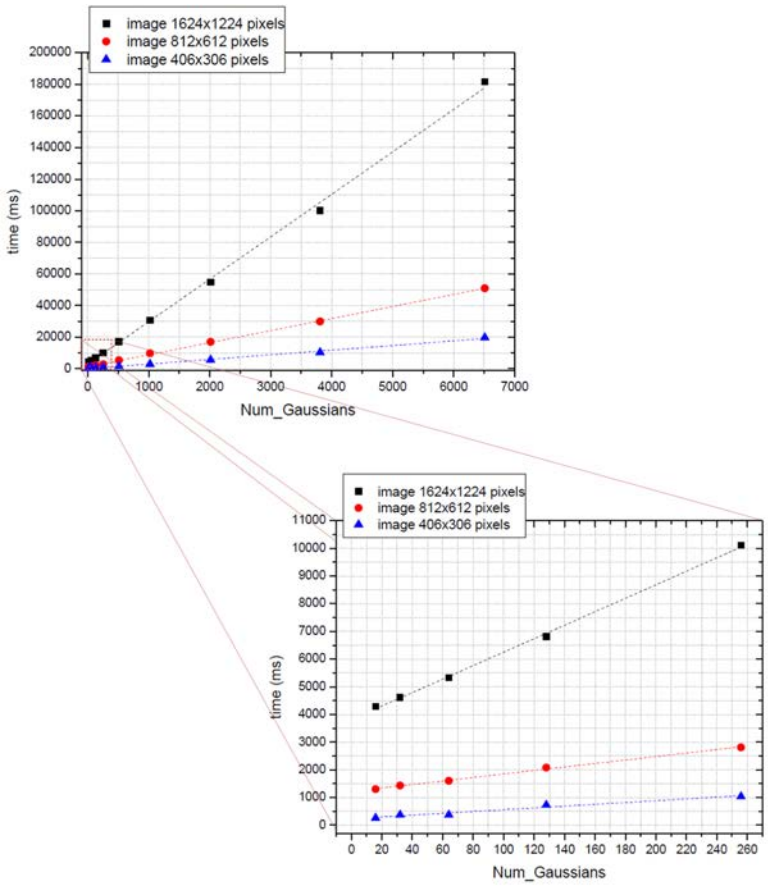


Figure 4.18: Computational cost graph of the of the probability computation, classification and updating steps. Processing time is analyzed according to the number of Gaussians of the foreground model and the resolution of the image.

Table 4.2: MUHAVI Data Base f_{measure} Comparison Results. In bold type the results corresponding to the best scores.

Sequence	**Person & Camera**	**Foreground Segmentation technique**					
		RGA	KDE	RGA+ sh. rem.	Bayes.	**Bayes.+ sh. rem.**	Bayes.+ sh. rem no MRF
RunStop	P1Cam3	0.50	0.74	0.83	0.86	**0.90**	0.87
	P1Cam4	0.69	0.78	0.85	0.83	**0.88**	0.85
	P4Cam3	0.61	0.68	0.81	0.82	**0.87**	0.82
	P4Cam4	0.70	0.78	0.85	0.83	**0.86**	0.84
Punch	P1Cam3	0.64	0.63	0.80	0.86	**0.92**	0.87
	P1Cam4	0.74	0.79	0.84	0.92	**0.93**	0.87
	P4Cam3	0.62	0.66	0.77	0.86	**0.91**	0.85
	P4Cam4	0.75	0.88	0.85	**0.90**	**0.90**	0.82
Kick	P1Cam3	0.62	0.58	0.77	0.88	**0.92**	0.87
	P1Cam4	0.70	0.84	0.88	0.90	**0.91**	0.87
	P4Cam3	0.63	0.66	0.80	0.87	**0.90**	0.84
	P4Cam4	0.78	0.88	0.89	**0.91**	0.90	0.85

Chapter 5

Bayesian Foreground Segmentation for Moving Camera Scenarios

5.1 Introduction

Objects segmentation and tracking in moving camera scenarios is of main interest on several high level computer vision applications like human behavior analysis or video sequence indexation among others, where a specific segmentation of the object, previously determined by the user, is needed. This kind of scenarios are common in video recordings, but present a special challenge for objects segmentation due to the presence of relative motion concerning the camera observer point and the foreground object to segment, which causes a non-stationary background along the sequence. Therefore, this scenario differs from fixed camera ones, where an exact background can be learned at a pixel-wise level [WADP02, SG00] and fixed camera with constrained motion scenarios, typical of surveillance cameras with a programmed camera path, which can be considered as a static mosaic from the dynamic scenes [IB98]. Instead, moving camera scenarios present a more difficult framework due to the impossibility of applying well known pixel-wise techniques for computing the background subtraction, and it has led to the publication of several new proposals that addresses this topic in the last few years. [CFBM10] presents a review of the most recent background segmentation systems within this area.

5.1.1 State of the Art

The different techniques proposed in previous works can be grouped into three classes:

-Techniques based on camera motion estimation. These methods compute camera motion and, after its compensation, they apply an algorithm defined for fixed camera. [AMYT00] uses frame differencing and active contour models to compute the motion estimation. In [SA02], the authors apply background subtraction using the background obtained through mosaicing numerous frames with warping transforms, while [JTD+08] proposes a multi-layer homography to rectify the frames and compute pixel-wise background subtraction based on Gaussian Mixture Model.

-Methods based on motion segmentation. In these methods the objects are mainly segmented by analyzing the image motion on consecutive frames. [SB02] proposes to use image features to find the optic flow and a simple representation of the object shape. [GT01] proposes a semi-automatic segmentation system where, after a manual initialization of the object to segment, a motion-based segmentation is obtained through region growing algorithm. In [CPV04] an approach based on a color segmentation followed by a region-merging on motion through Markov Random Fields is proposed, while in [VM08] the authors propose a Mean Shift segmentation and tracking applied to face recognition that relies on a segmentation of the area under analysis into a set of color-homogenous regions. In this proposal, the use of regions allows a robust estimation of the likelihood distributions that form the object and background models, as well as a precise shape definition of the object being tracked. This accurate object definition allows the object model to be updated through the tracking process, handling variations in the object representation.

-Based on probabilistic models: the objects to segment are modeled using probabilistic models that are used to classify the pixels belonging to the object. [LLR08] proposes a non parametric method to approximate, in each frame, a pdf of the objects bitmap, estimating the maximum a posteriori bitmap and marginalizing the pdf over all possible motions per pixel.

The main weakness of the systems based on motion estimation is the difficulty to estimate the object or camera motion correctly and the impossibility of subtracting the background when dynamic regions are present, which produces many false positive detections. On the other hand, proposals based on using foreground object probabilistic models present a more robust segmentation, but can lead to segmentation errors when the close background presents similar regions to the object.

In this chapter we propose a new technique for object segmentation in moving camera scenarios that deals with the last group of segmentation methods based on probabilistic models. We propose to use the region-based probabilistic model, the

Spatial Color Gaussian Mixture Model (SCGMM) to model not only the foreground object to segment, but also the close-background regions that appear surrounding the object, allowing, in this manner, a more robust classification of the pixels into foreground and background classes. The use of this technique achieves a correct segmentation of the foreground object via global MAP-MRF framework for the foreground (fg) and background (bg) classification task.

5.2 Proposal

The main strategies of the state of the art to achieve the segmentation of a certain object in a moving camera scenario, focus on analyzing two main factors: the scene motion between frames and the object characteristic features. These proposals are based on the principle that this kind of sequences present two different motions corresponding to the camera and to the object to segment.

We propose to extend the framework presented in the previous chapter to solve the segmentation problem in moving camera scenarios. This proposal was developed with the collaboration of Montse Solano who, carrying out her bachelor project [Pal11], contributed in testing the algorithm in different scenarios. Consider a moving camera sequence, where the camera performs some movements of translation, rotation and zoom and the object to segment is also moving inside the scene, changing also its orientation and making some rotations.

We will consider that the camera translation and rotation effects, together with the object orientation and translation changes are equivalent to consider a background motion behind the object to segment.

Therefore, using a dynamic region of interest, centered in the object detection obtained in the previous image, we will be able to consider that the background is a plane located behind the object to segment, which suffers some spatial modifications along the sequence and where new background regions appear in the limits of the image (usually due to camera displacements). Figure 5.1 shows an example of this dynamic region of interest.

To perform the segmentation we will use two probabilistic models: One to model the foreground object to segment, and another to model the background that is surrounding the object, with the objective that the background model assumes the new background regions that appear close to the object, achieving a robust classification process of the pixels among the two classes. Both models must also be flexible to assume possible camera zoom and object rotations that occur along the sequence.

The scene under analysis can suffer several spatial transformations: camera

Figure 5.1: Example of ROI. d' is a predefined size proportional to the object area that allows all possible movements of the object, so as to achieve a correct segmentation.

zoom, foreground object rotations and background rotation and translation. We propose a segmentation system that allows us to overcome all these situations, which consists of two separated parametric models to model the foreground object to segment and the close background that envelopes the object. For this purpose, we will use the Spatial Color Gaussian Mixture models (SCGMM). The work-flow of the system, shown in Figure 5.2, is as follows:

At the beginning, the system needs an input mask of the object that we want to segment. This region mask can be obtained via manual segmentation or using any segmentation tool, and it is used to:

- Define the dynamic Region of Interest of the object, defined as the bounding box that encloses the object with a percentage of close background.
- Initialize the foreground and the close background SCGMM that appear inside the already defined objects' ROI.

For each frame of the sequence, there is a three steps process: Classification of each pixel inside the bounding box according to the foreground and background models defined from the previous frame, updating of each model using the results obtained from the classification step and redefinition of the ROI according to the resultant foreground object segmentation. The details of this segmentation system will be explained in the following sections.

5.2.1 Dynamic Region of Interest

In order to achieve the segmentation of the foreground object, we make a local foreground object segmentation. We define the background model within a dynamic

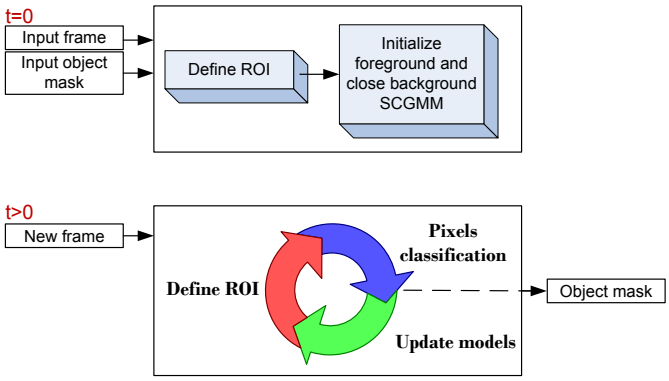


Figure 5.2: Work flow of the proposed system.

Figure 5.3: Dynamic Region of interest over the initialization mask. The T-shirt is the object under segmentation.

bounding box surrounding the foreground object. This neighborhood is defined according to some constraints of computational cost, and accuracy in the background modeling.

The bounding box has to present a certain size that allows the background model to achieve a correct close background representation in all the boundaries of the object, allowing all possible movements of the object to segment, but it has to be small enough to allow a reduced computational cost when updating models or calculating pixel probabilities. The model used has to be flexible enough to incorporate new parts of the background that appear around the object as the camera or the object move along the scene.

Thus, the bounding box will be centered at the geometric center of the object, with the limits of the object to segment plus an offset d that we define as a percentage of the largest axis of the ellipse that envelopes the object. 20% yields correct results in most considered scenarios. Figure 5.3 shows a graphical example of this bounding box.

5.2.2 Probabilistic Models

A good segmentation of foreground objects can be achieved if a probabilistic model for the foreground and also for the close background are constructed. Hence, we classify the pixels in foreground (fg) and background (bg) classes. Since in this kind of sequences the foreground and background are constantly moving and changing, an accurate model at a pixel level is difficult to build and update. For this reason, we use the region based Spatial Color Gaussian Mixture Model (SCGMM), as presented in the previous chapter, because foreground objects and background regions are better characterized by color and position, and GMM is a parametric model that describes accurately multi-modal probability.

Thus, the foreground and background pixels are represented in a five dimensional space. The feature vector for pixel i, $zi \in \mathbb{R}^5$, is a joint domain-range representation. The likelihood of pixel i is then,

$$\begin{aligned} P(z_i|l) &= \sum_{k=1}^{K_l} \omega_k G_l(z_i, \mu_k, \Sigma_k) \\ &= \sum_{k=1}^{K_l} \omega_k \frac{1}{(2\pi)^{5/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(z_i - \mu_k)^T \Sigma_k^{-1}(z_i - \mu_k)\right] \end{aligned} \tag{5.1}$$

where l stands for each class: $l = \{\text{fg}, \text{bg}\}$, ω_k is the mixture coefficient, $\mu_k \in \mathbb{R}^5$ and $\Sigma_k \in \mathbb{R}^{5\times5}$ are, respectively, the mean and covariance matrix of the k-th Gaussian distribution. As presented in the previous chapter, the spatial and color components of the SCGMM are considered decoupled, i.e., the covariance matrix of each Gaussian component takes the block diagonal form. With such decomposition, each foreground Gaussian component has the following factorized form:

$$G_l(z_i, \mu_k, \Sigma_k) = G(s_i, \mu_{k,s}, \Sigma_{k,s})\ G(c_i, \mu_{k,c}, \Sigma_{k,c}), \tag{5.2}$$

where $s_i \in \mathbb{R}^2$ is the pixel's spatial information and $c_i \in \mathbb{R}^3$ is its color value. The parameter estimation can be reached via Bayes' development, with the EM algorithm [DLR+77]. For this estimation an initialization frame is needed, containing a first segmentation of the foreground object.

5.2.2.1 Initialization and Updating

Once we have defined the Bounding box where the foreground and background models will work, the initialization of both models is done according to the object mask that is required as an input.

As in the previous chapter, the number of Gaussians that will compound each model should be slightly higher than the number of color-spatial regions of the foreground and background regions that appear within the ROI, to ensure that both

classes are correctly modeled with at least one Gaussian per region. We choose to analyze the RGB-histogram as explained in Section 4.4.1.1. Hence, we obtain a model with the correct number of Gaussians to represent the foreground object and the close background regions.

Once the number of Gaussians of each model is defined, we propose a fast two-steps initialization process that consists in:

- First, place the Gaussian distributions of the foreground and background models uniformly over the spatial region that corresponds to each model.

  We initialize the spatial and color domain of the Gaussians with the values of the pixels that are located within the region assigned to each Gaussian. Figure 5.4 displays a graphical and self-explicative example.

- Next,for each class, we use the Expectation Maximization (EM) algorithm ([DLR$^+$77]) in the overall five dimensional domain with the color-spatial information obtained from all the pixels belonging to the class we are analyzing, and located inside the ROI. This algorithm helps us to adjust the parameters of each Gaussian Mixture Model in the color and spatial domain, $\mu_{c,s}$ and $\Sigma_{c,s}$ of each model obtaining iteratively a maximization of the likelihood. Thanks to the spatially uniform distribution of the Gaussians, the initialization requires a few EM iterations to achieve the convergence of the algorithm and therefore, a correct representation of the foreground and background regions. A fix number of iterations equal to 3 yields correct results. Figure 5.4 shows the resultant initialization of the Gaussians in the spatial domain.

Figure 5.4: Initialization process. From left to right: spatially uniform distribution of the Gaussians, Foreground Gaussians after EM iterations and Background Gaussians after EM iterations. The spatial domain representation of the foreground Gaussians is in red color, background Gaussians are in green color.

Once each model has been correctly initialized, and for the next frames of the sequence, each model is updated with the foreground and background regions obtained from the previous segmentation according to the updating explained in Section 4.4.1.2. We assume a scene with moving background, moving foreground object as well as possible zoom effects of the camera, where new color-spatial regions of background and foreground classes inside the Region of Analysis appear in each

Figure 5.5: Example of foreground and background models. From left to right: input frame under analysis, background model and foreground model. Each ellipse represents one Gaussian of the SCGMM, colored with the mean color that each one is modeling.

frame. Thus, the spatial components of each Gaussian Mixture and also, the color ones, are updated after the classification in foreground and background of each frame. Figure 5.5 shows an example of foreground and background models for a frame under analysis, where one person is being segmented. As we can observe, the foreground and background models achieve a correct representation of the regions that each class is modeling.

5.2.3 Classification

Once the foreground and background models have been computed at frame $t-1$, a Bayesian labeling between foreground and background can be done for the frame t as proposed in Chapter 4.4.4, also by means of the energy functions that is solved for global optimum using a standard graph-cut algorithm [BVZ01].

5.3 Results

This section shows some tests to evaluate the quality and robustness of the proposed system. For this purpose, qualitative and quantitative evaluations have been performed. Quantitative results are obtained analyzing the cVSG public Data Base [TEBM08], which has been created by means of a chroma key, combining people to segment with different kind of background scenarios. We have compared it with the method proposed in [VM08]. Qualitative results are obtained analyzing another three different video sequences with different difficulty degree.

In Figure 5.6 the shirt of a running girl has been segmented. These results show how the shirt is correctly detected along the sequence despite the variability of the background regions. Moreover, in this sequence the evolution of the spatial foreground and background models along the sequence can be observed. Each ellipse is the graphical representation of each Gaussian distribution.

In Figure 5.7 the foreground segmentation of an skier can be observed. This

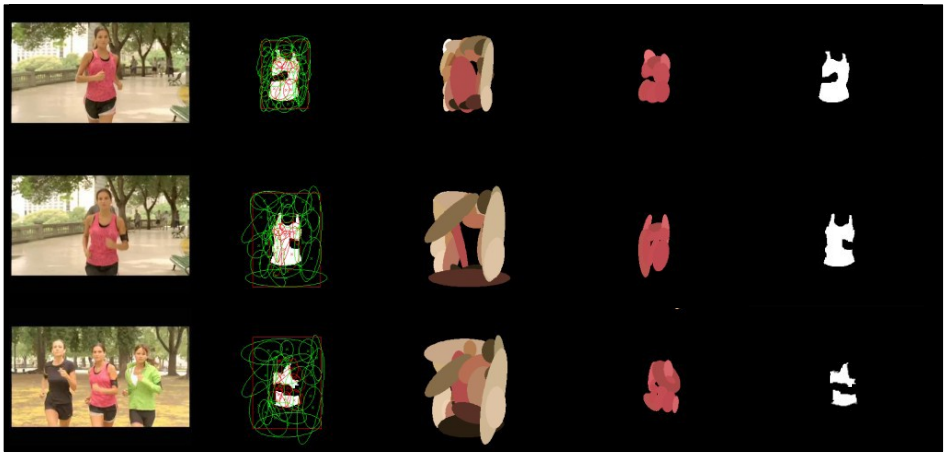


Figure 5.6: Results. Girl sequence. From left to right: original image, resultant mask with the Gaussians corresponding to spatial representation of the foreground model (red) and the background model (green), spatial representation of the background model (each Gaussian is colored with the mean color that it is modeling), spatial representation of the foreground model(each Gaussian is colored with the mean color that it is modeling), resultant foreground object mask.

sequence presents the following motion: object rotation, camera traveling and similarity between foreground and background regions. As it can be observed, the results show a correct definition of the foreground object segmentation despite the variability of the background regions, which are correctly assumed to the background model.

Figure 5.8 shows the results obtained in a F1 sequence that presents special difficulty due to object translation and rotation and the presence of other similar F1 cars within the area of analysis. It can be observed how the proposed system achieves a correct and robust object segmentation over these conditions, and adapts well to all these new regions that appear within the Dynamic Region of Analysis in each frame. Thanks to background model color and spatial updating, new background regions that appear in each frame, are incorporated into the background model before they affect the foreground model.

Table 5.1 shows the quantitative results using cVSG public database [TEBM08]. This database presents several sequences with different difficulty degree, depending on the background characteristics and the foreground to segment. We have used the full length of each sequence to compute the numerical results. The metrics used in the evaluation are: Precision (P), Recall (R) and f_{measure} metrics.

As it can be observed, the system proposed (Bayesian) achieves a high f_{measure} score in the overall data base although moving and dynamic background are present.

Regarding the computational cost, the system allows a speed of 1 frames/second, for a video sequence of 720x576 pixels with one object in scene, and using an Intel Xeon X5450 3.0GHz processor.

Table 5.1: Quantitative Results using cVSG Public Data Base [TEBM08]. In bold type the results corresponding to the best f_{measure} scores.

Sequence	Proposal	Precision	Recall	f_{measure}
Dancing (v.1 girl)	Bayesian	0.934	0.992	**0.962**
	[VM08]	0.933	0.975	0.954
Dancing (v.1 boy)	Bayesian	0.942	0.988	0.965
	[VM08]	0.953	0.987	**0.969**
Dangerous race	Bayesian	0.958	0.994	**0.975**
	[VM08]	0.935	0.935	0.935
Exhausted runner	Bayesian	0.986	0.985	**0.986**
	[VM08]	0.958	0.984	0.971
Bad manners	Bayesian	0.978	0.991	**0.984**
	[VM08]	0.931	0.891	0.910
Teddy bear	Bayesian	0.916	0.981	**0.948**
	[VM08]	0.953	0.939	0.946
Hot day	Bayesian	0.980	0.985	**0.983**
	[VM08]	0.959	0.958	0.958
Playing alone	Bayesian	0.997	0.984	**0.990**
	[VM08]	0.943	0.947	0.945

5.4 Conclusions

This chapter presents an application of the Bayesian region-based segmentation (presented in previous chapter) between foreground and background classes for moving camera scenarios, based on the use of the region-based spatial-color GMM to model the foreground object to segment and moreover, the close background regions that surrounds the object. We have proposed a framework for this kind of sequences that has allowed us to consider the probabilistic modeling of these close-background regions to achieve the classification of the pixels inside the ROI into the foreground and background classes within a MAP-MRF framework. The results show that the proposed system achieves a correct object segmentation reducing the false positives, and false negatives detections also in those complicated scenes where camera motion, object motion and camera zoom are present, as well as similarity between foreground and background colors.

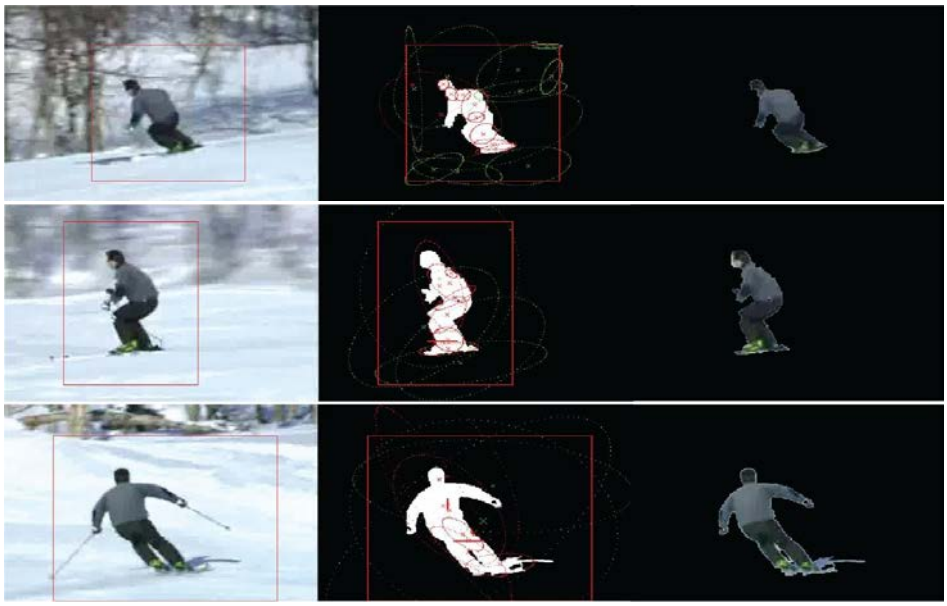


Figure 5.7: Results. Skier sequence. From left to right: original image, resultant mask with the ellipses corresponding to spatial representation of the foreground model (red) and the background model (green), the resultant mask colored with the original colors.

Figure 5.8: Results. F1 sequence. From left to right and from top to bottom: original image and the resultant object mask.

Part II

Proposals. Foreground Segmentation in Multi-Sensor Sequences

When we use more than one camera sensor to record the scene under analysis, the foreground segmentation process, and in consequence the posterior high level steps, can be improved by combining the camera sensors information thus, exploiting the redundancy that appears and is shared by the cameras. This redundancy depends on several factors, like for instance, the number of sensors that are recording the scene, their position and the kind of devices utilized in the acquisition set-up. How to exploit this extra-information, obtaining collaborative foreground segmentation methods is a non-trivial task, and has involved the work of many researchers along the years. In this part of the thesis we present some proposals developed to deal with this kind of sequences, in order to enhance the final segmentation results. Four proposals are presented in the following chapters:

- Foreground segmentation task in color $c = RGB$ + depth $d = Z$ multi-view sequences. In this chapter, a foreground segmentation system that combines these two sensors in a Bayesian Logarithmic Opinion Pool Decision framework is presented, in order to combine the probabilistic models used to characterize the foreground and the background classes for each one of the sensors.
- Multi-view foreground segmentation in smart-room scenarios:
  - Reliability maps applied to robust SfS volumetric reconstruction between foreground and background/shadow models.
    In this chapter, we use the the reliability maps of each sensor by computing the Hellinger distance between foreground and background/shadow models. The 2D reliability maps are used to obtain a robust SfS reconstruction.
  - Joint Multi-view Foreground Segmentation and 3D Reconstruction with Tolerance Loop.
    A loop between foreground segmentation and 3D reconstruction is proposed in this research line by updating the foreground model, defined in each view, with the conservative 3D volume reconstruction of the object in an iterative way.
  - 3D Foreground probabilistic model.
    A foreground model designed in the 3D space is proposed in this chapter. Monocular 2D segmentation projecting the 3D model to each 2D view is used in order to obtain the 2D foreground masks.

Chapter 6

Foreground Segmentation in Color-Depth Multi-Sensor Framework

6.1 Introduction

New devices suitable for capturing the depth of the scene, which have been developed in the recent years, are creating a new trend on the foreground segmentation area towards the new available depth information of the scenes. ToF and structured light depth cameras are an example of this kind of devices that have offered an alternative to the stereo systems and their complex problems in the disparity estimation. In this chapter we propose a system to combine color and depth sensors information, in a probabilistic framework between foreground and background classes, in order to improve the foreground segmentation results taking advantage of the possibilities that each one of the sensors offers.

6.2 State of the Art

For several years, many authors have been working in foreground segmentation using static color camera devices. We have seen some examples of these proposals in previous chapters. Despite foreground modeling methods improve the performance of the color foreground segmentation, all these methods present problems when foreground objects have similar color to the background, the camouflage problem, or when lighting or shadow affect the foreground and background.

Depth data allows a more robust segmentation of the object of interest towards the color camouflage problem than the systems based on color segmentation: [GLA+08, SS11] use a pixel-wise exception to background segmentation using MoG background model, while [XCA11] defines a depth region growing in a depth thresholding condition.

Nevertheless, as seen in Chapter 3, all these systems present other problems concerning the segmentation with depth sensors: lack of precision in the segmentation due to the noisy and low resolution depth maps obtained by the sensors, and presence of camouflage errors when foreground and background present similar depth. How to solve these problems combining both color and depth sensors is an important topic in order to achieve a precise and robust objects segmentation that uses the best characteristics of each sensor, avoiding, as far as possible, color and depth camouflage problems. Some authors have proposed some solutions in this research line combining depth thresholding segmentation and detection refinement:

In the method explained in Section 3.2.2.1.1 ([CTPD08]), the authors propose a simple depth thresholding segmentation followed by a color-depth trimap analysis to improve the precision of the segmentation in the borders of the object. In the proposal reviewed in the same section ([FFK11]), the authors use a thresholding technique to separate foreground from the background in multiple planes, and a posterior trimap refinement to reduce the artifacts produced by the depth noise. In [WBB08] the depth thresholding segmentation allows to automatically obtain a pentamap that is used to make a more efficient color graph cut regularization.

These kind of methods allow to obtain correct results under limited constraints on the scenario set and the depth thresholding, but present some segmentation errors in presence of difficult situations provided by depth camouflage problem.

Other authors have addressed the problem trying to combine the color and depth sensors in a more robust framework:

In [BW09] the authors propose a color-depth Mean Shift segmentation system, of the overall image, based on the depth noise analysis in order to weight the depth reliability, while the proposal reviewed in Section 3.2.2.1.2 ([SK11]) uses a pixel-wise probabilistic background model in color-depth domain, to perform a more complete exception to background segmentation. Although these kind of methods present a more robust and general framework in front of camouflage situations than the thresholding approaches, they still present some problems for correctly combining the color and depth sensors information when camouflage situations appear because of the lack of foreground objects information to detect it and thus, to improve the final segmentation results. In this way, [WZYZ10] proposes a probabilistic fusion framework between foreground and background classes for color and depth cues, which achieve correct results in close-up sequences. The probabilistic models of each

one of the classes are combined according to the foreground-background histogram similarity.

Combining different sensors for foreground segregation is an important topic in order to achieve a precise and robust objects segmentation. In this area, [PL04] propose a sensor fusion combination based on Multi Bayesian utility functions, while several authors in statistics have well addressed the fusion of information provided by several sensors in a Bayesian framework ([SA99, Kun04]).

In this Chapter, we present a foreground segmentation system that belongs to the last group of proposals. We propose a system that combines in a probabilistic framework both, color and depth sensors information to perform a more complete Bayesian segmentation between foreground and background classes. The system, suitable for static color-depth camera sequences in a close-up and long-shot views, achieves a correct segmentation results taking into account the spatial context of the models showing a combination of color-spatial and depth-spatial region-based models for the foreground and a color and depth pixel-wise models for the background in a Bayesian Logarithmic Opinion Pool decision model. In order to improve the foreground segmentation precision, we add a final segmentation refinement based on a trimap analysis.

All the sequences used in this chapter has been recorded by means of the kinect sensor, developed by Microsoft, combined with the OpenNI SDK configured with the factory calibration presets, in order to obtain the synchronized and registered color and depth maps of the sequences under analysis.

6.2.1 Proposed System

We propose a segmentation system that exploits the advantages of each sensor type. With this aim we present an algorithm where parametric foreground models in color-space and depth-space domains are evaluated against pixel-wise color and depth background models. The improvements of the proposed method in the segmentation process are twofold: first, we combine probabilistic models of color, space and depth, in order to obtain a correct pixel classification according to the color and depth sensors and in a posterior step, we correct the errors of precision that the depth sensor introduces to the overall process and are converted into some false positive detections in the borders of the foreground object. Once the approximate position of the borders in the current image is known, it is better to disregard the depth information at those positions where the color sensor provides enough discrimination. The reliability of the sensors, based on the Hellinger distance ([Ber77]) between foreground and background models is used in both stages.

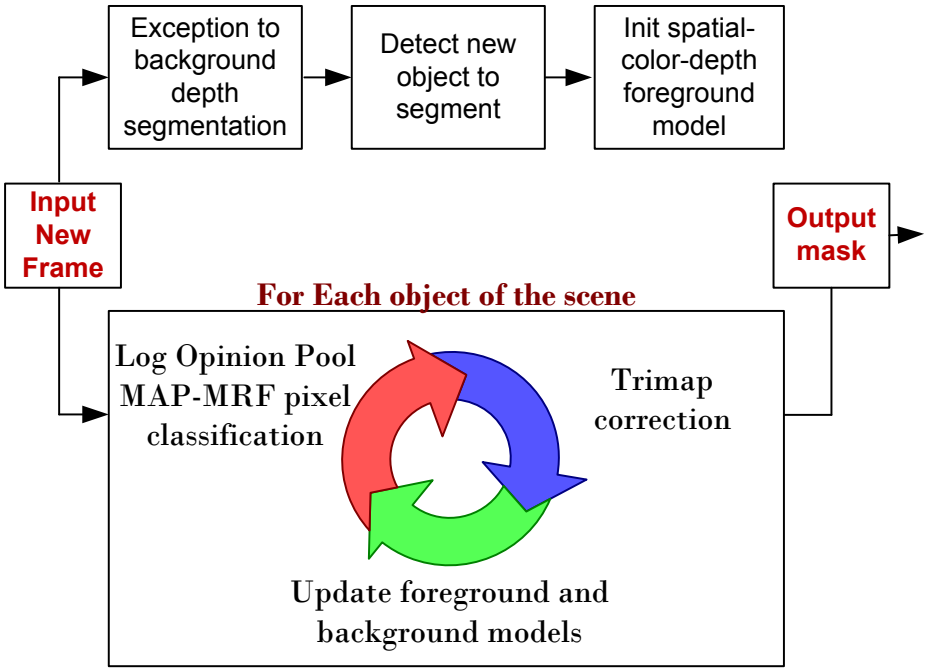


Figure 6.1: Work flow of the system.

Figure 6.1 shows the overall work-flow of the system. The main modules are:

- Foreground and background models initialization (Section 6.3): Automatic initialization of the foreground and background models is used by means of a simple exception to background segmentation in the depth domain.
- Logarithmic Opinion Pool sensor fusion and Bayesian pixel classification (Section 6.4 and Section 6.5) respectively: If we consider that the informations of each sensor are not correlated between them, we can assume a sensor fusion system where team members are allowed to exchange sensor information. In order to achieve a correct combination of the color, spatial and depth domains of each class, Logarithmic Opinion Pool sensor fusion is used. We propose to test the reliability of the information that each sensor can add to the overall team posterior probability, in order to fuse them, by using the Hellinger distance to evaluate the distance between foreground and background models in each sensor, thus maximizing the reliability of the final decision.
- Improve the precision of the segmentation using a trimap approach (Section 6.6): A final correction step is proposed in order to reduce the errors that the depth sensor generates in the borders of the object. After the pixel classification step into foreground and background classes, a trimap segmentation of the pixels of the frame in analysis is defined as background, foreground and unknown. We define the unknown region as all the pixels that appear inside a boundary of the object to segment, which can present segmentation errors.

- Foreground and background models updating, (Section 6.3): After the final labeling, we adapt the foreground and background models to the variations that appear in the object's movements and the background regions respectively.

The remainder of the Chapter is organized as follows: Section 6.3 describes the foreground and background probabilistic models. The Logarithmic Opinion Pool decision model used for color and depth sensor fusion is explained in Section 6.4. Section 6.5 is devoted to the Bayesian pixel classification. Section 6.6 addresses the proposed trimap correction to solve the false positive detections that the depth sensor produces. Finally, some results and conclusions are presented in Section 6.7 and Section 6.8 respectively.

6.3 Probabilistic Models

Since we want to achieve a correct foreground segmentation in static color-depth sequences, specific probabilistic models are used to represent the foreground and background classes for the color and depth sensors. We use two pixel-wise Gaussian models for the background: one for the color and another for the depth domains, and two region based models for the foreground: Spatial-Color GMM and Spatial-Depth GMM for the foreground. Therefore, for each frame of the sequence I_t, our objective is to obtain for each class an updated model parameter set θ that maximizes the data likelihood:

$$\theta_l = \arg\max_{\theta_l} \prod_{x_{i,l} \in I_{t,l}} \left[P(x_{i,l}|\theta_l) \right], \tag{6.1}$$

where l stands for classes $\{\text{fg}, \text{bg}\}$ and $x_i \in \mathbb{R}^6$ is the input feature vector for pixel i in the $x = (RGB\ XY\ Z)$ domain. Hereinafter we refer to the color, spatial and depth domains as: $c = RGB \in \mathbb{R}^3, s = XY \in \mathbb{R}^2$ and $d = Z \in \mathbb{R}$ respectively.

6.3.1 Background Model

A spatially precise background model is used in order to obtain a description in color and depth domains. The model consists of two independent Gaussians per pixel, one in the RGB domain and the second one in the Z domain.

The likelihood of the color Gaussians is defined as:

$$\begin{aligned} P(c_i|\text{bg}) &= G_{\text{bg}}(c_i, \mu_{c,i}, \Sigma_{c,i}) \\ &= \frac{1}{(2\pi)^{\frac{3}{2}} |\Sigma_{c,i}|^{\frac{1}{2}}} \exp\left[-\frac{(c_i - \mu_{c,i})^T \Sigma_{c,i}^{-1} (c_i - \mu_{c,i})}{2} \right], \end{aligned} \tag{6.2}$$

where $c_i \in \mathbb{R}^3$ is the input vector of the i-th pixel, $\mu_{c,i} \in \mathbb{R}^3$ and $\Sigma_{c,i} \in \mathbb{R}^{3\times3}$ are, respectively, the mean and covariance matrix of the Gaussian distribution.

For the depth domain, the likelihood of the Gaussian is defined as:

$$\begin{aligned} P(d_i|\text{bg}) &= G_{\text{bg}}(d_i, \mu_{d,i}, \sigma_{d,i}) \\ &= \frac{1}{(2\pi)^{\frac{1}{2}}\ \sigma_{d,i}}\ \exp\left[-\frac{(d_i-\mu_{d,i})^2}{2\sigma_{d,i}}\right], \end{aligned} \tag{6.3}$$

where $d_i \in \mathbb{R}$ is the input depth value of the i-th pixel, $\mu_{d,i} \in \mathbb{R}$ and $\sigma_{d,i}$ is the standard deviation.

Analogously to Chapter 4, we extend the pixel-wise background models, to a region based model in color and depth domains in order to make comparable the background models with the foreground:

$$\begin{aligned} P(c_i, s_i|\text{bg}) &= \delta(s_i - \mu_{s,i})\ G_{\text{bg}}(c_i, \mu_{c,i}, \Sigma_{c,i}), \\ P(d_i, s_i|\text{bg}) &= \delta(s_i - \mu_{s,i})\ G_{\text{bg}}(d_i, \mu_{d,i}, \sigma_{d,i}), \end{aligned} \tag{6.4}$$

where δ denotes the Kronecker delta and therefore, we are using one Gaussian per pixel centered in space at the pixel position ($\mu_{s,i}$) with a zero spatial variance.

Initialization The color and depth pixel-wise models are initialized in a learning step by using a short sequence free of foreground objects.

Updating

When a pixel value is classified as background, its model is updated, in color and depth domains, in order to adapt it to progressive image variations.

Both initialization and updating processes follow the Running Gaussian average model ([WADP02]),

6.3.2 Foreground Model

We use two parametric region-based foreground models that combine color, space and depth domains. We propose the Spatial Color Gaussian Mixture Model (SCGMM) and the Spatial Depth Gaussian Mixture Model (SDGMM) in order to obtain a reliable probabilistic representation of the foreground pixels for the color and depth sensors respectively. The likelihood of pixel i for the color sensor is then:

$$P(c_i, s_i|\text{fg}) = \sum_{k=1}^{K_{\text{fg}}} \omega_k G_{\text{fg}}(c_i, s_i, \mu_{k,c,s}, \Sigma_{k,c,s}), \tag{6.5}$$

For the depth domain the likelihood is defined as:

$$P(d_i, s_i|\text{fg}) = \sum_{k=1}^{K_{\text{fg}}} \omega_k G_{\text{fg}}(d_i, s_i, \mu_{k,d,s}, \Sigma_{k,d,s}), \tag{6.6}$$

where ω_k is the mixture coefficient of the k-th Gaussian distribution. In order to simplify the design, we impose both models to have the same number of Gaussians (K_{fg}) with the same spatial distribution. We assume that the spatial component is decoupled from the color and depth domains thus, we will be able to speed up the computational problem designing a parallel implementation.

With such decomposition, each color and depth foreground Gaussian component has the following factorized form:

$$\begin{aligned} G_{\text{fg}}(c_i, s_i, \mu_{k,c,s}, \Sigma_{k,c,s}) &= G_{\text{fg}}(s_i, \mu_{k,s}, \Sigma_{k,s})\ G_{\text{fg}}(c_i, \mu_{k,c}, \Sigma_{k,c}), \\ G_{\text{fg}}(d_i, s_i, \mu_{k,d,s}, \Sigma_{k,d,s}) &= G_{\text{fg}}(s_i, \mu_{k,s}, \Sigma_{k,s})\ G_{\text{fg}}(d_i, \mu_{k,d}, \Sigma_{k,d}), \end{aligned} \tag{6.7}$$

where $G_{\text{fg}}(c_i, \mu_{k,c}, \Sigma_{k,c})$ and $G_{\text{fg}}(d_i, \mu_{k,d}, \Sigma_{k,d})$ are defined as in the equations 6.2, 6.3 respectively, and

$$G_{\text{fg}}(s_i, \mu_{k,s}, \Sigma_{k,s}) = \frac{1}{2\pi|\Sigma_{s,i}|^{\frac{1}{2}}}\ \exp\left[-\frac{(s_i - \mu_{k,s,i})^T \Sigma_{k,s,i}^{-1}(s_i - \mu_{k,s,i})}{2}\right], \tag{6.8}$$

where $s_i \in \mathbb{R}^2$ is the input vector of the i-th pixel, $\mu_{s,i} \in \mathbb{R}^2$.

6.3.2.1 Initialization

Analogously to Chapter 4, the initial parameter estimation can be reached via Bayes' development with the EM algorithm ([DLR+77]). An initial segmentation of the foreground object is required in order to initialize the foreground model. We propose to use an initial exception to background segmentation in the depth domain to achieve this first detection of the object because it is more robust to color camouflage problems, which are more common than the depth ones. Once one foreground connected component is detected, it has to present a minimum size and some temporal correspondence along the sequence to be considered as an object and continue with the foreground model initialization. Next, we estimate how many Gaussians are needed for correctly modeling the object to segment in the color domain. Analogously to the method proposed in Chapter 5, Section 5.2.2.1, we analyze the color histogram for this purpose and initialize it in a fast way, first distributing the Gaussians uniformly in the spatial domain within the foreground object and later, using few iterations of the EM algorithm in the $z = RGB\ XY$ domains. Once the Color Spatial Gaussians are correctly initialized, we initialize the SDGMM taking the same number of Gaussians and Spatial distribution than the SCGMM, and assigning to each Depth Gaussian the mean and variance depth of the spatial region that this Gaussian is modeling. The advantages of this configuration is twofold: it is useful to achieve a correct spatial distribution of the depth model in order to adapt well the model to the different parts of the object and their movements, thus reducing the false positive errors, and to speed up the process

using just one spatial initialization and updating processes for both color and depth models.

6.3.2.2 Updating

As in previous chapters, in order to adapt the foreground models to these displacements, we propose to update the components of the Gaussian Mixtures in the color, space and depth domains in a two-steps updating:

Spatial domain updating: We use the pixels classified as foreground to update only the spatial components of the Gaussian Mixtures. We assign each pixel to the Gaussian k that maximizes:

$$P(k|x_i, \text{fg}) = \frac{P(x_i|\text{fg}, k)}{\sum_k P(x_i|\text{fg}, k)} = \frac{P(x_i|\text{fg}, k)}{P(x_i|\text{fg})}, \tag{6.9}$$

where $P(x_i|\text{fg})$ is the likelihood that the color and depth sensors combination present for the pixel i (will be defined in Section 6.4), and $P(x_i|\text{fg}, k)$ is the likelihood of both sensors given by the Gaussian k. Once each pixel has been assigned to a Gaussian, the spatial mean and covariance matrix of each Gaussian are updated with the spatial mean and variances of the region that it is modeling.

Also, in order to achieve a better adaptation of the model into the silhouette of the object, we apply a Gaussian split criterion according to the spatial size of the Gaussian (Section 4.4.1.2).

Color domain updating: once the spatial components of the Gaussians have been updated, we update the foreground SCGMM according to the color domain. For each foreground Gaussian, we check if the data that it is modeling (according to the pixels assigned to this Gaussian) follows a Gaussian distribution. Otherwise, a new Gaussian is created to correctly model this region.

Depth domain updating: In order to adapt the foreground model to the depth variations of the object, we perform a complete depth updating of each Gaussian of the SDGMM with the mean and variance depth that the region assigned to this Gaussian presents. When regions without depth information appear in the depth map due to sensor errors in the depth detection process, we identify these non reliable pixels, thus avoiding to use this information in the updating process and in the next classification step.

6.4 Sensor Fusion Based on Logarithmic Opinion Pool

Given the set of sensors $J \equiv \{\text{color,depth}\}$ that are recording different data from the scene, our aim is to correctly combine the information that we receive from each one in order to maximize the robustness of the foreground segmentation in each one of the frames, resolving the possible inconsistencies that can appear among them. In this way, we design a Logarithmic Opinion Pool framework for combining the sensors' information, extensible to any kind of image capturing sensors ([SA99, Kun04]). The task of this decision maker is, in the first instance, to combine probabilistic information from all the sources and then to make decisions based on the global posterior. According to the Bayesian theory and assuming that we have some knowledge of foreground, and background prior probabilities, $P(\text{fg})$ and $P(\text{bg})$ respectively, we can define the global posterior of the color and depth sensors as:

$$P(l|x_i) = \frac{P(x_i|l)P(l)}{P(x_i)} \propto P(x_i|l)P(l), \tag{6.10}$$

where $l \in \{\text{fg,bg}\}$ and $i \in I_t$ stands for the pixel in analysis. The normalizing denominator is the same for foreground and background and, thus, can be disregarded.

How to combine the different likelihoods $P(x_i|l)$ of each one of the sensors is the most important part of the combiner. A basic product formulation of the likelihoods has the drawback that a single close to zero probability in one of the sensors leads to the cancellation of the overall combination. In order to avoid this zero probability problem, which could lead to important misclassification errors, we use the Logarithmic Opinion Pool that matches with the Consensus theory ([Kun04]). Hence, we can formulate the global multi-sensor likelihood as follows:

$$P(x_i|l) = P(c_i, s_i|l)^{\alpha_{c,i}} \cdot P(d_i, s_i|l)^{\alpha_{d,i}}, \tag{6.11}$$

where $\alpha_{c,i} \in \mathbb{R}$ and $\alpha_{d,i} \in \mathbb{R}$ are the weighting factors for the color and depth likelihoods for the i-th pixel and accomplish $\alpha_{c,i} + \alpha_{d,i} = 1$.

Taking logarithms in the above expression leads to the following log-likelihood expression:

$$\log P(x_i|l) = \alpha_{c,i} \log P(c_i, s_i|l) + \alpha_{d,i} \log P(d_i, s_i|l), \tag{6.12}$$

As we can observe, the definition of the weighting factors is central to the correct working of the sensor fusion system.

6.4.1 Weighting Factors

We define the weighting factors according to the reliability that each one of the sensors presents. For that, we propose to analyze the similarity between foreground and background classes for each one of the sensors, assuming that:

-High similarity implies that both classes are modeling the same space in a camouflage situation, and thus, the decision is not reliable.

-Small similarity implies classes separated enough to achieve a correct decision.

Hence, for each one of the image pixels $x_i \in I_t$, we propose to compute the Hellinger distance ([Ber77]), to detect the degree of similarity between foreground and background models that the sensors present in the color and depth domains:

$$H_i^j(q_{\text{fg},i}^j, q_{\text{bg},i}^j) = \sqrt{1 - \text{BC}_i^j}, \tag{6.13}$$

where $0 \leqslant H(q_{\text{fg},i}^j, q_{\text{bg},i}^j) \leqslant 1$, $q_{\text{fg},i}^j$ and $q_{\text{bg},i}^j$ are the p.d.f.'s that model the i-th pixel of the jth-sensor color or depth for the foreground and background classes respectively. BC is the Bhattacharyya Coefficient, which is formulated, for a multivariate Gaussian distribution, as follows:

$$\text{BC}_i^j = \frac{1}{\left(\frac{|\Sigma_i^j|}{\sqrt{|\Sigma_{\text{fg},i}^j||\Sigma_{\text{bg},i}^j|}}\right)^{\frac{1}{2}}} \exp\left[-\frac{(\mu_{\text{fg},i}^j - \mu_{\text{bg},i}^j)^T(\Sigma_i^j)^{-1}(\mu_{\text{fg},i}^j - \mu_{\text{bg},i}^j)}{8}\right], \tag{6.14}$$

where $0 \leqslant \text{BC}_i^j \leqslant 1$, $\Sigma_{\text{fg},i}^j$ and $\Sigma_{\text{bg},i}^j$ are the covariance matrices of the models associated to the i-th pixel, for the jth-sensor of the foreground and background classes respectively. $\mu_{\text{fg},i}^j$ and $\mu_{\text{bg},i}^j$ are the mean vectors of each class, and $\Sigma_i^j = \frac{\Sigma_{\text{fg},i}^j + \Sigma_{\text{bg},i}^j}{2}$.

Note that $H(q_{\text{fg},i}^j, q_{\text{bg},i}^j) = 0$ means that foreground and background models are equal, and thus, strong camouflage situation is present in this pixel, and otherwise $H(q_{\text{fg},i}^j, q_{\text{bg},i}^j) = 1$ implies that both models are completely different and there is not similarity between them.

Since we are working in the foreground class with the color and depth foreground models (SCGMM and SDGMM), $\mu_{\text{fg},i}^j$ and $\Sigma_{\text{fg},i}^j$ will be chosen according to the Gaussian k that maximizes the probability of the i-th pixel under analysis (equation (6.9)). In the case of the background class, since we have defined a pixel-wise model, $\mu_{\text{bg},i}^j$ and $\Sigma_{\text{bg},i}^j$ will be directly obtained from the background Gaussians associated to this pixel.

Hence, we design the α_i^j weights as:

$$\alpha_i^j = \frac{H(q_{\text{fg},i}^j, q_{\text{bg},i}^j)}{\sum_{j \in J} H(q_{\text{fg},i}^j, q_{\text{bg},i}^j)}, \tag{6.15}$$

As we can observe, sensors that present a high degree of similarity between foreground and background classes, will have a close to zero α_i^j weight that will be equivalent to inhibit the sensor from the decision maker, thus avoiding misclassification errors in front of camouflage problems.

The Hellinger distance present two main characteristics that are very interesting for this application: Unlike the Bhattaharyya Distance (BD), or the Kullback-Leibler divergence (KL) ([Kul87]), which give us a similarity distance bounded between $[0, \infty)$, the Hellinger distance allows us to achieve a normalized distance among models bounded between $[0, 1]$. Moreover, unlike the Kulback-Leibler divergence, the Hellinger distance is symmetric and thus, $H(q_{\text{fg}}, q_{\text{bg}}) = H(q_{\text{bg}}, q_{\text{fg}})$.

6.5 Pixel Classification

Once the combined likelihoods for each class and sensor has been decided in each pixel of the image, a Bayesian pixel classification between foreground and background classes is used to obtain the resultant foreground segmentation. In each frame I_t, a pixel i may be assigned to the class $l = \{\text{fg,bg}\}$ that maximizes $P(l|x_i) \propto P(x_i|l)P(l)$,

where $P(x_i|l)$ is obtained from the Logarithmic Opinion Pool decision (Section 6.4), and foreground and background prior probabilities $P(l)$ are calculated according to the percentage of the image that each class present in frame $t-1$.

Analogously to previous chapters, we introduce the spatial context in the segmentation decision by using the graph-cut algorithm for the pixel labeling.

6.6 Trimap Analysis

The depth sensor presents a lack of precision in the depth estimation that causes many false positive detection errors in the contours of the object. In this final step, we propose to correct this specific error, by applying a trimap correction in these areas. Since color segmentation presents a more precise segmentation, we propose to define an uncertainty area that contains the contour region of the object, susceptible of presenting these detection errors, and try to apply a more precise color foreground detection to correct them.

Therefore, given S, the number of pixels labeled as foreground in the previous classification step, we obtain the subgroup S_u corresponding to the uncertainty pixels of the contour as:

$$S_u = S \cap \text{dil}(\bar{S}, D), \tag{6.16}$$

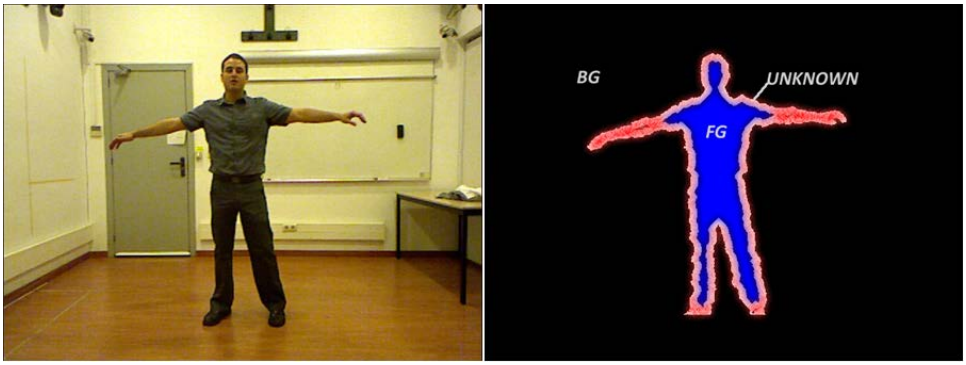


Figure 6.2: Image segmentation into a trimap among Foreground (fg), Background (bg) and Unknown regions.

where dil operator refers to a morphological dilation, D is the 8x8 structuring element, and $\bar{S}$ denotes the complementary region of S and thus, the background detection.

We consider a trimap where background regions are outside of the uncertainty area, and the foreground region is the area inside it. Figure 6.2 shows an example of the uncertainty area.

Once the uncertainty area is defined, a new labeling classification in this region is applied according to the reliability of the color sensor, which will allow us to correct the errors generated by the depth sensor. Hence, for all the pixels of this region, we will use only the color sensor information when it presents no color camouflage between foreground and background, according to:

$$H_{i,c}(q_{\mathrm{fg},i,c}, q_{\mathrm{bg},i,c}) > H_{\mathrm{th\_max}}, \tag{6.17}$$

where $H_{\mathrm{th\_max}}$ are the thresholds used to determine if the sensor is reliable. In our experiments, $H_{\mathrm{th\_max}} = 0.7$ yield correct results since it ensures that the foreground and background color models present enough distance each other to consider the pixel as reliable.

Figure 6.3 shows a graphical example of the final uncertainty regions that will be analyzed in order to improve the precision of the final segmentation. As we can observe, the uncertainty area is only taken into account when foreground and background do not present any color similarity. Otherwise, the uncertainty is removed to maintain the original segmentation.

Note that the spatial updating of the foreground model help us to obtain a precise model of the object at the contours, achieving a better color decision in the borders.

(a)

(b)

Figure 6.3: Example of unknown region. a) shows an example where foreground and background present color similarity. b) displays another example without foreground-background similarity.

6.7 Results

Qualitative and quantitative results have been obtained in order to evaluate the proposed system. We have analyzed our own database, which consists of nine single person sequences, recorded with a kinect device, to show depth and color camouflage situations that are prone to errors in color-depth scenarios.

Quantitative results are obtained analyzing the nine sequences of our own database. Qualitative results have been obtained analyzing three difficult sequences from this database.

In these results, we present a comparison between the proposed method (denoted as 'LogPool' in the Figures) with the Running Gaussian Average pixel-wise segmentation method proposed in [WADP02] and applied to the RGB color domain (RGA-RGB) and the depth domain (RGA-DEPTH). Moreover, in order to evaluate the improvement against a region based system, we evaluate the color domain segmentation presented in [GPH09] (SCGMM) in our comparison. Finally, we also analyze the results obtained using the segmentation system that has been described in Section 3.2.2.1.2 ([SK11]) (Schiller) that combines color and depth information. Since this method is based on the ToF sensor, and its effects over the borders of the objects, we adapt it to the kinect sensors by using the uncertainty area that the we

Figure 6.4: Qualitative results of sequence 1. From left to right: original sequence recorded by the color and depth sensor, pixel-based fg detection in the depth domain [WADP02], pixel-based fg detection in the color domain, region-based foreground segmentation using the method proposed in Chapter 4, color and depth segmentation obtained using [SK11] and the results obtained by our proposal by combining color and depth sensors.

obtain in the borders of the object to apply the segmentation technique.

The metric used in the quantitative evaluation is the f_{measure}, which gives us the relationship between the Precision (P) and Recall (R) results for each frame.

Figure 6.4 shows the results where depth camouflage problems appear when the person of interest sits down in a chair. As we can observe in the third column, pixel-wise segmentation in the depth domain obtains some false positive detections due to the lack of precision and the projection problems of the depth sensor. Moreover, when the person sits on the chair, camouflage depth problem arises and the segmentation of the object is completely lost. When using color segmentation in an exception to background framework and in the region-based approach (fourth and fifth column respectively), some false negatives appear due to the color camouflage problem. Despite of this, the segmentation is not lost in front of depth camouflage problems. Our approach (last column) improves the results resolving the camouflage depth situation and reducing the false negative detections that each one of the sensors adds to the segmentation. Quantitative results for each one of the frames are displayed in Figure 6.5. As we can observe, the pixel-wise method in the depth domain (RGA-DEPTH) presents a high number of False negative errors when the person sits down on the chair (region 'A'). In region 'B', the RGA-RGB and RGA-DEPTH segmentation increases the number of false positive and false negative detections since the person interacts with the background, modifying the setup. Unlike pixel-wise methods, the region based system SCGMM allow us to obtain a more robust segmentation in the color domain thanks to the presence of a foreground model in the segmentation. The results obtained by the Schiller method present some false positive and negative errors when the person sits on the chair, but unlike other pixel-wise approaches, it maintains the robustness of the segmentation in this depth camouflage situation. Our proposal can overcome the depth camouflage problem that appear in this sequence thanks to the robust combination

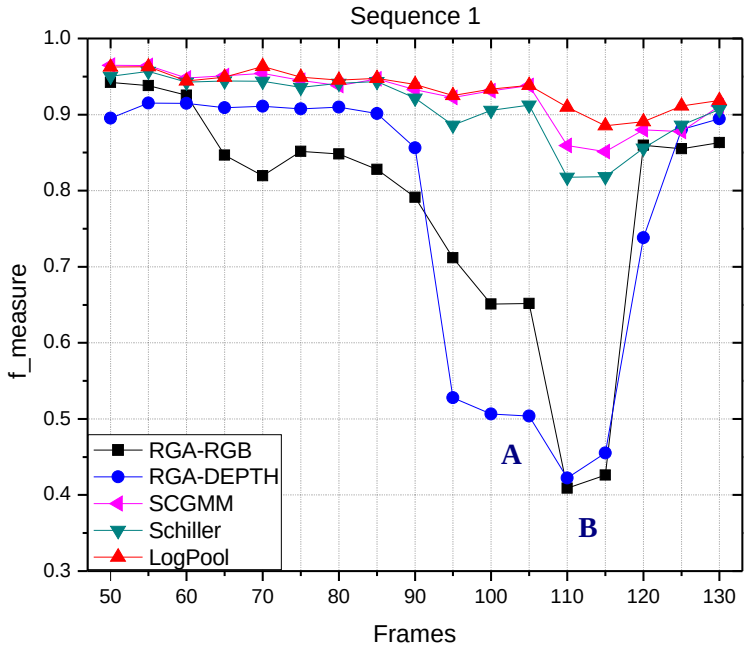


Figure 6.5: Quantitative results of sequence displayed in Figure 6.4. The sequence presents a depth camouflage situation in the frames surrounding the region 'A' and region 'B', and a background setup modification in region 'B'.

of both color and depth sensors, which detects the depth camouflage situation and then omits the depth sensor decision for those frames.

Figure 6.8 displays the results obtained in a smart-room sequence where depth camouflage problems appear when the person of interest is close to the wall getting its depth value. As we can observe in the second column, depth pixel-wise segmentation fails in this situation completely losing the object detection. When using pixel-wise color segmentation in third column, some false detections appear due to illumination variations, while in region based color segmentation (fourth column), small false positive detections appear due to these illumination changes and shadows. The segmentation obtained by the Schiller method, in fifth column, achieves a segmentation result that also presents these some false positive detections in the contours of the object. Sixth column shows the results obtained by our system. As we can see, we improve the segmentation of the object, taking the most of each sensor, avoiding depth camouflage errors and illumination variations. Note that some spurious detections that can appear in the segmentation process can be easily removed by using a simple area filtering. The quantitative evaluation showed in Figure 6.6, show us that the system based only on the depth segmentation presents a f_{measure} close to 0.3 in the frames involving the depth camouflage situation (Region 'A'), while our method maintains the segmentation quality along the sequence

with f_{measure} values between 0.85 and 0.97.

Figure 6.9 shows a sequence where foreground and background present similar color. As we can see, methods based only on color segmentation present many false negative errors due to the difficulty of correctly classifying between both foreground and background classes (third and fourth columns). When using only depth segmentation (second column), the detection of the object is correct, although some false positive errors still appear in the segmentation, especially in the borders of the object. The method proposed by Schiller (fifth column), segments the uncertainty area in the contours of the object analyzing only the color information and because of that it presents many false negative errors in this area. Our results (Sixth column), achieves a segmentation that correctly solves the color camouflage problem and improves the precision in the borders of the object thanks to the trimap enhancement step, thus maintaining the correct detection of the object. Figure 6.7 shows the quantitative evaluation along the sequence.

In order to observe the effects of the trimap refinement over the contours of the object, Figure 6.10 displays the segmentation results of two frames before and after applying the trimap analysis. As we can observe, the trimap enhancement improves the precission of the final segmentation mask by removing the false positive detections generated by the depth detection errors due to the characteristics of the acquisition setup.

Finally, Table 6.1 shows the quantitative results according to the f_{measure} score obtained in the nine sequences of our database. As we can observe, pixel-wise methods RGA-RGB and RGA-DEPTH give in general low scores due to camouflage situations of the sequences and the noisy illumination environment. The SCGMM method gives correct results when the foreground model can be correctly initialized in sequences with low foreground-background similarity. The Schiller method achieves low scores, in general, due to the simple pixel-wise segmentation method used for the color and depth domains. Moreover, this method can not reach an effective borders correction in those sequences where the color camouflage is present, but it achieves stable results around 0.86 for all kind of camouflage situations. In the other hand, our system achieves high scores in both color and depth camouflage situations allowing a robust segmentation for these kind of scenarios.

Regarding the computational cost, our system allows a speed of 0.3 frames/second, segmenting a kinect video sequence of 640x480 pixels with one object in the scene, and using an Intel Xeon X5450 3.0GHz processor. It uses 300 MB of RAM without implementing any memory optimization. Parallel CUDA or OPENCL implementation can be used to improve the speed rates.

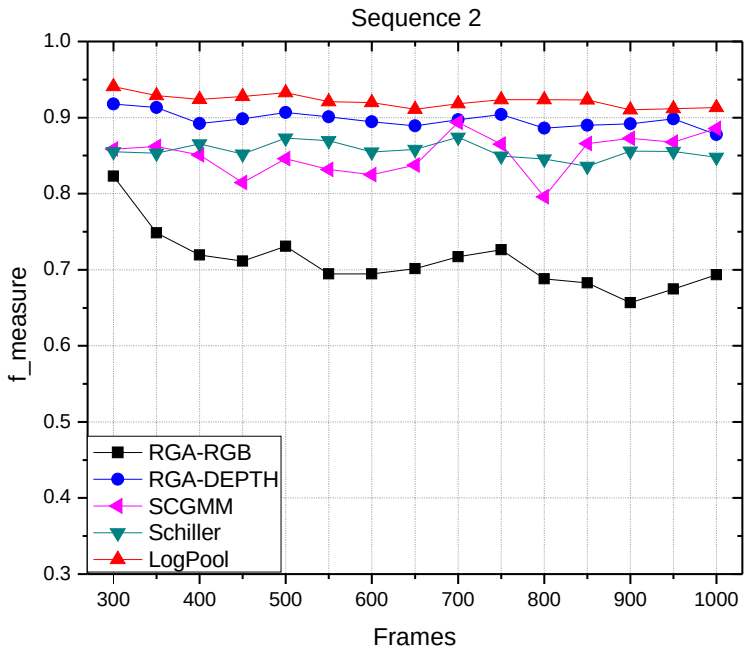


Figure 6.6: Quantitative results of the sequence displayed in Figure 6.8. Frames surrounding 'A' present a depth camouflage situation between foreground and background.

6.8 Conclusions

We have presented in this chapter a foreground segmentation system that combines color and depth sensors information in a Bayesian Logarithmic Opinion Pool framework. We propose a Spatial Color GMM and a Spatial Depth GMM to model the foreground, and two pixel-wise Gaussian models to model the color and the depth background domains. Those models are combined by using the Logarithmic Opinion Pool and the Hellinger distance in order to achieve a correct and robust classification of the pixels of the scene. Our system is robust in front of color and depth camouflage problems between the foreground object and the background, and also improves the segmentation in the area of the objects' contours by reducing the false positive detections that appear due to the lack of precision of the depth sensors.

Since we are using a probabilistic region-based model to model the color and depth information of the object, the quality of the foreground segmentation will depend on its correct initialization and the correct modeling of all the regions of the object. Therefore, both, the initialization and the updating processes are of main importance in order to ensure that the foreground model adapts correctly to the changes and movements of the object along the sequence.

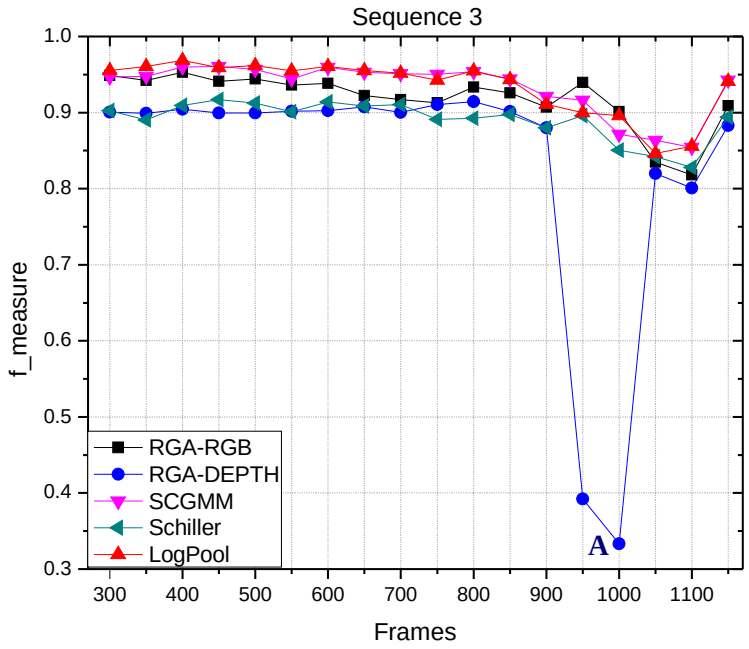


Figure 6.7: Quantitative results of the sequence depicted in Figure 6.9. Color similarity between foreground and background classes is present along the sequence.

The complexity of the foreground model in terms of number of Gaussian distributions is another important factor that will condition the quality of the final segmentation of the object. The number of Gaussians is determined by the number of color-spatial regions of the object, and the updating process. Therefore, we can improve the precision of the foreground segmentation correctly adapting the spatial model to the silhouette of the object, which means the use of high number of foreground Gaussian distributions. Since the computational cost is related with this number of Gaussians, there is a trade-off between the resolution of the foreground model and the frame-rate of the system. We can control the number of foreground Gaussians by changing the Gaussian split threshold presented in Section 6.3.2.2.

We propose to give the same resolution in number of Gaussians to the color and depth foreground models because this spatial configuration achieves a correct spatial modeling of the object to avoid false positive detections. Moreover, the fact that both color and depth models share the same spatial distribution, allow us to speed up the spatial model initialization and updating processes.

Shadow effects need also a special comment. The region based foreground model that we use to model the color information of the object presents high tolerance to the shadow effects and avoids many false negative errors. Moreover, since the depth sensor is not affected by the shadows, the system can avoid its effects under normal

Figure 6.8: Qualitative results of sequence 2. From left to right: original sequence recorded by the color sensor, pixel-based fg detection in the depth domain [WADP02], pixel-based fg detection in the color domain, region-based foreground segmentation using the method proposed in Chapter 4, color and depth segmentation obtained using [SK11] and the results obtained by our proposal by combining color and depth sensors.

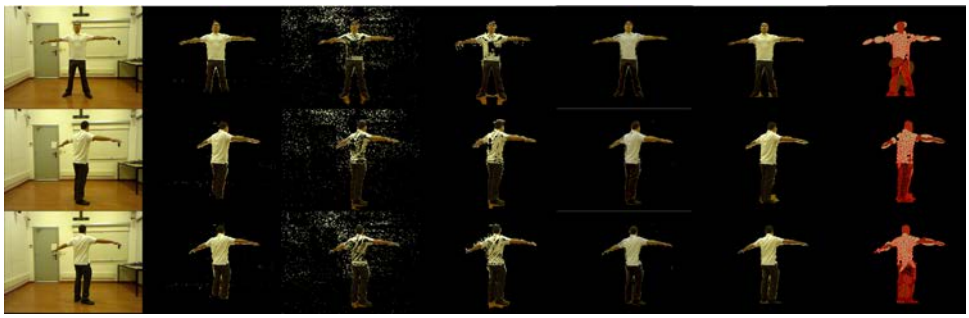


Figure 6.9: Qualitative results of sequence 3. From left to right: original sequence recorded by the color sensor, pixel-based fg detection in the depth domain [WADP02], pixel-based fg detection in the color domain, region-based foreground segmentation using the method proposed in Chapter 4, color and depth segmentation obtained using [SK11] and the results obtained by our proposal by combining color and depth sensors, and the foreground SCGMM model.

circumstances. Hence, the main errors produced by the shadows can be basically an increase of the false positive detections when the foreground depth model present low reliability, this is: in depth camouflage situations and in the boundaries of the object, where we are refining the foreground detection with the trimap analysis just with color information.

Finally, in this chapter we present a proof of concepts about how to combine color and depth sensors information. Hence, this method can be adapted to other kind of segmentation systems better designed to other setups and special circumstances with minimal modifications. For instance, the background model that we use is appropriate for indoor scenarios with stable illumination and background configuration but other kind of background models like the adaptive pixel-wise GMM proposed by [SG00] can be used for color and depth domains in outdoor or noisy scenarios.

Figure 6.10: Effects of the trimap refinement step. From left to right: original sequence recorded by the color sensor, the results obtained by our proposal without applying the trimap refinement step, and the results obtained by our proposal after using it.

Table 6.1: Data Base f_{measure} Comparison Results. In bold type the results corresponding to the best scores.

Sequence	Foreground Segmentation technique				
	RGA-RGB	**RGA-DEPTH**	**SCGMM**	**Schiller**	**LogPool**
office1	0.82	0.83	0.92	0.92	**0.93**
office2	0.62	0.91	0.84	0.90	**0.94**
sroom1	0.87	0.83	0.87	0.86	**0.89**
sroom2	0.61	0.87	0.93	0.84	**0.94**
sroom3	0.65	0.89	0.95	0.88	**0.96**
sroom4	0.88	0.88	**0.94**	0.85	0.93
sroom5	0.78	0.87	0.83	0.84	**0.92**
sroom6	0.62	0.66	0.77	0.86	**0.91**
sroom7	0.72	0.89	0.94	0.88	**0.95**

Chapter 7

Reliability Maps Applied to Robust Shape From Silhouette Volumetric Reconstruction

7.1 Introduction

3-dimensional reconstruction from multiple calibrated planar images is a major challenge in the image processing area in order to obtain a realistic volumetric representation of the objects and people under study. In this field, Shape from Silhouette (SfS) gather all the techniques to reconstruct the 3-dimensional structure from a set of segmentation masks obtained from multi-view smart-room scenarios. Many of the SfS proposals are based on the Visual Hull concept presented by [Lau91] and based on the 3-dimensional geometric modeling, first introduced by [Bau74].

As explained in Section 3.2.3.1, the Visual Hull (VH) is defined as the largest solid volume equivalent to the real object that explains the silhouettes of each one of the views, obtained as the geometric intersection of all visual cones explaining the projection of a silhouette in each corresponding view. Therefore, the quality of the 3-dimensional reconstruction will depend on the configuration of the acquisition setup used to record the scene: number of camera sensors, their position in the smart room, the kind of sensors utilized in the recording and their calibration.

Moreover, since the Visual Hull is based on the intersection of the rays that 2D foreground points in each view define in 3D space, these methods are also highly

dependent on the quality and consistency of the silhouettes obtained in each one of the views since a miss in a view propagates this error into the 3D volume reconstruction. Even if we assume a correct configuration and calibration of the set of cameras that performs the acquisition setup, these errors in silhouettes consistency can arise due to the foreground-background configuration of the scene. Most common errors appear due to the presence of shadows and camouflage situations between foreground and background regions. Therefore, there is a clear dependency of the 3D reconstruction with respect to the foreground segmentation, which makes foreground segmentation central to the problem of obtaining an automatic volumetric reconstruction.

In this chapter we focus on multi-view smart-room sequences recorded by means of an acquisition setup composed of M static color cameras used for a posterior 3-dimensional reconstruction. We will use the improvements presented in previous chapters, regarding 2D foreground segmentation and sensor reliability analysis, in 3D SfS systems. Our objective is to establish a more complete communication between the foreground segmentation process and the 3-dimensional reconstruction in order to obtain an enhanced 3D object volume.

7.1.1 State of the Art

Since Shape from Silhouette techniques are based on 2-dimensional foreground masks, previous work in this area can be presented as the foreground segmentation techniques suitable for 2-dimensional smart-room scenarios, SfS proposals focused on the 3-dimensional reconstruction algorithms, and those techniques that try to enhance the final 3-dimensional volume by improving the communication between both steps. In addition to the techniques presented in 3.2.3.1, this section will extend the previous work knowledge, by introducing other important methods of the literature.

7.1.1.1 Foreground Segmentation

A common approach for segmenting the foreground objects in multi-view smart-room sequences consists in defining individual strategies for each one of the views, which can lead to waste memory resources and robustness in the overall segmentation process, since these techniques are not taking into account the redundant information that appear among views. In this kind of sequences, this redundancy is strongly present in the data available to define the foreground objects to segment, and thus, it can be utilized to improve the foreground segmentation in each one of the views.

7.1.1.2 Shape from Silhouette

Recently, some SfS proposals have been presented in order to improve the resultant 3D volumetric reconstruction. [FB03, LFP07, MBM01] worked with Polyhedral Visual Hull techniques, which computes the 3D surface of the visual hull and describes it as a polygon mesh, while, more recently, [FLB07] proposed a polygonized Visual Hull.

7.1.1.3 Shape from Silhouette with Enhanced Robustness

Many authors have been working in 3-dimensional reconstruction techniques that deal with the inconsistency of the silhouettes proposing SfS techniques with enhance robustness. In these proposals, consistency tests between views and further processing is applied in order to overcome the limitations in the silhouette extraction. [AP09] uses techniques based on minimization of energy functions including functionals based on the local neighborhood structures of three-dimensional elements and smoothing factors. Algorithms based on graph cuts allow to obtain a global minimum of the defined energy function ([KZ02]) with great computational efficiency. The method explained in Section 3.2.4.2 ([FB05]) proposed the Space occupancy grids where each pixel is considered as an occupancy sensor, and the visual hull computation is formulated as a problem of fusion of sensors with Bayesian networks, while the system introduced in 3.2.4.1 ([LP06a]), worked with the Shape from Inconsistent Silhouette for cases where silhouettes contain systematic errors, by combining the probabilities of each one of the pixel. [DMMM10] proposed a Shape from silhouette using Dempster-Shafer theory which takes into account the positional relationships between camera pairs and voxels to determine the degree in which a voxel belongs to a foreground object.

Although these techniques increase the computational cost, the results obtained overcome the simple systems that consider the foreground segmentation and the 3-dimensional reconstruction as separated steps. In spite of this, many of these methods uses probabilistic modeling of the background and the foreground classes with simple models, which can lead to decision errors in the final volume reconstruction.

7.1.2 Proposed Method

We propose a Shape from Silhouette system that matches the SfS with Enhanced Robustness proposals, and follows the work-flow displayed in Figure 7.1:

Foreground segmentation: We propose to use the Bayesian region-based foreground segmentation method for each one of the views explained in Chapter 4,

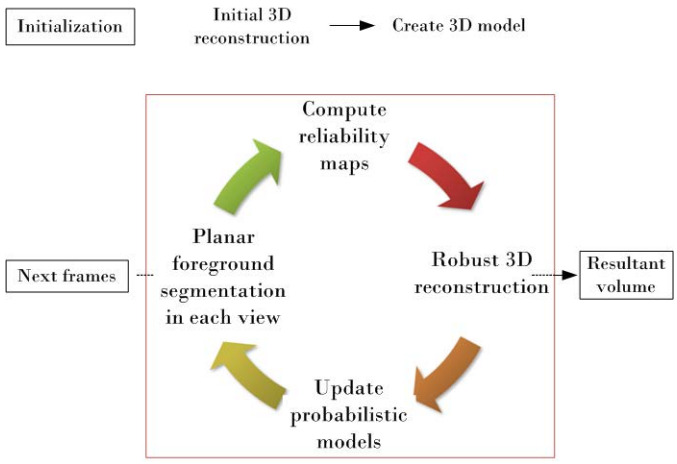


Figure 7.1: Work-flow of the proposed shape from silhouette system.

which combines pixel-wise background model with region-based spatial color Gaussian Mixture Model for the foreground and region-based spatial color Gaussian Model for the shadow regions. Hence, we achieve a correct modeling of the foreground object that will improve the final foreground segmentation masks in each view.

Reliability maps: The advantages of the foreground segmentation method is two-fold, it achieves a correct foreground detection of the objects and, moreover, it allows us to compute the reliability map of each view by comparing the probabilistic models of the foreground, background and shadow classes one another. According to the development presented in Section 6.4.1, and for each pixel, we compute the Hellinger distance [Ber77] between the foreground model and the background and shadow models. This distance will give us a [0,1] bounded value which will be used as a reliability value.

3-dimensional volumetric reconstruction: We compute the Visual Hull reconstruction based on the intersection of the rays that 2D foreground points in each view define in 3D space, but using only the pixels of each view that present enough reliability to be taken into account in the process. That is, working only with those pixels where foreground model is separated in the color domain from the background and shadow models, thus, dealing with inconsistent silhouettes obtained in foreground-background camouflage situations.

Update probabilistic models: Since we use the 2D planar segmentation proposed in Chapter 4, the updating of the probabilistic models used for the foreground, background and shadow classes will be performed according to this method by using the segmentation obtained in previous frame $t-1$.

This system allows a reconstruction which automatically defines the optimal tolerance to errors for each one of the voxels of the volume, in order to obtain a robust 3D volume of the object, improving the traditional Shape from Silhouette reconstruction obtained by defining a fixed tolerance for the overall volume (tolerance to errors reconstruction was introduced in Section 3.2.3.1.1).

The chapter is organized as follows: Section 7.2 explains the Bayesian foreground segmentation method utilized in each view. Section 7.3 is devoted to explain the reliability maps while Section 7.4 describes the robust 3-dimensional reconstruction. Finally, Section 7.5 and Section 7.6 focus on the results and conclusion respectively.

7.2 Multi-View Foreground Segmentation

Specific probabilistic models are used to represent the foreground and background classes for each one of the color sensors. Analogously to Section 4, we use one pixel-wise Gaussian model in the color domain for the background, and two region based models for the foreground and shadow classes: Spatial-Color Gaussian Mixture Model (SCGMM) and Spatial-Color Gaussian Model (SCGM) respectively. All the processes concerning the 2D planar foreground segmentation: initialization, classification and updating, are based on the development carried out in Chapter 4. We refer the reader to this chapter in order to extend this information.

7.3 Reliability Maps

Analogously to the development proposed in Chapter 6, we obtain the reliability maps of each camera view γ^{C_j}, by analyzing the similarity between the foreground and the background classes, but in this approach, we also compute the similarity between the foreground and the shadow classes in order to take into account the shadow effects as well. Hence, for each one of the image pixels $z_i = RGB\ XY \in I_t$, in each camera view C_j, we compute the Hellinger distance ([Ber77]) in the color domain, to detect the degree of similarity between foreground and $l \in \{\text{background, shadow}\}$ models that each one of the camera sensors $J \equiv \{C_1, \dots C_j, \dots C_M\}$ presents in the color $c = RGB$ domain:

$$H_i^{C_j}(q_{\text{fg},i}^{C_j}, q_{l,i}^{C_j}) = \sqrt{1 - \text{BC}_i^{C_j}}, \tag{7.1}$$

where $0 \leqslant H(q_{\text{fg},i}^{C_j}, q_{l,i}^{C_j}) \leqslant 1$, $q_{\text{fg},i}^{C_j}$ and $q_{l,i}^{C_j}$ are the p.d.f.'s that model the i-th pixel for the foreground and l class respectively in the C_j view. BC is the Bhattacharyya

Coefficient, which is formulated, for a multivariate Gaussian distribution, as shown in Chapter 6, Equation (6.14).

Since the 2D foreground classes are modeled by means of SCGMMs, $q_{\mathrm{fg},i}^{C_j}$ will be chosen according to the Gaussian k that maximizes the probability of the i-th pixel under analysis for each view:

$$P(k|z_i, \mathrm{fg}) = \frac{\omega_k G_{\mathrm{fg}}(z_i, \mu_k, \sigma_k)}{\sum_k \omega_k G_{\mathrm{fg}}(z_i, \mu_k, \sigma_k)} \tag{7.2}$$

In the case of the background, since we have defined a pixel-wise model, $q_{\mathrm{bg},i}^{C_j}$ will be directly obtained from the background Gaussians associated to this pixel. For the shadow class, $q_{\mathrm{sh},i}^{C_j}$ is the SCGM used to model the shadow projected by the person. The foreground-shadow reliability will be utilized only over the spatial region modeled by the shadow Gaussian, since it is the only region affected by the shadow effects.

Therefore, for each one of the pixels of the camera sensors, we will obtain the final reliability value $\gamma_i^{C_j}$, according to the comparison between foreground-background and foreground-shadow models. The final reliability maps for each camera, γ^{C_j}, is obtained according to:

$$\gamma_i^{C_j} = \begin{cases} \min\left(H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{bg},i}^{C_j}),\ H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{sh},i}^{C_j})\right) & \rightarrow \quad \text{shadow model region} \\ H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{bg},i}^{C_j}) & \rightarrow \quad \text{otherwise} \end{cases} \tag{7.3}$$

where the most restrictive distance between $H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{bg},i}^{C_j})$, $H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{sh},i}^{C_j})$ is chosen in the regions belonging to the spatial shadow models, and $H_i^{C_j}(q_{\mathrm{fg},i}^{C_j}, q_{\mathrm{bg},i}^{C_j})$ in the rest of the image.

7.4 Robust 3-Dimensional Reconstruction

The concept of Visual Hull (VH) is strongly linked to the one of silhouettes' consistency. Total consistency hardly ever happens, due to errors in the 2D segmentation process, and tolerance to errors (τ) can be used in the 3D reconstruction process. This approach will lead to reduce the number of false negative errors although losing precision in the final reconstructed volume.

We propose a *SfS* reconstruction method based on the silhouette reliability principle. Our system validates the reliability of the background regions of the silhouettes, since they are the ones which propagate misses to the 3D volume re-

construction, and uses these reliable background pixels of each view to compute the robust Visual Hull of the object, thus dealing with 2D errors. Since we are using a foreground, background and shadow modeling, we use, for each pixel i in each view C_j, the reliability $\gamma_i^{C_j}$ explained in the previous section, according to the similarity that the foreground model presents with respect to the background and shadow probabilistic models.

The robust shape from silhouette algorithm that we propose is shown in Algorithm 5.

Algorithm 5 Reliable Shape from Silhouette algorithm

Require: : Silhouettes: S(c), Reliability Test: RT(voxel, camera),
Projection Test: PT (voxel, silhouette)

```
for all voxel do
    voxel ← Foreground
    for all cameras do
        if PT (voxel, S(c)) is false and RT(voxel, camera) > R_th then
            voxel ← Background
        end if
    end for
end for
```

The projection test (PT) consists in testing the central pixel within the splat of the voxel in camera C_j, which is, in fact, the pixel placed in the centroid of the number of pixels under the projection of the voxel in the j-th view.

Once the projection Test has been carried out, we can use the voxel-pixels correspondence to check the reliability that each one of the pixels present. The Reliability Test (RT) is based on the analysis of the reliability value for each one of the pixels that appear in the voxel's projection for each one of the views, ($\gamma_i^{C_j}$), which is [0,1] bounded.

We define the Reliability threshold R_{th} as a value $0 < R_{\text{th}} < 1$ which will determine the minimum reliability value to consider the background pixels in the final reconstruction process. In our experiments, we have tested that a reliability factor $R_{th} = 0.7$ yields correct results in the final reconstruction process.

This 3-dimensional reconstruction is equivalent to define an optimal error tolerance value τ for each one of the voxels of the image, improving the precision of the volume in those regions where no tolerance is necessary, while reducing the false negative errors in regions with reliable misses.

7.5 Results

We have evaluated our proposal by analyzing four multi-view sequences, of the database presented in [INR], which present strong difficulties to achieve a correct 3-dimensional reconstruction due to the similarity between some foreground regions and the background. These sequences have been recorded with different acquisition setups in order to better analyze the effect of the errors tolerance in the volumetric reconstruction:

- Baton sequence, recorded with 16 cameras. 180 frames.
- Dancer sequence, recorded with 8 cameras. 250 frames.
- Karate sequence, recorded with 16 cameras. 150 frames.
- Open arms sequence, recorded with 18 cameras. 300 frames.

The sequences are used to carry out a qualitative and quantitative evaluation of the proposal presented in this chapter. Qualitative results are obtained by computing the overall sequences and selecting some representative frames, while the quantitative results are obtained by comparing the results, in the first camera view of each sequence, with the ground truth for ten equally-distributed frames of each one.

The evaluation is obtained by processing each sequence using our proposal (Robust3D) and we have compared the volumetric reconstruction results with the ones obtained by using the Visual Hull reconstruction with different tolerance to errors (τ) in order to achieve a conservative volume of the objects (Tol=0, Tol=1 and Tol=2).

Figure 7.2 and Figure 7.3 display the results obtained in these four sequences recorded with 16 cameras (first and third sequences), 8 cameras (second sequence) and 18 cameras (fourth sequence). Some representative views of the overall multi-view sequence have been selected in each case.

Figure 7.2, shows the volumetric reconstruction results in each one of the sequences. The segmentation masks used in all these reconstructions are the ones obtained thanks to the segmentation system proposed in Chapter 4, and are displayed on the first row, in second column. As we can observe, some false negative errors are present in some of the views, due to the foreground-background camouflage problem and the presence of shadows, which reduces the recall of the results increasing the false negative detections.

In the first row-third column of Figure 7.2, we can see the spatial representation of the foreground model. Each ellipse represents one Gaussian of the foreground

model, and is colored with the mean color that each distribution is modeling.

From second to fifth row, we can observe the different volumetric reconstructions that we can obtain using the Visual Hull reconstruction with different tolerance to errors. When we do not use any tolerance to errors ($\tau = 0$) (second row), any false negative error that appears in the 2D segmentation is propagated to the final 3D volume, thus generating critical false negative errors in the resultant reconstruction. When using tolerance to errors (third and fourth rows), we reduce significantly the propagation of the false negative errors to the 3D space, although losing precision in the volumetric reconstruction, thus obtaining a coarse representation of the object. Our system (fifth row) achieves a 3-dimensional reconstruction that only applies the tolerance to errors in those background pixels where the reliability between foreground and background and shadow classes is low, thus reducing the propagation of those errors to the 3D space. As we can see, our system achieves an object reconstruction that presents similar precision than the Visual Hull reconstruction without tolerance ($\tau = 0$), but solving a high percentage of false negative errors.

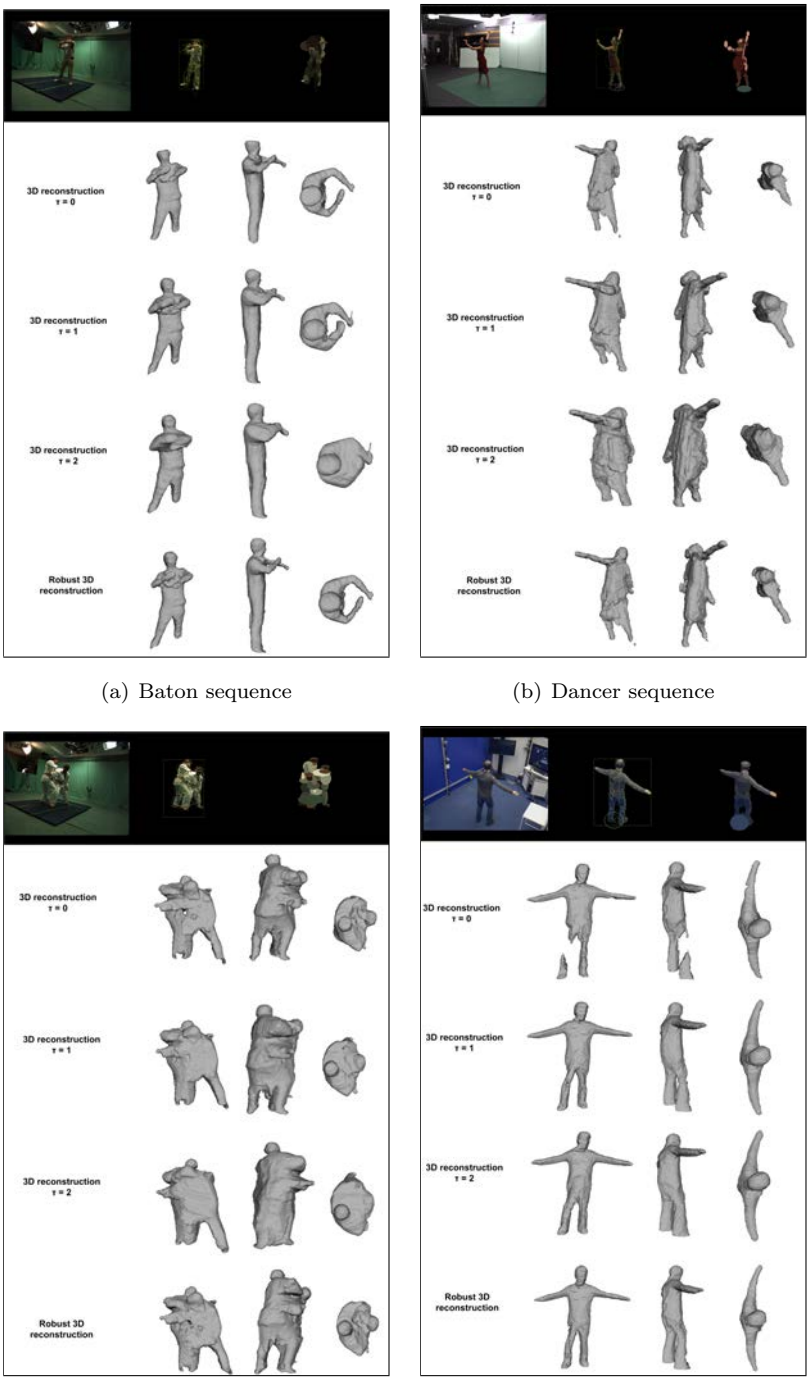
Figure 7.3 shows more qualitative results obtained by projecting the resultant volumes to the view under analysis. Second column of this figure shows the volumetric reconstruction obtained by our proposal where, voxels computed with a volumetric reconstruction without tolerance to errors (tol=0) are displayed in white color, voxels that present tol=1 are colored in red and finally, voxels obtained by means of a tol=2 reconstruction are depicted in green color. As we can observe, the volumetric reconstruction obtained by means of the method presented in this chapter, achieves a better reconstruction of the volume thanks to the different tolerance to errors that each one of the voxels present according to the reliability of the the pixels in each view.

Finally, quantitative results of these sequences are displayed in Figure 7.4, Figure 7.5, Figure 7.6 and Figure 7.7. As we can see, the method that we propose, achieves a volumetric reconstruction that adapts better to the circumstances of the sequence under analysis than the reconstructions with fixed tolerance. Our method maintains a high f_{measure} value for the sequences under study, maintaining the precision of the volumetric reconstruction while reducing the false negative detections.

Since this system utilizes the foreground segmentation method proposed in Chapter 4, the computational cost of our system depends on the number of Gaussians of the model and the sizes of the images. Considering a parallel processing for computing the foreground segmentation and reliability maps in each camera sensor, the system achieves a speed of 0.3 frames/second analyzing a standard sequence and using an Intel Core2 Duo 3GHz processor and 20 GB RAM.

7.6 Conclusions

In this chapter, we have introduced a novel multi-view segmentation and 3D reconstruction system. To this end, we have proposed a robust Visual Hull reconstruction that uses the reliability of the pixels to avoid those views where the pixels detected as background, present high similarity between foreground, background and shadows models. Although the system is highly dependent on the foreground segmentation model and how it represents the foreground object in each one of the views, our approach achieves better accuracy of the reconstructed volume while reducing the critical misses that appear in a direct 3D reconstruction with $\tau = 0$, and reducing the false positive regions that appear if we decide to use a direct $\tau = 1$ or $\tau = 2$ reconstruction.



(a) Baton sequence (b) Dancer sequence

(c) Karate sequence (d) Open arms sequence

Figure 7.2: Qualitative foreground segmentation and 3D volume reconstruction results. First row shows from left to right: original view; Bayesian foreground segmentation proposed in the paper: Color ellipses correspond to the Gaussians of the projected foreground model, white ellipse corresponds to the spatial representation of the shadow model; Foreground model projected to the view. The ellipses correspond to the foreground model projected to this view and they are colored with the mean color that are modeling. Next rows are: the projected volume computed with tolerance $\tau = 0$; volume with $\tau = 1$; volume with $\tau = 2$; Robust 3D reconstruction using our method.

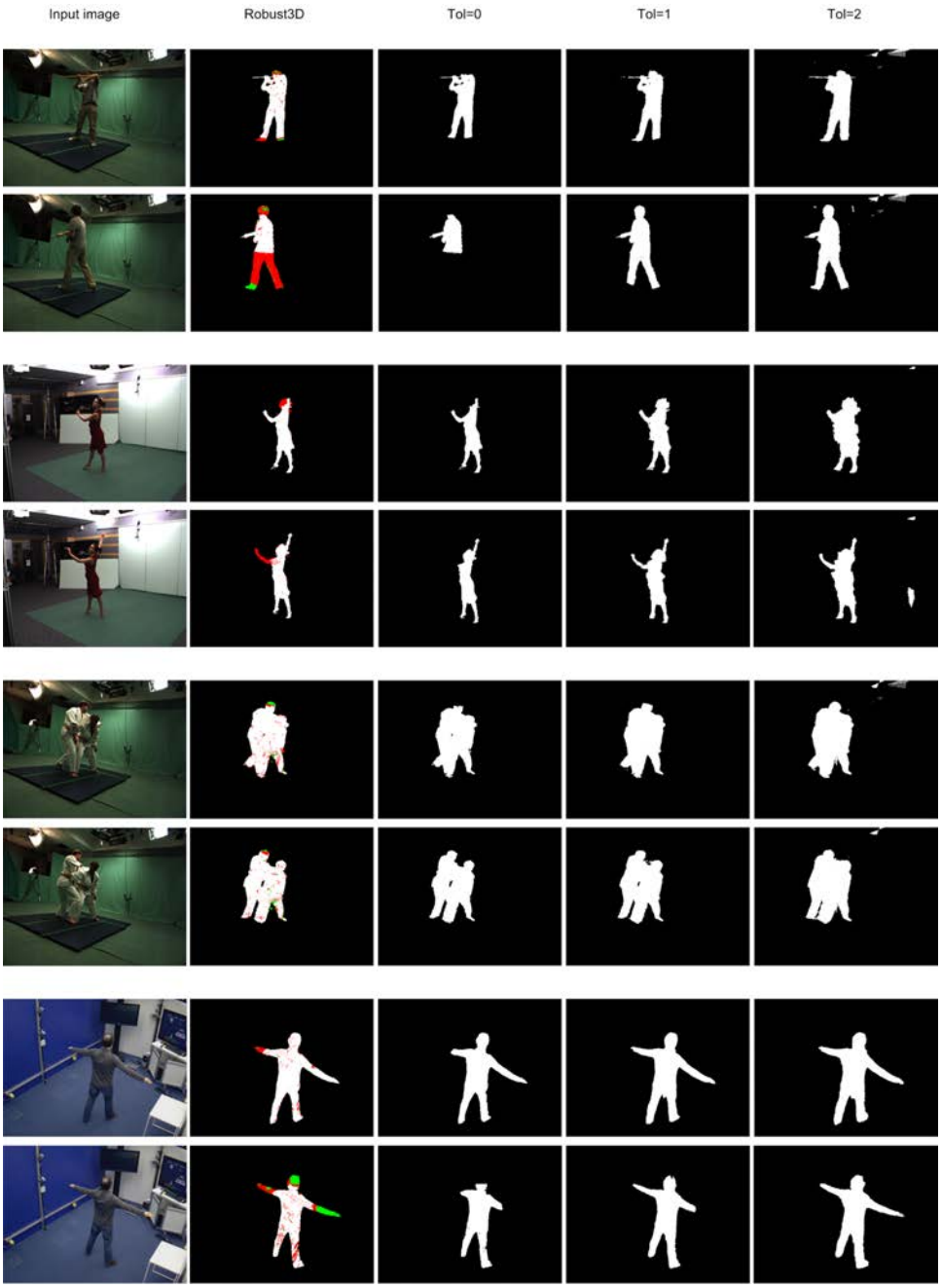
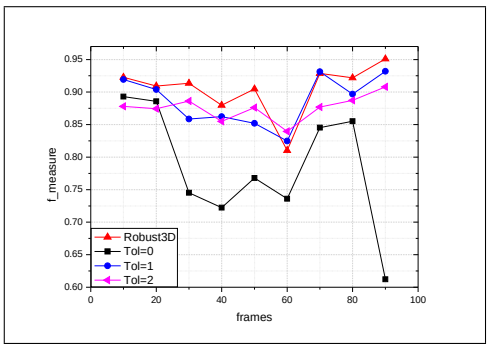
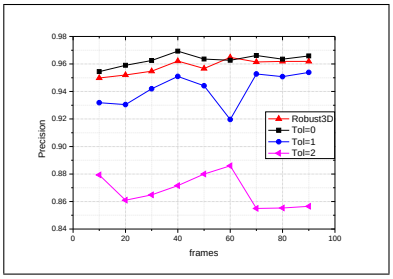


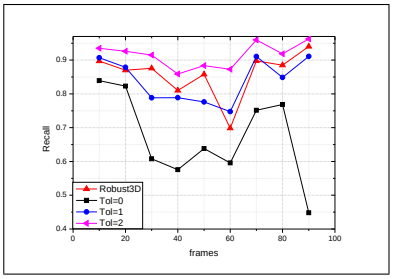
Figure 7.3: Qualitative 3D reconstruction results. Projection of the 3D reconstructed volume over the 2D view under analysis. From left to right: original view; robust 3D reconstruction using our method; the projected volume computed with tolerance $\tau = 0$; volume with $\tau = 1$; volume with $\tau = 2$. In second column, white voxels are the ones belonging to the tol=0 reconstruction, red voxels come from tol=1, and green voxels are their counterpart for tol=2.



(a)

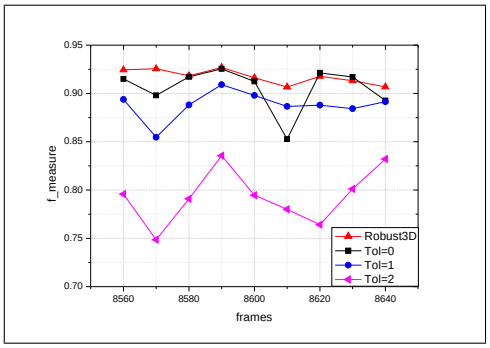


(b)

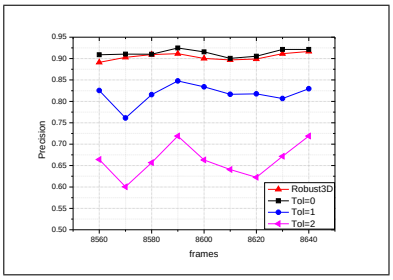


(c)

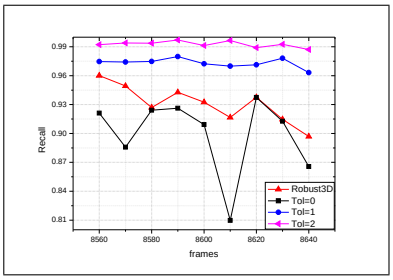
Figure 7.4: Quantitative evaluation of baton sequence (corresponding to Figure 7.2(a)).



(a)

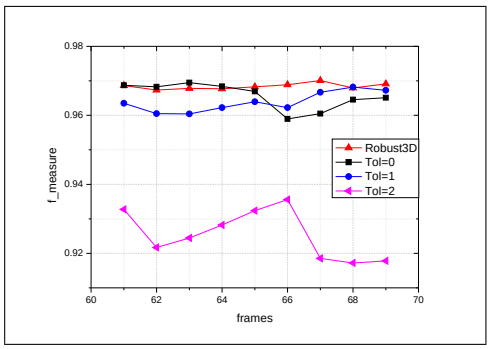


(b)

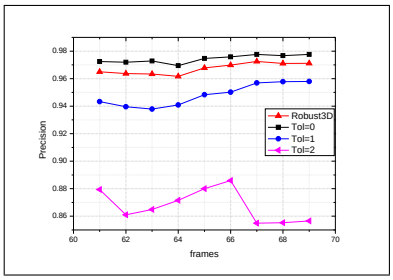


(c)

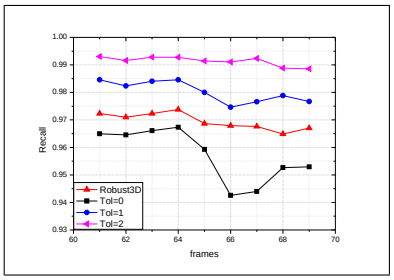
Figure 7.5: Quantitative evaluation of dancer sequence (corresponding to Figure 7.2(b)).



(a)

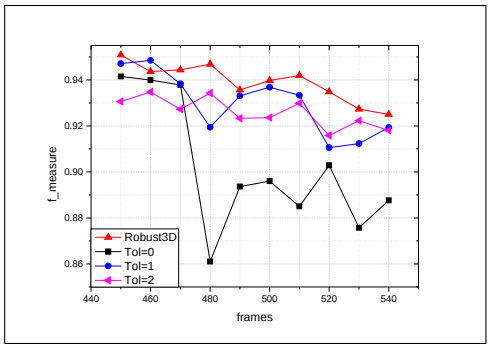


(b)

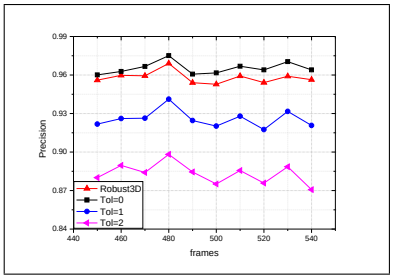


(c)

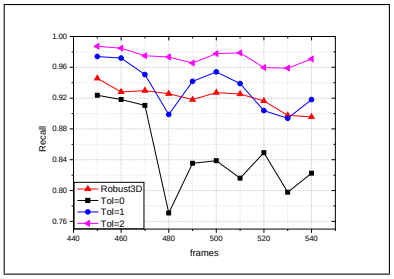
Figure 7.6: Quantitative evaluation of karate sequence (corresponding to Figure 7.2(c)).



(a)



(b)



(c)

Figure 7.7: Quantitative evaluation of open arms sequence (corresponding to Figure 7.2(d)).

Chapter 8

Joint Multi-view Foreground Segmentation and 3D Reconstruction with Tolerance Loop

8.1 Introduction

In this chapter we present a foreground segmentation and 3D reconstruction system for multi-view scenarios based on a different principle than the method presented in the previous chapter. This proposal was developed jointly with Dr. Jordi Salvador Marcos, expert in 3D objects reconstruction, in order to achieve a cooperative framework between the foreground segmentation and 3D reconstruction processes. In this system, we introduce the spatial redundancy of the multi-view data into the foreground segmentation process by combining segmentation and 3D reconstruction in a two steps work-flow. First, the segmentation of the objects in each view uses a monocular, region-based foreground segmentation in a MAP-MRF framework for foreground, background and shadow classes. Next, we compute an iterative volume reconstruction in a 3D tolerance loop, obtaining an iteratively enhanced SfS volume. Foreground segmentation is improved by updating the foreground model of each view at each iteration. The results presented in this chapter show the improved foreground segmentation and the reduction of errors in the reconstruction of the volume.

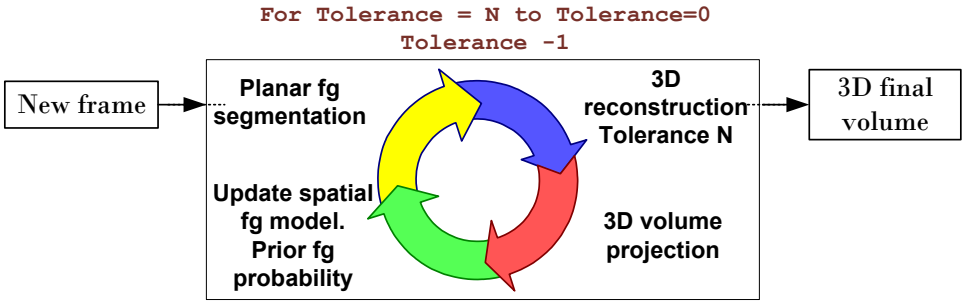


Figure 8.1: Work-flow of the proposed system.

8.1.1 Proposed System

We propose a system for foreground segmentation and 3D reconstruction that combines foreground segmentation with volumetric reconstruction, better exploiting the spatial data redundancy in multi-view scenarios. The system is based on the principle that improved planar foreground segmentation of each view also improves the 3D reconstruction. Hence, we propose an iterative loop involving both processing steps for each frame of the sequence where foreground segmentation in each view is performed by means of SCGMM foreground modeling presented in Chapter 4 and Visual Hull reconstructions help, in turn, to improve the segmentation, as shown in the overview of the system work-flow in Figure 8.1.

As commented in previous chapters, the volumetric reconstruction is very sensitive to the presence of foreground detection errors in any view. A miss in a view propagates this error into the 3D volume reconstruction. In this chapter, we present an enhanced Conservative Visual Hull reconstruction with error tolerance to achieve volumetric reconstruction avoiding the propagation of silhouette misses. The resulting 3D volume, which will initially show more false positives, will be projected in some iterations for the spatial updating of the foreground model. This projection is also used to increase the prior foreground probability of the pixels belonging to the projected volume, with the aim of recovering foreground object regions that were not correctly modeled by the foreground model of each view. Afterwards, a new segmentation will be obtained for each view. This refined planar foreground segmentation is then used again for reconstructing the Visual Hull iteratively decreasing the tolerance to errors until we reach a zero tolerance Visual Hull reconstruction. The final 3D volume improves the performance of a Visual Hull reconstruction obtained with tolerance 2, reducing false positives, and the one obtained directly with tolerance 0, improving the completion of the volume by reducing false negatives.

This chapter is organized as follows: foreground segmentation system is explained in Section 8.2. Section 8.3 is devoted to the proposed 3D reconstruction technique. Section 8.4 defines the 3D reconstruction feedback. Finally, results and

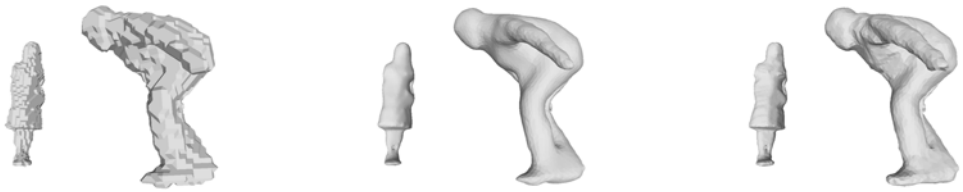


Figure 8.2: Proposed 3D reconstruction. From left to right, conservative visual hull; smoothed surface after 10 iterations of low-pass mesh filtering; and resulting surface after fitting and one step of low-pass mesh filtering.

conclusions are presented in Section 8.5 and Section 8.6 respectively.

8.2 Planar Foreground Segmentation

As in the previous chapter, we use the foreground segmentation based on the work presented in Chapter 4 , that combines background, foreground and shadow models into a MAP-MRF framework. We use a probabilistic pixel-wise background model in the *RGB* color domain to obtain initial foreground and shadow pixels via exception to background analysis that are used to initialize the region-based foreground and shadow models. For each frame, a Bayesian pixel classification is done among the background, the foreground, and the shadow models. Finally, this classification is used to update the foreground, shadow and background models.

8.3 3D Reconstruction Technique

We use a technique to extract an accurate surface which is robust to inconsistent silhouettes presented in [SM11]. The method consists in the concatenation of: (1) a *conservative* estimate of the *visual hull*, with tolerance to silhouette segmentation errors and spatial smoothness constraints; (2) an iterative low-pass surface filtering for extracting a smooth surface with consistent vertex normals and limited curvature from a mesh with correct topology extracted with *marching cubes*; and (3) a surface fitting that provides a more accurate surface estimate. Surfaces obtained after each of these three stages are shown in Figure 8.2, using 16 views from [INR].

8.3.1 Conservative Visual Hull

Voxel occupancy is defined by the minimization of an energy function, represented as a bidirectional graph, with a data term determined by a conservative consistency test and a constant regularization term for spatial smoothness.

The bidirectional graph is built as follows. The data term –graph node– is set to 1 when a voxel's center projects onto pixels classified as foreground in a number of views at least equal to the number of *frusta* in which it is included minus an error tolerance (τ). It is set to 0 otherwise. The estimate of the visual hull is *conservative* in the sense of assuming that τ foreground under-segmentation errors can occur. The regularization term –graph link– is set to a constant smoothing constraint $\lambda \in [0.25, 0.5]$ between each pair of 8-neighbor voxels. The max-flow algorithm [BVZ01] obtains the minimum cost graph-cut, which results in the final labeling of each voxel as occupied or empty.

8.3.2 Iterative Low-Pass Mesh Filtering

Marching cubes [LC87] is applied to the resulting volume, resulting in a topologically correct triangle mesh of its surface. Due to limited volumetric sampling, it lacks accuracy with respect to the original silhouettes when re-projected onto the original viewpoints. A per-vertex fitting of the surface can improve its accuracy, but it requires a robust estimate of per-vertex normals.

Therefore, a smoothed version of the input mesh is obtained through the iterative application of a local filter. This filter consists in setting each vertex's new position as the midpoint between its old position and the average of its adjacent vertexes. After a number of 10 iterations, vertex normals can be estimated by averaging the normals of the adjacent faces to each vertex.

8.3.3 Surface Fitting

This stage fits the surface to the input silhouettes, using a modification of the dynamic surface extraction algorithm in [SSC10]. It consists in a per-vertex dilation by a distance r_d –set to the voxel size– followed by an erosion along its inverted normal, in search of its optimal location. The following method is applied to each vertex $\mathbf{x}_i$:

1. Define a virtual segment, which joins $\mathbf{x}_i$'s dilated position $\mathbf{x}_i^d := \mathbf{x}_i + r_d\hat{\mathbf{n}}_i$ and its eroded position $\mathbf{x}_i^e := \mathbf{x}_i - r_d\hat{\mathbf{n}}_i$
2. Shrink the virtual segment by displacing the dilated position $\mathbf{x}_i^d$ towards $\mathbf{x}_i^e$. Along this path, store the closest position to $\mathbf{x}_i$, namely $\mathbf{x}_i^s$, at which the moving extreme crosses the limit of the conservative visual hull
3. If $\mathbf{x}_i^s$ is found, set it as the new position $\mathbf{x}_i := \mathbf{x}_i^s$

The r_d parameter can be set to the same value as the voxel size used for the voxelized estimate of the conservative visual hull, delivering correct results in most cases. The

rest of the parameters are equivalent to those in the shape-from-silhouette stage.

Finally, in order to improve the visual quality, a single-pass smoothing like the one in Section 8.3.2 is applied on the mesh, resulting in an accurate conservative estimate of the visual hull.

8.4 3D Reconstruction Feedback

Once the volume reconstruction is computed with the corresponding tolerance value, the feedback between the 3D reconstruction and the planar foreground model of each view is performed. The 3D volume of the object is projected to each view, obtaining a projection mask that contains robust information about the foreground segmented in the other views. The projection mask will be taken into account for updating the foreground model of the object in each view, and for increasing the prior probability of the foreground class.

8.4.1 Spatial Foreground Model Updating

At each tolerance loop iteration, we propose to update the foreground model of each view with the projection of the 3D volume, but only in the spatial domain and not in the color domain. This is done in order to reduce error propagation due to false positives appearing at the tolerance loop iterations. And we follow the spatial updating proposed in Chapter 4, Section 4.4.1.2, assigning each pixel belonging to the volume projection to the Gaussian k that maximizes:

$$P(k|z_i, \text{fg}) = \frac{\omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k)}{\sum_k \omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k)} \tag{8.1}$$

the denominator is equal for all classes and can be disregarded:

$$P(k|z_i, \text{fg}) \propto \omega_k G_{\text{fg}}(z_i, \mu_k, \sigma_k) \tag{8.2}$$

Once each pixel has been assigned to a Gaussian, the spatial mean and covariance matrix of each Gaussian are updated with the spatial mean and variances of the region it is modeling. The Gaussians not modeling any pixel are removed from the model. In order to achieve a better adaptation of the model into the foreground object shape, we propose a Gaussian split criterion according to the spatial size of the Gaussian.

8.4.2 Prior Foreground Probability

After the foreground model updating, a new foreground segmentation is computed with the new configuration of the spatial foreground model. For this new segmenta-

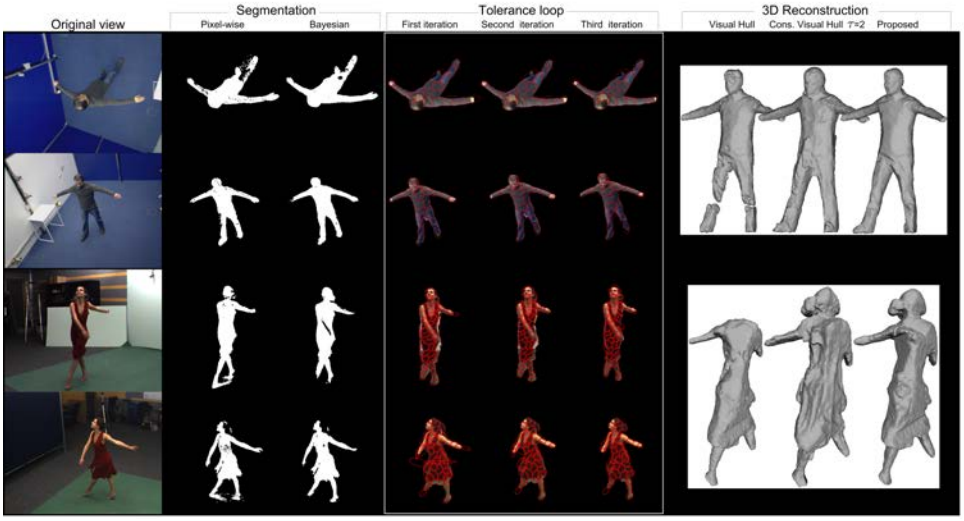


Figure 8.3: Qualitative foreground segmentation and 3D volume reconstruction results of dancer and open arms sequences. From left to right: original view; pixel-wise foreground segmentation as proposed in [WADP02]; foreground segmentation with the presented Bayesian method; tolerance loop iterations showing the projected volume computed with tolerance $\tau = 2$, $\tau = 1$ and $\tau = 0$; 3D reconstruction: initial $\tau = 0$; 3D reconstruction after first loop iteration ($\tau = 2$) and final 3D reconstruction $\tau = 0$ after tolerance loop with 3 iterations.

tion at each loop iteration, we propose to increase the prior foreground probability of the pixels that belong to the projection mask. A constant proved enough in our tests as a factor for scaling the foreground prior probability of the model, thus improving the segmentation in those regions where foreground and background model present similar probability by using the information of the other cameras. Hence, the final foreground probability is defined as follows:

$$P(\mathrm{fg}_i|z_i) \propto P(z_i|\mathrm{fg}_i)P(\mathrm{fg}_i) \tag{8.3}$$

where i is the pixel belonging to the volume projection in the view under analysis, $P(z_i|\mathrm{fg}_i)$ is the foreground probability of the planar fg model and $P(\mathrm{fg}_i)$ is the planar prior probability scaled with the constant factor at each loop iteration.

8.5 Results

We test our proposal with the sequences published in [INR] and with the same data set introduced in Section 7.5. Figure 8.3 displays results obtained in two different scenarios (open arms and dancer) with 18 cameras (top) and 8 cameras (bottom). Two representative views of the overall multi-view sequence have been selected in each case. We have processed each sequence using our system with 3 iterations of

the tolerance loop: $\tau = 2$, $\tau = 1$ and $\tau = 0$. The third column shows the Bayesian foreground segmentation that we obtain before the tolerance loop iteration. Despite some false negatives, this segmentation gives us a foreground mask with less misses than the classical pixel-wise segmentation method [WADP02] in the second column.

The projections of the volume achieved at each tolerance loop iteration are shown at columns four, five and six. We can see how the volume projection adjusts better to the actual shape of the object at each iteration, reducing the false positives due to the tolerance effect while correcting false negatives of the initial foreground segmentation. Also the spatial domain of the foreground model is represented at each iteration (red coloured ellipses) to observe the updating process of the 3D reconstruction feedback. Finally, 3D reconstruction results are shown in the last three columns to illustrate the system improvement. Column seven shows the volume obtained with $\tau = 0$ without using the tolerance loop, column eight shows the volume obtained using $\tau = 2$ -also without tolerance loop- and the last column shows the results obtained by the overall system with three tolerance loop iterations. As we can observe, our method achieves better accuracy of the reconstructed volume reducing the critical misses that appear in a direct 3D reconstruction with $\tau = 0$, and reducing the false positive regions that appear if we decide to use a direct $\tau = 2$ reconstruction.

More qualitative results are displayed in Figure 8.4, where the resultant 3D volume obtained by using our proposal (Tol. loop, in second column), is compared with the volumetric reconstruction obtained by computing the Visual Hull reconstruction with different tolerance to errors (τ) (Tol=0, Tol=1 and Tol=2). In this Figure, we have projected the 3D object reconstruction to the view under analysis for each one of the sequences (baton, dancer, karate and open arms). As we can see, the 3D reconstruction obtained by means of the method presented in this chapter, achieves a correct reconstruction of the volume by computing the tolerance loop for the detection-3D reconstruction process, thus reducing the false negative errors that can appear in the sequence which are not consistent with more than two views.

Finally, quantitative results of these sequences are displayed in Figure 8.5, Figure 8.6, Figure 8.7 and Figure 8.8. In these Figures, the resultant 3D volumes, projected to each view, are analyzed in terms of Precision, Recall and f_{measure}. The system proposed in this Chapter achieves a volumetric reconstruction that reduces the false negative errors that appear in the detection of the object, thus maintaining a correct rate between False negative and False positive errors along the four sequences. Although some false positive detections can appear in the boundaries of the object, our method maintains a high f_{measure} value for the sequences under study, improving the results obtained by the 3D reconstruction methods with tolerance to errors.

Regarding the computational cost of the overall system, the iterative process

carried out in the tolerance loop, makes the time consumption three times higher than the one required to develop a direct foreground detection and 3D volumetric reconstruction.

8.6 Conclusions

In this chapter, we have presented a novel system for multi-view foreground segmentation and 3D reconstruction. By combining both steps in a tolerance loop reconstruction, it improves planar foreground segmentation and, consequently, 3D reconstruction. An iterative 3D reconstruction and foreground segmentation loop allows exploiting the redundancy in the multiple views for correcting the misses of the foreground segmentation of each view, without increasing the false positive errors. The results show how the system outperforms direct 3D reconstruction with $\tau = 0$, reducing the misses of the resulting volume, and with $\tau = 2$, increasing the precision of the volume.

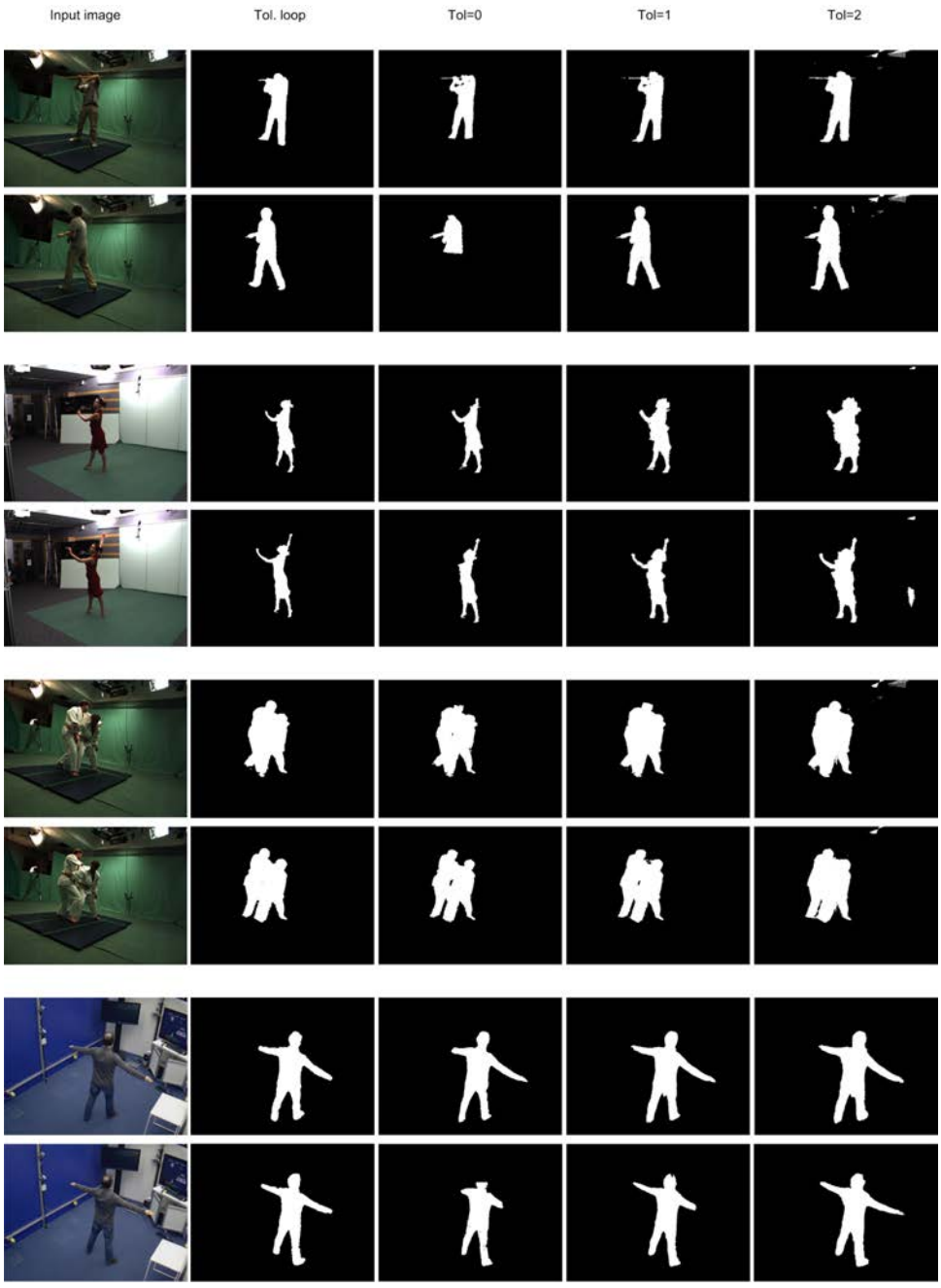
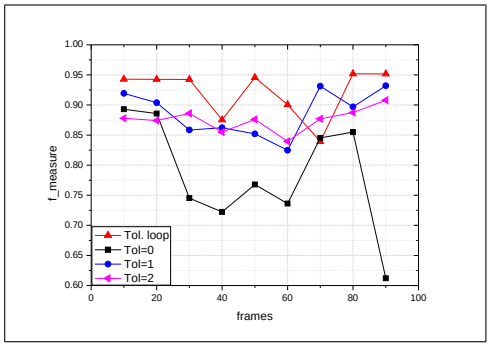
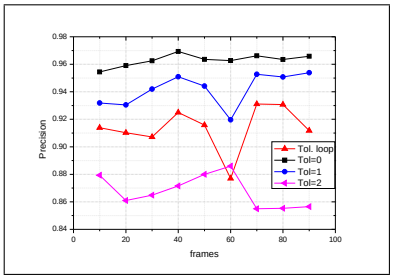


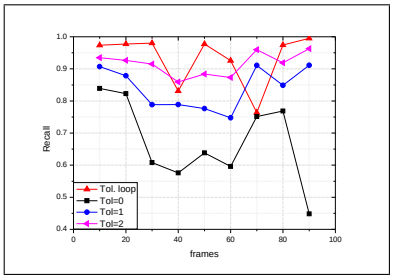
Figure 8.4: Qualitative 3D reconstruction results. Projection of the 3D reconstructed volume over the 2D view under analysis. From left to right: original view; 3D reconstruction using our method; the projected volume computed with tolerance $\tau = 0$; volume with $\tau = 1$; volume with $\tau = 2$.



(a)

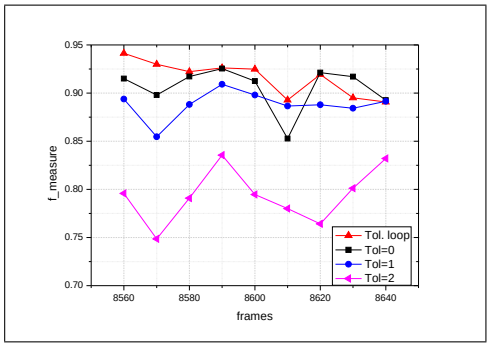


(b)

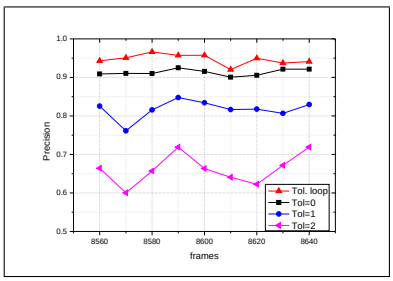


(c)

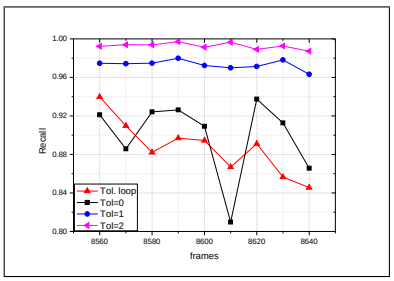
Figure 8.5: Quantitative evaluation of baton sequence.



(a)

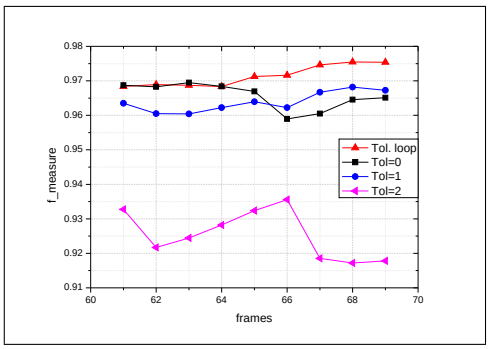


(b)

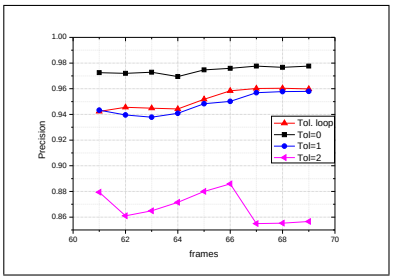


(c)

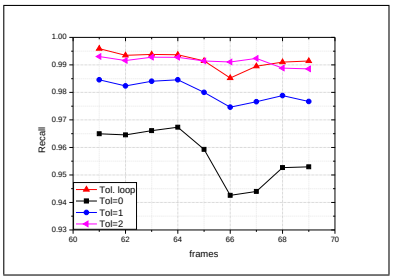
Figure 8.6: Quantitative evaluation of dancer sequence.



(a)

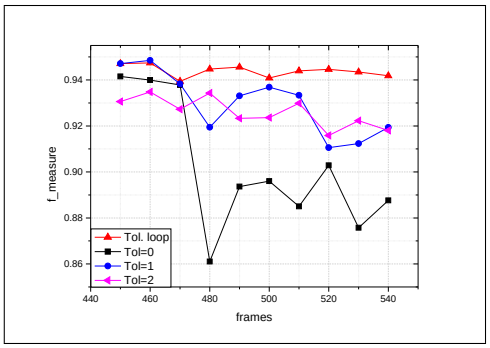


(b)

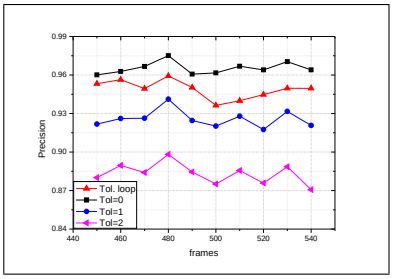


(c)

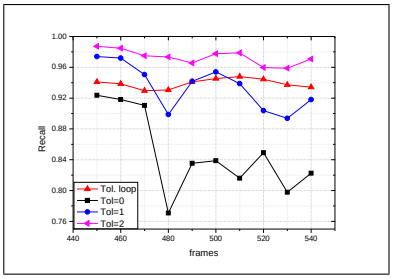
Figure 8.7: Quantitative evaluation of karate sequence.



(a)



(b)



(c)

Figure 8.8: Quantitative evaluation of open arms sequence.

Chapter 9

Multiview Foreground Segmentation Using 3D Probabilistic Model

9.1 Introduction

As we have seen in previous chapters, it is possible to establish a collaboration between views in order to increase the robustness of the overall system (foreground segmentation + 3D reconstruction). The proposals presented so far, have an independent processing for each one of the views, and try to improve the final results by combining the reliability in each view (Chapter 7) or by using the back projection of the resultant 3D reconstructions (Chapter 8). In this chapter, we explain the last proposal of the manuscript that leads us toward the complete integration of the multi-view smart-room segmentation and 3D reconstruction. We propose to define a 3-dimensional modeling of the foreground object under analysis in order to centralize the probabilistic information of the object, for all the views, in the 3-dimensional space, thus giving robustness to the process. This model will be used to achieve the objects' segmentation in each view, preserving the robustness of the model in those views where foreground and background present high similarity and also, it can be exploit to achieve 3D information of the object's movements.

In this system, we define a probabilistic 3D model of the foreground object, where the 3D spatial-color Gaussian Mixture Model (3D SCGMM) is defined to model the probabilistic information of the foreground object to segment in the $v = RGB\ XYZ$ domains. This model will be used as a non-rigid characterization of the object. Therefore, in order to correctly define this model, the 3-dimensional

reconstruction of the object under analysis and the texture that this object presents in the multi-view sequence are necessary. Figure 9.1 shows an example of the 3D reconstruction, and the projection of the colors in each one of the voxels.

(a) (b)

Figure 9.1: Example of colored voxels. a) is the ground truth. b) shows the voxelized 3-dimesional reconstruction with colorized .

Since this chapter deals with 3-dimensional object models as well as the multi-view foreground segmentation, next section is devoted to extend the state of the art presented in previous chapters by reviewing approaches related with 3D models.

9.1.1 State of the Art

In the recent years, there have been special interest in monitoring the human activities and movements in order to obtain a semantic information of the scene. Hence, approaches based on rigid human body models have been proposed in the literature to deal with this analysis. Human motion capture has been extensively studied, [MG01, MHK06, SBB10] give and in-depth survey of the literature. In [GRBS10], the multi-layer framework is proposed by means of particle-based optimization related to estimate the pose from silhouette and color data. The approaches in [BS10, LE10, SBF00] require training data to learn either restrictive motion models or a mapping from image features to the 3D pose. In [SHG+11] the authors propose a rigid human body model that comprises a kinematic skeleton and an attached body approximation modeled as a Sum of Gaussians where 58 joints work together to model a detailed spine and clavicles. In [GFBP10] shape and motion retrieval are detected by means of EM framework to simultaneously update a set of volumetric voxel occupancy probabilities and retrieve a best estimate of the dense 3D motion field from the last consecutive frame set.

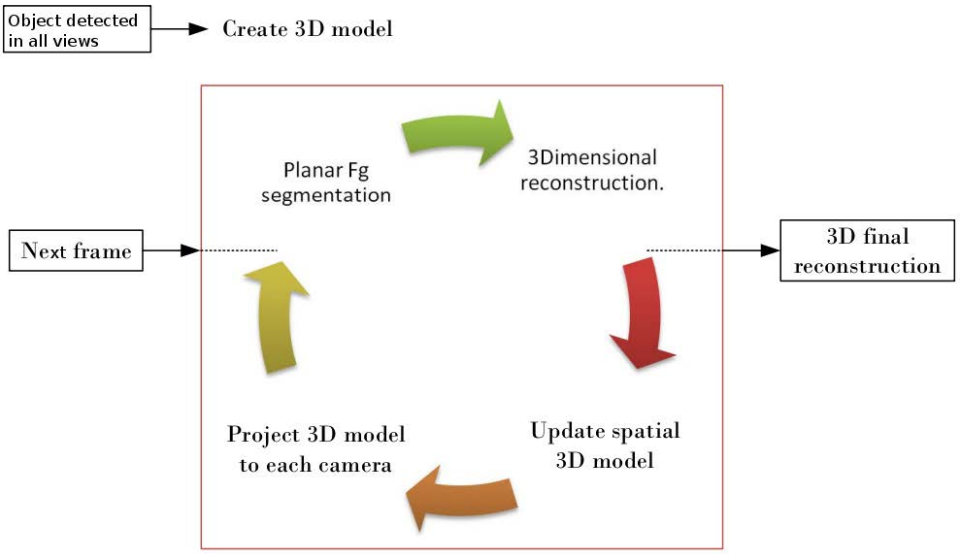


Figure 9.2: Work-flow of the proposed system.

9.1.2 Proposal

Figure 9.2 shows the work-flow of the system. The main steps of this work-flow are:

Create 3D model: Once all the cameras of the multi-view system have detected and segmented the object under analysis, the foreground 3D SCGMM can be created with the 3D reconstruction obtained from the 2D silhouettes. Although any SfS technique can be used to perform the volumetric reconstruction, we utilize a conservative Visual Hull reconstruction with tolerance $\tau = 1$ in order to reduce the possible misses without increasing too much the false positive detections. Moreover, the voxels of this volume are colorized with the object colors in order to obtain a realistic volume reconstruction, by obtaining the average color that the pixels belonging to the voxel's projection present in each view. The voxels spatial and color information will be used to initialize the foreground 3D SCGMM by means of the EM algorithm [DLR+77]. Next frames of the sequence will utilize the 3D model in the segmentation process.

Foreground segmentation: Foreground segmentation is computed by means of the system proposed in Chapter 4, thus combining in a Bayesian MRF-MAP framework pixel-wise background model with SCGMM and SCGM foreground and shadow models respectively.

3-dimensional volumetric reconstruction: As in the 3D model creation, conservative Visual Hull reconstruction with tolerance $\tau = 1$ is used in order to obtain the 3D reconstruction of the foreground object that will result the output of the system.

Figure 9.3: Example of foreground 3D SCGMM from different points of view. Each ellipsoid is the spatial representation of the 3D SCGMM.

Spatial updating of the 3D model: The 3D object reconstruction will be used to update the 3D foreground model in order to adapt it to the movements that the foreground object performs at each frame. If the model is correctly initialized in the color and spatial domains, only a spatial updating will be necessary to achieve a correct characterization of the object since, unlike the 2D SCGMM, the 3D reconstruction does not present regions occluded to the camera.

Projection of the 3D SCGMM to 2D views: The final step of this work-flow consists in projecting the 3D SCGMM to each one of the views, in order to use the 3D model in the 2D foreground segmentation. Therefore, for each camera sensor, the 2D foreground model will be composed by the projection of the 3D Gaussians that model voxels which present direct visibility from the camera sensor.

The chapter is organized as follows: Section 9.2 describes the 3D foreground model. Section 9.3 explains the projection of the 3D SCGMM to the 2D views. Finally, some results and conclusions are presented in Section 9.4 and Section 9.5 respectively.

9.2 3D Foreground Model

In order to utilize the data redundancy that appear among views, we propose to characterize the foreground object by defining a 3D spatial probabilistic model. This model will gather all the information of the object under analysis, thus increasing the robustness of the multi-view segmentation process.

Since the foreground objects that appear in scene are constantly moving and changing along the sequence, we propose the 3D SCGMM at region based level to model the spatial (XYZ) and color (RGB) domains of the 3D object volume

Therefore, at each time t of the multi-view sequence, our objective is to obtain an updated model parameter set:
$\theta \equiv \{\hat{\omega}, \hat{\mu}, \hat{\Sigma}\} \equiv \{(\omega_1, \mu_1, \Sigma_1) \dots (\omega_k, \mu_k, \Sigma_k) \dots (\omega_{K_{3D}}, \mu_{K_{3D}}, \Sigma_{K_{3D}})\}$, that maxi-

mizes the foreground volume (V_t) data likelihood:

$$\theta_{V_t} = \arg\max_{\theta_{V_t}} \prod_{v_i \in V_t} \left[P(v_i|\theta_{V_t}) \right], \tag{9.1}$$

where $v_i \in \mathbb{R}^6$ is the input feature vector for voxel i in the $v = (RGB\ XYZ)$ domain and $P(v_i|\theta_{V_t})$ is the likelihood of voxel i formulated as follows:

$$P(v_i|\theta_{V_t}) = \sum_{k=1}^{K_{3D}} \omega_k G_{\text{fg}}(v_i, \mu_k, \Sigma_k), \tag{9.2}$$

where K_{3D} is the total number of Gaussians that belong to the foreground 3D SCGMM model and $G_{\text{fg}}(v_i, \mu_k, \Sigma_k)$ denotes the pdf of the k-th Gaussian formulated as:

$$G_{\text{fg}}(v_i, \mu_k, \Sigma_k) = \frac{1}{(2\pi)^3 |\Sigma_i|^{\frac{1}{2}}} \exp\left[-\frac{(v_i - \mu_k)^T \Sigma_k^{-1} (v_i - \mu_k)}{2} \right], \tag{9.3}$$

where $\mu_k \in \mathbb{R}^6$ is the mean of the 3D Gaussian and $\Sigma_k \in \mathbb{R}^{6\times 6}$ denotes its Covariance matrix.

Figure 9.3 displays an example of the 3D foreground model. As we can observe, the 3D SCGMM presents a non-flexible 3D modeling, thanks to the free movement that the 3D Gaussians present, thus adapting well to the real shape of the object without having any movement restrictions.

9.2.1 Initialization

An initial segmentation of the foreground object in each view is required in order to achieve its first 3D reconstruction. In order to achieve it, we use the planar foreground segmentation system proposed in Chapter 4 in each one of the views. Once the foreground object has been initialized and segmented in all the views, we use conservative Visual Hull reconstruction with tolerance $\tau = 1$, in order to achieve the voxelized 3D volume. This volume is colorized assigning to each voxel belonging to the surface of the volume, the color of the 2D pixels correspondent to the voxel projection.

Given this initial colored volumetric reconstruction, the foreground model parameter estimation can be reached via Bayes' development with the EM algorithm ([DLR+77]) in the $RGB\ XYZ$ domains. For this aim, we use only the surface voxels of the volume, since they are the only ones with useful information for the multi-view segmentation analysis, and thus, this will speed up the process.

We estimate how many Gaussians are needed for correctly modeling the object analogously to the proposal presented in Section 4.4.1.1, i.e. by analyzing the color histogram for this purpose.

After the initialization of the 3D SCGMM, next frames of the sequence will be processed by projecting this 3D foreground model to each one of the views. Hence, in frame t, we will use the projection of the model obtained from $t-1$, to carry out the 2D planar detection in each view. These planar foreground masks will make possible to achieve the 3D SfS reconstruction for frame t, which will be used, in turn, to update the 3D SCGMM before analyzing the next frame of the sequence.

9.2.2 Updating

Analogously to the previous chapters, the foreground objects perform some displacements and rotations along the scene that makes necessary the model updating at each frame. Since the probabilistic model works in the 3D XYZ domain, and the color of the object is correctly modeled from the initialization in the overall volume, only spatial updating is the necessary along the frames. We propose to update the components of the 3D Gaussian Mixture in the spatial domain, for frame t, in a two-steps updating, by using the 3D volumetric reconstruction obtained in the previous step.

9.2.2.1 Spatial Domain Updating

We use the color and spatial information of the voxels classified as foreground to update only the spatial components of the Gaussian Mixtures. Similarly to the initialization step, we will work with the surface voxels of the 3D volume. Hence, we assign each voxel to the Gaussian k that maximizes:

$$P(k|v_i,\theta_{V_t}) = \frac{P(v_i|\theta_{V_t},k)}{\sum_k P(v_i|\theta_{V_t},k)} = \frac{P(v_i|\theta_{V_t},k)}{P(v_i|\theta_{V_t})}, \tag{9.4}$$

where $P(v_i|\theta_{V_t})$ is the likelihood of the foreground model for the voxel i (defined in Equation 9.2), and $P(v_i|\text{fg},k)$ is the likelihood given by the Gaussian k. Once each voxel has been assigned to a Gaussian, the spatial mean and covariance matrix of each one are updated with the spatial mean and variances of the surface voxels that each one is modeling.

Regularization of the Gaussians displacements:

Once each Gaussian has been spatially updated, we regularize the displacements that each one suffers in the 3D space by using the information obtained from the neighbor Gaussians, thus achieving a more homogeneous spatial evolution of the 3D SCGMM. Hence, given the foreground parameter set $\theta_{V_{t-1}}$ before the spatial updating, and the parameter set after the updating: θ_{V_t}, we calculate the spatial displacements $d_{s=x,y,z} = (d_x, d_y, d_z)$ of the Gaussian k by computing: $d_{s,k} = (\mu_{s,k,t} - \mu_{s,k,t-1})$.

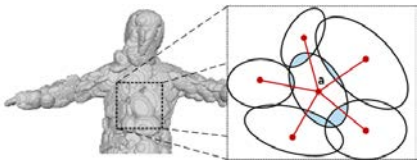


Figure 9.4: Example of neighborhood and connectivity between Gaussians that belong to the 3D foreground model. In blue color, the overlapped volume regions between the ellipsoid a under analysis, and the neighbor Gaussians.

We define this neighborhood according to the connectivity that each one presents in the surface of the volume with respect to the rest of the Gaussians. If we establish the 3D spatial representation of each Gaussian, as an ellipsoid whose axis (ϵ) are defined by the three eigenvalues of its spatial covariance matrix $(\lambda_1, \lambda_2, \lambda_3)$ as: $\epsilon_i = 2\sqrt{\lambda_i}$, then two Gaussians will be connected if both present an overlapped region of their spatial ellipsoids (formulated in Cartesian coordinates as: $\frac{(x-\mu_X)^2}{\epsilon_1} + \frac{(y-\mu_Y)^2}{\epsilon_2} + \frac{(z-\mu_Z)^2}{\epsilon_3} = 1$). Figure 9.4 shows an example of this connectivity where the Gaussian under analysis presents some overlapped regions with the rest of the Gaussians.

Hence, we propose a convolution between the set of displacements that the Gaussians suffer in the spatial updating d_s, and a Gaussian kernel (GK), thus smoothing the spatial evolution of the foreground Gaussians along the sequence obtaining the set of displacement vectors $\hat{d}_s$.

$$\hat{d}_{s,k} = \sum_{i_1,i_2,i_3}^{N_b} \mathrm{GK}(i_1, i_2, i_3) \cdot d(x + i_1, y + i_2, z + i_3), \tag{9.5}$$

where N_b is the neighborhood utilized in the Gaussian k smoothness. Hence, we maintain the consistency of the foreground model, in order to give robustness to the overall system.

Also, in order to achieve a better adaptation of the model into the silhouette of the object, we apply a Gaussian split criterion presented in Chapter 4 (Section 4.4.1.2) according to the spatial size of the Gaussian. Gaussians with big area are split into two smaller Gaussians in the direction of the eigenvector associated to the largest eigenvalue ($\lambda_{\max}$).

9.3 Projecting 3D Foreground Model to 2D

The 3D foreground model gathers all the information of the foreground object that we want to segment and reconstruct. In order to use it for 2D foreground segmentation in each view, we need to project the 3D Gaussians to each one of the cameras according to the visibility that the surface voxels present from every view.

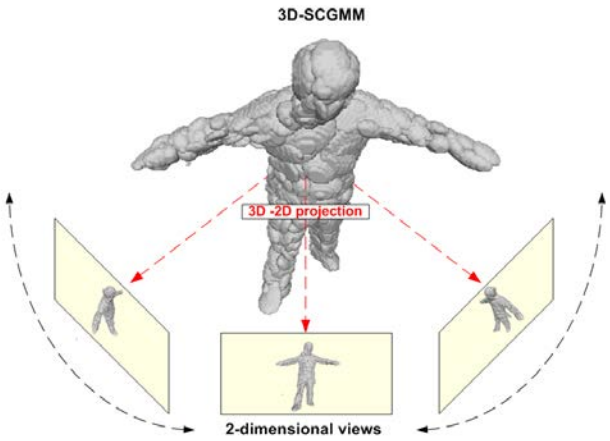


Figure 9.5: 3D SCGMM projected to the 2-dimensional views.

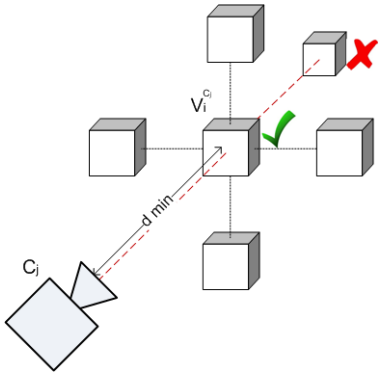


Figure 9.6: Visibility test. Graphical representation.

Figure 9.5 shows a graphical representation of the 3D SCGMM projection. Hence, a $\mathbb{R}^3 \rightarrow \mathbb{R}^2$ projection is proposed in each camera sensor C_j:

First, a visibility test of the surface voxels is performed for each one the views. We consider only the foreground voxels that are visible from camera C_j thus rejecting all those foreground voxels that appear occluded by the visible ones. As we can observe in Figure 9.6, the visibility test consists in obtaining the distance from the sensors to each one of the foreground voxels, thus obtaining the minimum distance d_{min} in each projection line corresponding to the closer voxel to the camera. Applying this to each one of the camera sensors, we obtain the bag of visible voxels ν for each view: ν^{C_j}.

Next, we assign each voxel $v_i \in \nu^{C_j}$ to the 3D Gaussian k that maximizes the Equation (9.4), thus obtaining the group of Gaussians that model visible voxels from each one of the views ζ^{C_j}.

Therefore, for each one of the views C_j, we project the visible Gaussians belonging to ζ^{C_j} according to the projection matrices and focal length that each camera

sensor presents. These Gaussians will be used in the 2D planar foreground segmentation for each camera according to the proposal of Chapter 4.

9.4 Results

We have evaluated the multi-view segmentation system presented in this chapter by analyzing four multi-view sequences of the database published in [INR] recorded by means of different acquisition setups. The data set used for this evaluation is the same data set used in Chapter 7 and Chapter 8, and has been presented at the beginning of Section 7.5. In these tests we want to evaluate the viability of the 3D SCGMM to represent the foreground object in the 3-dimensional space, and the subsequent 2-dimensional foreground segmentations that take place in each view by means of the 3D model projection to the 2D images. Hence, we will show in this section qualitative and quantitative results of the current proposal.

For each one of the sequences, the work-flow presented in Figure 9.2 has been applied in order to obtain the 3D SCGMM of the objects under analysis. Figure 9.7 displays the spatial representation of the models created in each one of the sequences. We can observe how these models adapt well to the shape of the objects achieving a complete 3D characterization. Analogously to the 2D SCGMM, the number of Gaussians of the model determines the precision of the modeling: the higher the number of Gaussians of the model, the better the definition of the 3D SCGMM, but the computational cost will increase proportionally. In this evaluation, around one hundred Gaussians have been used for each model in order to achieve a correct characterization of the foreground object.

Complete qualitative results are displayed in Figure 9.9 and Figure 9.10, where four frames of each sequence are displayed. In second column we can observe the projection of the 3D SCGMM to the view under analysis. Here, the Gaussians of the 3D model are projected to the view only if they model any of the visible voxels obtained for each camera ν^{C_j}. Each Gaussian is drawn with the mean RGB color that each one is modeling, and we can observe how the 2D spatial-color representation adjust correctly to the real shape of the object.

In the third column we can see the 2D foreground segmentation obtained by using the 3D probabilistic model(depicted in second column) in the Bayesian MAP-MRF foreground segmentation explained in Chapter 4. This segmentation achieves correct results also in those regions where foreground and background present camouflage situations. The robustness added by the 3D modeling avoids independent 2D errors to be propagated in consecutive frames.

Fourth column shows the 3D volumetric reconstruction with Tolerance to errors

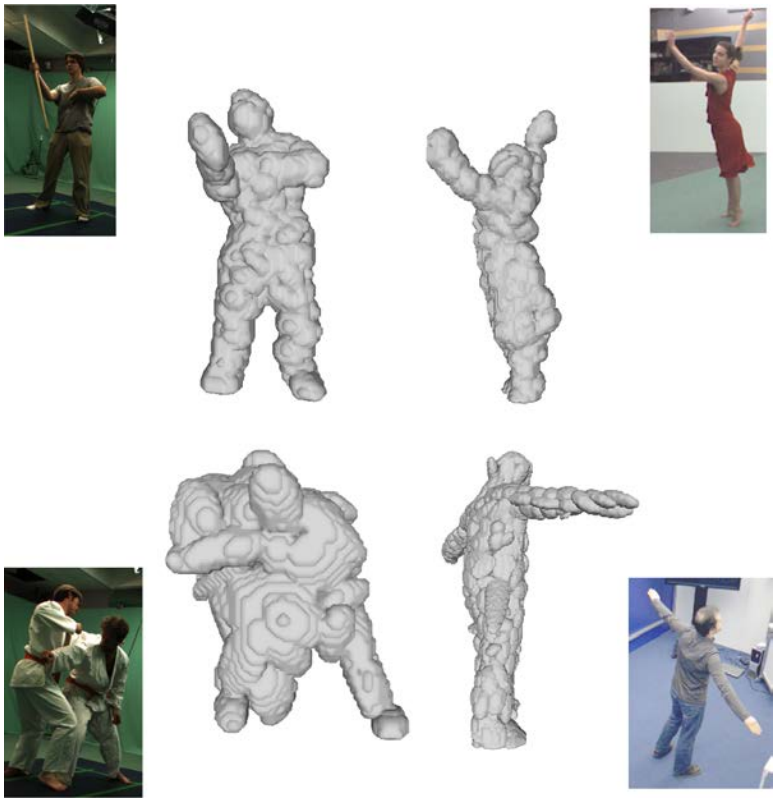


Figure 9.7: Resultant foreground 3D SCGMM. Each ellipsoid represents one Gaussian of the foreground model projected to one 2D view.

$\tau = 1$, computed with all the 2D silhouettes of the multi-view sequence and projected to the views under analysis. We can observe that the final reconstruction presents correct results since we reduce the percentage of errors in the 2D silhouettes. In order to depict the color modeling that the foreground 3D SCGMM is applying to the 3D object, fifth column shows the volumetric reconstruction of the object where each foreground voxel is colored with the RGB color of the Gaussian that better represents it, according to the Equation 9.4. Hence, we can realize that the 3D SCGMM achieves a correct color-spatial representation of the object along the sequence.

Some quantitative results are displayed in Figure 9.11, where we have analyzed the resultant 2D foreground segmentation computed in the first view of each one of the sequences (Third column of Figure 9.9 and Figure 9.10). The data set utilized is the same as the one presented in Section 7.5, where ten representative and equally-distributed frames of each sequence have been used to compare the results with the ground truth segmentation. We have computed the f_{measure} metric in order to compare the 2D foreground segmentation obtained by the method presented in this chapter (3D SCGMM), with the foreground segmentation system presented in Chapter 4 (Bayes+sh.rem.).

As we can observe, when 3D foreground segmentation model is correctly initialized, the results present a very low percentage of false positive and false negative errors, very similar to the results obtained by the method presented in Chapter 4 when the 2D model represents correctly the object to segment. Using the 3D model, the effects of false positive and false negative errors in the sequences are less strong than analyzing a single view, since the information of the rest of the views helps to maintain the robustness of the foreground model in each one, thus allowing to overcome these situations faster. We can see an example of this in Figure 9.11(d), where in frame 480 a difficult situation in the object detection, leads the Bayes+sh.rem. system to loose 0.03 points of f_{measure} from 0.955 to 0.92.5, while the 3D SCGMM method present a reduction of 0.015, from 0.95 to 0.94.

The results obtained in Figure 9.11 are summarized in Table 9.1, where the Precision, Recall and f_{measure} results of the frames compared with the ground truth are displayed for each one of these sequences. Again, we can see how the overall results are very similar to the ones obtained by means of method Bayes+sh.rem., since, when the models are correctly initialized, both approaches present similar features. Note that only strong false negative errors in the 3D volumetric reconstruction could lead to errors in the 3D probabilistic modeling, which could propagate the errors to next frames of the sequence, thus producing a degeneration of the 3D SCGMM.

Finally, Figure 9.8 depicts an example of the effects produced by the regularization of the Gaussians displacements in the spatial updating process. As we can see, this part of the spatial updating helps to maintain the robustness of the foreground model when false negative regions appear in the 3D reconstruction. Since the 3D foreground model is updated with the projection of the volumetric reconstruction to the view under analysis, if no regularization is applied (Figure 9.8(b)), the Gaussians of the model are spatially displaced to the foreground regions, thus propagating false negative errors to the next frames of the sequence. On the other hand, when applying the regularization process (Figure 9.8(c)), strong variations of the model, due to false negative regions, are smoothed, thus maintaining the spatial structure of the model.

Regarding the computational cost, each one of the processes that appear in the work-flow of Figure 9.2 spends an important part of the overall time: If we implement the 2D foreground segmentation in a parallel structure, we can obtain a computational cost according to the tables presented in Chapter 4 for this step. The 3D volumetric reconstruction is computed with a SfS technique, by means of a real-time processing. The projection of the 3D SCGMM to the 2D views can also be implemented, in a parallel way by computing all the views at once, thus reducing the computational burden. The 3D SCGMM updating presents an important computational burden since the number of voxels to analyze can be high and thus, computationally expensive to work with. With these requirements, and

considering a foreground model with no more than 100 Gaussians, we approximate a computational cost of 0.08 frames/second, analyzing a standard sequence and using an Intel Core2 Duo 3GHz processor and 20 GB RAM. Since the objective of this research was only to propose a new framework for multi-view foreground segmentation and 3D reconstruction, no optimization work has been carried out in order to reduce this number. Hence, this computational cost can be improved by developing more efficient algorithms which could work over GPU.

Table 9.1: Quantitative results

Sequences	**Method**	**Precision**	**Recall**	$f_{\textbf{measure}}$
Stick	3D SCGMM	$0,98$	0.97	0.98
	Bayes+sh.rem.	$0,97$	0.94	0.96
Dancer	3D SCGMM	0.96	0.96	0.96
	Bayes+sh.rem.	0.94	0.97	0.95
Karate	3D SCGMM	0.97	0.97	0.97
	Bayes+sh.rem.	0.98	0.98	0.98
Open arms	3D SCGMM	0.92	0.97	0.95
	Bayes+sh.rem.	0.95	0.95	0.95

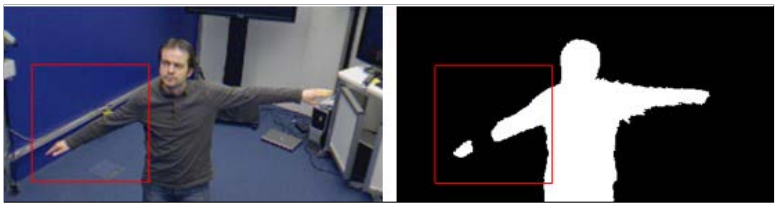
9.5 Conclusions

We have presented in this chapter of the manuscript a foreground segmentation system for multi-view smart-room scenarios that uses a parametric non-rigid probabilistic model to characterize the object under analysis in the 3D space. We have called this model 3D SCGMM and, as in the case of the SCGMM explained in Chapter 4 and utilized in all the developments presented in this thesis, it is performed by color-space Gaussians but applied, in this framework, to the 3D XYZ space. Hence, we have proposed this new technique to develop a multi-view foreground segmentation system, which combines the information obtained from each one of the views to define the 3D SCGMM for the 3D volumetric representation of the object under analysis.

As we have seen in this chapter, this probabilistic modeling of the object achieves a robust representation of the foreground object, which is projected to each view to perform a Bayesian foreground segmentation (introduced in Chapter 4). This system achieves correct results, by reducing the false positive and false negative errors in sequences where some camera sensors can present camouflage situations between foreground and background. Since the foreground segmentation process, and in general, all the work-flow of the system is based on the probabilistic modeling of the object, the initialization step must be correct in order to avoid errors in the

object modeling. Moreover, since at each frame the probabilistic model is updated with the final 3D object reconstruction, errors in final reconstruction could lead to errors in the color-spatial representation of the object which could be propagated to next frames of the sequence.

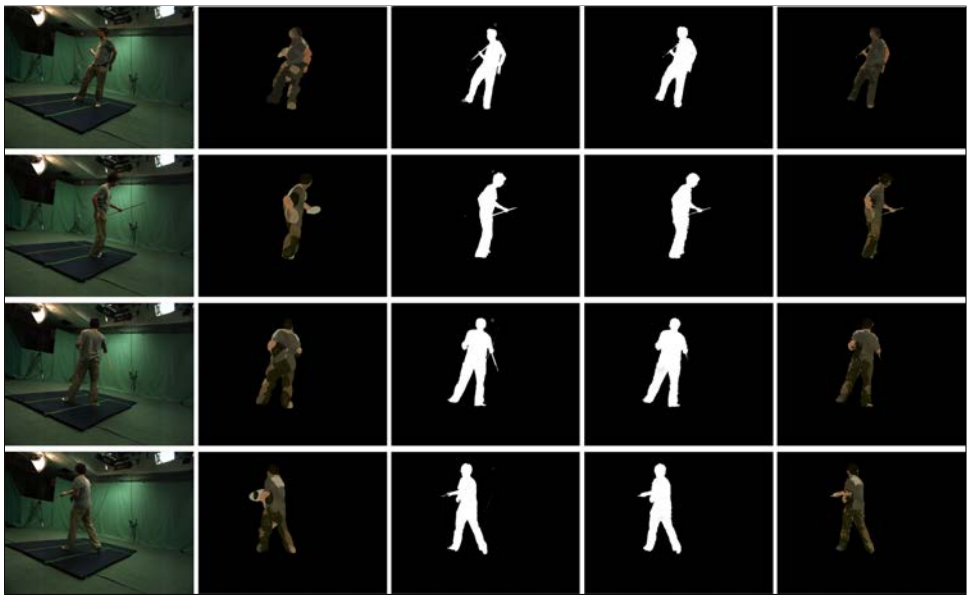
Finally, we would like to introduce the possibilities that this model could represent in objects recognition or human activity understanding. Figure 9.12 shows an example of the evolution of the model in consecutive frames for the sequence *dancer*. In this figure, we can observe how the Gaussians of the model perform a movement along the sequence according to the real one performed by the object. Although the model is non-rigid, and the Gaussians are not spatially linked one another, the evolution of the model is soft and progressive (thanks to the regularization of the Gaussians displacements) and the Gaussians of the model are associated to the real regions of the object, which are, in fact, the regions that each one is better modeling. Therefore, as we show in Figure 9.12, and similarly to other approaches of the state of the art like [GFBP10] new direction in dense geometric and temporal 3D analysis can be exploited by using the 3D SCGMM probabilistic modeling in multi-view foreground segmentation and 3D reconstruction analysis.



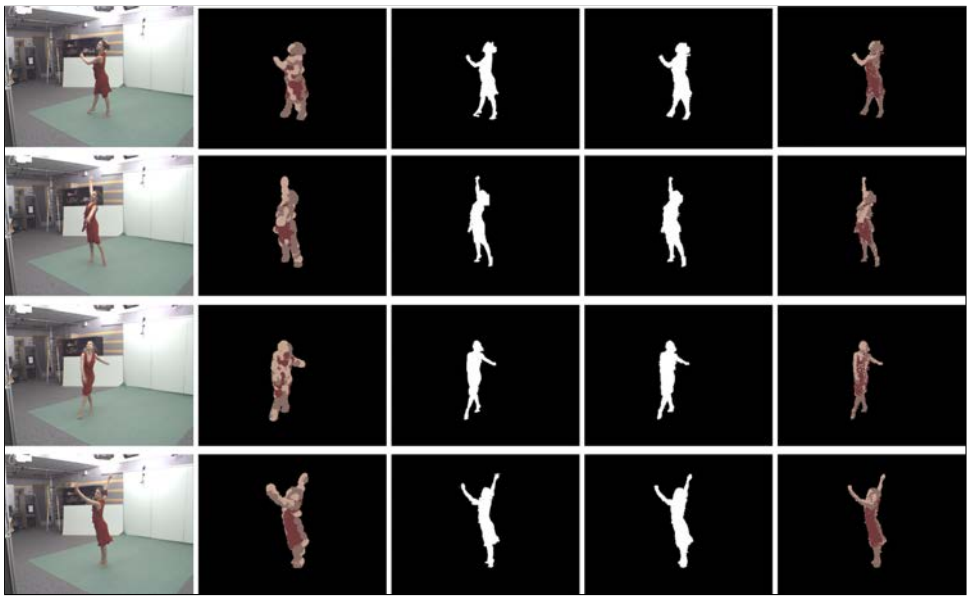
(a) Original image the 3D volume projection to the view under analysis. A false negative error appear in the 3D object reconstruction.

(b) (c)

Figure 9.8: Example of the effect of the Gaussians displacements regularization. Figure 9.8(a) shows the 3D reconstruction with a false negative region when reconstructing the arm of the person under analysis. b) displays the updating results of the 3D SCGMM without applying the regularization of the Gaussians displacements. c) depicts the updating results when applying the regularization of the Gaussians displacements.



(a) Stick sequence. 16 cameras.



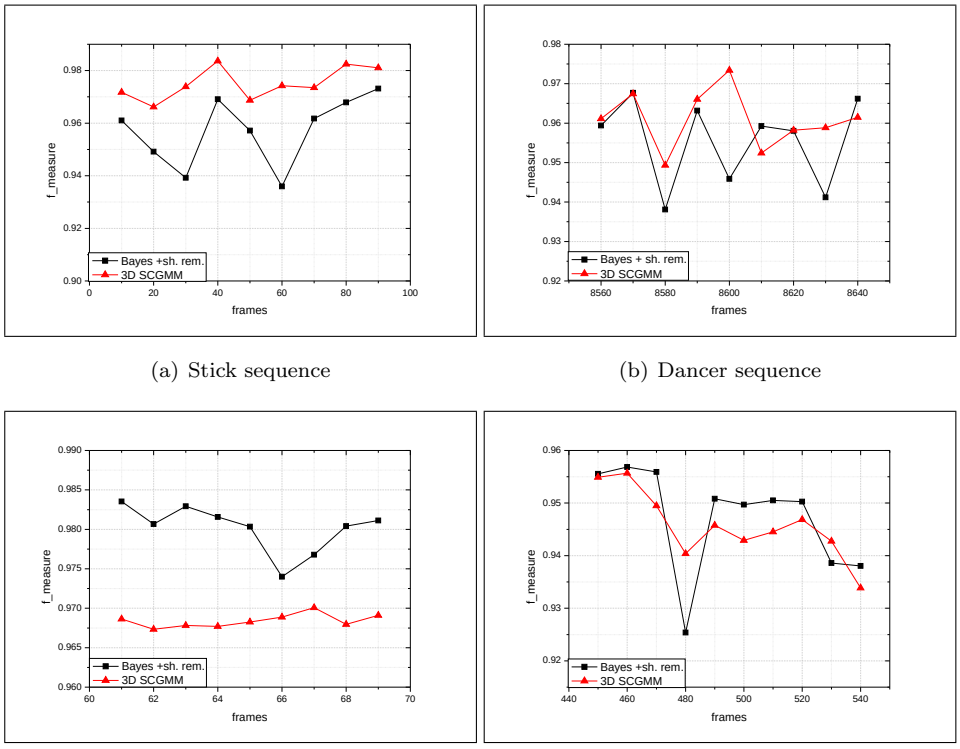
(b) Dancer sequence. 8 cameras.

Figure 9.9: Qualitative results. In the first column, the original frames. Second column shows the 3D SCGMM projection to the view under analysis, where each ellipse represents one Gaussian of the model with the mean color that each one is modeling. Third column is the 2D foreground segmentation obtained by means of the model depicted in second column. Fourth column displays the 3D reconstruction projected to the view under analysis, obtained by means of the foreground segmentation of each view. Fifth column is the 3D reconstruction where each voxel is colored with the mean RGB color value of the 3D Gaussian that better represents the voxel (according to Equation 9.4).

(a) Karate sequence. 16 cameras.

(b) Open arms sequence. 18 cameras.

Figure 9.10: Qualitative results. In the first column, the original frames. Second column shows the 3D SCGMM projection to the view under analysis, where each ellipse represents one Gaussian of the model with the mean color that each one is modeling. Third column is the 2D foreground segmentation obtained by means of the model depicted in second column. Fourth column displays the 3D reconstruction projected to the view under analysis, obtained by means of the foreground segmentation of each view. Fifth column is the 3D reconstruction where each voxel is colored with the mean *RGB* color value of the 3D Gaussian that better represents the voxel (according to Equation 9.4).



(a) Stick sequence

(b) Dancer sequence

(c) Karate sequence

(d) Open arms sequence

Figure 9.11: Quantitative results.

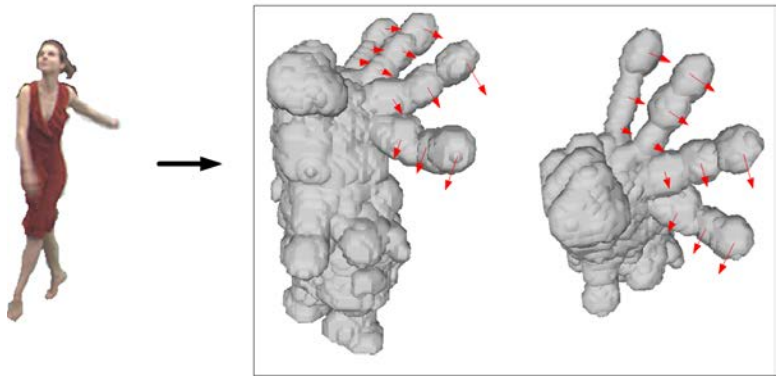


Figure 9.12: Tracking Gaussians for human activity understanding. The 3D SCGMM of four consecutive frames of the sequence are depicted together in order to represent the evolution of the 3D Gaussians. Red arrows are hand made to represent the evolution of the movement that each Gaussian performs.

Chapter 10

Conclusions and Future Work

In the development of this thesis entitled *Parametric Region-Based Foreground Segmentation in Planar and Multi-View Sequences*, novel proposals for foreground segmentation in monocular and multi-view sequences have been presented with the objective to improve the existing techniques of the state of the art in this image processing area. After an in-depth study of the main reference work of the literature, summarized in Chapter 3, we detected the weakness and necessities present in each one of the specific frameworks, according to the characteristics of the scenario and the acquisition setups, and we developed new techniques to solve them in order to improve the resultant foreground segmentation. The consequence of this research has been explained in this manuscript organized as a research work starting from static 2D planar foreground segmentation systems, and the generalization of these methods to the multi-view foreground segmentation and 3D reconstruction techniques. In this thesis we have demonstrated that the use of region-based parametric models for modeling the classes, provides a correct color-spatial modeling of the regions that can be used to improve the foreground segmentation results in 2D planar scenarios as well as in multi-view setups.

10.1 Contributions

Each one of the chapters presented in this thesis, belonging to the research Part I and Part II, deals with one specific scenario where foreground segmentation is necessary, and each one contributes to improve the state of the art on that area. The contributions of this thesis are listed below:

10.1.1 Contributions to Foreground Segmentation in 2D Planar Scenarios

The research carried out in Part I is devoted to this scenario. In Chapter 4, we have presented a complete foreground segmentation system that combines in a Bayesian MAP-MRF framework, a pixel-wise background model (Gaussian pixel model) with parametric region-based foreground (SCGMM) and shadow models (SCGM). All the system runs as an implementation of the simple concept of surveillance: be aware for external changes, detect new objects that appear in the scene, and focus on the new objects by improving the information about it. The system has proved to achieve correct results also in those regions where foreground and background present camouflage problems. The contributions to the state of the art resides on:

- **Work-Flow design.** The work-flow of the system, which proposes the combination of an exception to background analysis with a tracking system to perform the detection and management of new objects that appear in the scene. Once a new object is detected, foreground and shadow region-based models are created and associated to it in order to achieve a correct characterization.
- **Shadow model.** The creation of the region-based SCGM to model the shadow regions that each object projects on the scenario. This model is associated to each foreground object as well as the foreground model, in order to remove the shadow regions locally, without creating false negative errors inside the shape of the object due to false shadow detections.
- **Probabilistic models.** The combination of pixel-wise background model with region-based foreground and shadow models is also an important part of the overall system, since the difference of dimensionality made it a difficult task to solve. Also, the Bayesian classification step between foreground, shadow and background models in a MAP-MRF framework has supposed a real improvement to the final results.
- **Foreground SCGMM updating.** In order to achieve a correct updating of the foreground SCGMM, and speed up the process, we have proposed an alternative to the EM algorithm to update the Gaussian model in the color and spatial domains. This updating not only updates the Gaussians parameters to the new foreground detection obtained at each frame, but also updates the number of Gaussians of the foreground model in order to adapt well to the real shape of the object along the sequence.

The system proposed in this chapter have been used in the following ones as an starting point for other improvements of the state of the art.

In Chapter 5 we have proposed an application of the principles proposed in Chapter 4 to those sequences recorded by means of moving camera, where one object of interest must be segregated from the background. In this system, we combine two SCGMM for the foreground and background inside a Region of Interest (ROI) in a MAP-MRF classification framework. The ROI is designed in order to achieve the foreground model to be in the middle of the ROI, surrounded by the background one. Supposing stationarity of foreground and background regions during the classification process, and that new background regions are modeled first by the background model, the approach of Chapter 5 offers correct foreground segmentation for moving camera sequences and an alternative to other reference methods.

10.1.2 Contributions to Foreground Segmentation in Multi-View Scenarios

Part II of this thesis gathers all the research developed for multi-view scenarios. In this part, four proposals have been presented with the consequent contributions:

Chapter 6 presents a foreground segmentation system for sequences recorded by means of color RGB + depth Z sensors. This system allows us to achieve a correct foreground segmentation also when camouflage problems arise in one of the sensors. The contributions of this system are:

- **Probabilistic models.** We define foreground SCGMM and background pixel-wise color Gaussian model for the color camera, and foreground SDGMM and background pixel-wise depth Gaussian model, thus resulting four probabilistic models.
- **Combine probabilities with Logarithmic Opinion Pool.** For each class, we combine the probability provided by each sensor by means of the logarithmic opinion pool technique. This technique consists in the sum of the weighted log-likelihood probabilities obtained from each sensor in order to obtain a mixed probability according to the reliability that each sensor presents.
- **Reliability maps using Hellinger distance.** We propose to use the Hellinger distance in order to achieve the reliability maps for each sensor. Hence, we compute this distance between foreground and background models in the color RGB and depth Z domain to obtain the final weight for each sensor.

Chapter 7 shows the first proposal of collaborative foreground segmentation and 3D reconstruction in multi-view smart-room scenarios with the objective to

achieve a reliable 3D volumetric reconstruction based on SfS. The system achieves a conservative 3D reconstruction of the object under analysis by applying tolerance to errors only when the sensors present a non-reliable detection. Results displayed in Chapter 7 show how the robust 3D reconstruction improves the results of the conservative reconstructions with fixed tolerance to errors. The contributions of this chapter are:

- **Use reliability maps in the 3D reconstruction step.** For each camera sensor, we apply the foreground segmentation system proposed in Chapter 4 in order to obtain the 2D silhouettes of the object to segment. We propose to obtain the reliability maps of each camera computing the Hellinger distance between foreground and background in the RGB domain. The reliability of each pixel in each sensor is taken into account in the Visual Hull reconstruction, thus avoiding the cameras where the pixel under analysis detect background and present low reliability.

Chapter 8 is devoted to explain the second proposal of collaborative foreground segmentation and 3D reconstruction. Since the 3D volume reconstruction can be interpreted as the combination of the information shared by all the camera sensors, in this chapter, we propose to use the 3D volume reconstruction in order to update the 2D foreground models defined in each camera sensor. Tolerance to error reconstruction is used to carry out this updating in an iterative way according to the tolerance to error used in the reconstruction. The results obtained with this system, shows that the foreground segmentation and the 3D reconstruction can be improved implementing this feedback between processes. The main contribution of this system is:

- **Iterative volume reconstruction in a 3D tolerance loop.** As in the previous approach, for each camera sensor, we apply the foreground segmentation system proposed in Chapter 4 in order to obtain the 2D silhouettes of the object to segment. Since 3D reconstruction with tolerance to errors avoids the propagation of silhouette misses, we propose a loop based on enhanced conservative Visual Hull reconstruction with error tolerance to update the foreground segmentation. At each iteration, the 2D foreground models are updated with the projected 3D volume and a new foreground segmentation is performed with less misses, which is used to iteratively achieve a more precise and robust foreground segmentation and 3D reconstruction.

Finally, Chapter 9 shows the third proposal in 2D-3D cooperative systems. In order to achieve a more general multi-view foreground segmentation, we propose the 3D SCGMM to model the foreground object in the 3D XYZ space, instead of maintain a separated foreground model for each one of the camera sensors. This system

achieves a more robust framework for foreground modeling, since we centralize the foreground model in the 3D space, updating it with the 3D reconstruction obtained by all the camera sensors. The results of this system show how this proposal can be a good alternative to develop multi-view segmentation systems avoiding updating errors in cameras where foreground and background present color similarity. The contribution of this chapter is:

- **3D model for multi-view foreground segmentation.** We propose the 3D SCGMM to model the volumetric reconstruction of the object in the $RGB\,XYZ$ space. Once the model is created with an initial object reconstruction, for next frames, it is projected to each camera in order to perform the 2D foreground segmentation and the subsequent 3D volume reconstruction. The model is updated in the spatial domain with the resultant 3D volume, and smoothed in order to avoid misses. This model can also be used to perform a geometry or temporal 3D analysis over the objects under analysis.

10.2 Publications and Collaborations

Part of these contributions have been published in journal and conference papers:

- **Conference papers**
  - Gallego, J., Pardas, M., Haro, G. **Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model.** Proc. IEEE Int. Conf. on Image Processing, 2009, pp. 3205-3208.
  - Gallego, J., Pardas. **Enhanced Bayesian foreground segmentation using brightness and color distortion region-based model for shadow removal.** Proc. IEEE Int. Conf. on Image Processing, 2010, pp. 3449-3452.
  - Gallego, J., Salvador, J., Casas, J.R., Pardas, M. **Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop.** Proc. IEEE Int. Conf. on Image Processing, 2011, pp. 997-1000.
  - Gallego, J., Pardas, M., Solano, M. **Foreground objects segmentation for moving camera scenarios based on SCGMM.** Lecture Notes in Computer Science: Computational Intelligence for Multimedia Understanding, 2012, pp. 195-206.

- **Journal papers**
  - Gallego, J., Pardas, Haro, G. **Enhanced foreground segmentation and tracking combining Bayesian background, shadow and foreground modeling.** Pattern Recognition Letters, Springer, num. 33, 2012, pp. 1558-1568.
  - Gallego, J., Pardas. **Region based foreground segmentation combining color and depth sensors via logarithmic opinion pool decision.** Journal of Visual Communication and Image Representation, Elsevier, 2013.

Parts of the contributions and investigations conducted in this dissertation have been undertaken in answer to the challenges raised by some of the projects where the Image Processing Group of the UPC has been involved. In particular, this work has been developed within the framework of the Spanish projects HESPERIA (Homeland sEcurity: tecnologíaS Para la sEguridad integRal en espacIos públicos e infrAestructuras), Vision (Comunicaciones de Vídeo de Nueva Generación), i3media (Management of multimedia content) and the European project FASCINATE (Format-Agnostic SCript-based INterAcTive Experience).

10.3 Future work

The work presented in this manuscript can be continued by following several research lines that can improve the performance of the systems proposed in previous chapters. These research lines are:

- In planar foreground segmentation, the updating process of the models is carried out by means of the segmentation obtained in the current frame. Hard foreground detection errors can appear and, in these situations, the updating could lead to a wrong modeling of the object, with the consequence of possible errors propagation in next frames. Although this situation rarely appears, one possible research line could focus on the updating processes and the hard errors detection in the segmentation in order to control better the evolution of process.
- In multi-view foreground segmentation, the position of the cameras in the acquisition setup can be also studied and incorporated in the analysis methods in order to improve the collaboration between sensors. Therefore, when using the robust SfS, the 3D reconstruction should not consider only the reliability maps, but also the relative position of the cameras in order to combine better the sensors' information.

- The use of energy minimization techniques to regularize the resultant masks could be improved by using also, in the global optimization, the information belonging from the previous frames, thus adding robustness to the classification process, by taking into account the evolution of the masks along the sequence.

- Other color spaces can be incorporated in the foreground segmentation process, in order to develop parallel segmentations in different domains, which could help to improve the foreground detection process.

- In multi-view foreground segmentation and 3D reconstruction, the 3D model research can be continued in order to use it for activity recognition, object identification or object's geometry analysis. A possible line of research could be the combination of the foreground model with existent human body models to improve the performance of the system.

- Real-time implementation of some proposals can be addressed in the future by means of parallel processing over GPU.

Appendix A

Parametric Model GMM

Foreground segmentation methods based on parametric models like Gaussian distributions are widely used in foreground/background classification. In this thesis, we propose to use the parametric Gaussian Mixture Model to probabilistically model the regions under analysis. In this Chapter we are going to see an in depth analysis of this kind of probabilistic models.

The use of parametric models to approximate probability density functions is a common technique utilized in classification problems where either we cannot obtain an analytical description of the real one, or despite we have it, is too complex to work with. Therefore, in order to simplify the mathematical operations implied in the classification process, the approximation of each class by using parametric models will give us a reliable framework that will speed-up all the related processes.

The parametric models are a family of distributions that can be described using a finite number of parameters. These parameters are usually collected together to form a single n-dimensional parameter vector $\Theta = \{\theta_1, \theta_2, \theta_3, ..., \theta_n\}$. The model is formulated as:

$$\mathcal{P} = \left\{ P_\theta \;\middle|\; \theta \in \Theta \right\} \tag{A.1}$$

One of the most prominent parametric models is the Gaussian Distribution. This distribution is present in a huge number of natural processes, and arises from the central limit theorem, which states that given general conditions, the mean of a sufficiently large number of independent random variables, each with finite mean and variance, will be approximately normally distributed, irrespective of the form of the original distribution. This gives it exceptionally wide application in several areas like machine learning and classification.

A.1 Gaussian Distribution

The Gaussian distribution, is a bell-shaped unimodal continuous probability distribution that belongs to the exponential family. It is parametrized by $\Theta = (\mu, \sigma)$, where $\mu \in \mathbb{R}$ is the mean (location of the peak), and $\sigma > 0$ is the standard deviation as well as σ^2 is the variance (the measure of the width of the distribution). This function is used as a simple model for complex phenomena. The distribution has a probability density function formulated as follows:

$$P(v|\mu, \sigma) = \frac{1}{\sigma\sqrt{(2\pi)}} \quad \exp\left[-\frac{(v-\mu)^2}{2\sigma^2}\right], \tag{A.2}$$

where $v \in \mathbb{R}$ is the input data.

The factor $\frac{1}{\sigma\sqrt{(2\pi)}}$ in this expression works as a normalization factor, and ensures that the total area under the Gaussian curve is equal to one.

The exponent factor $\frac{(v-\mu)}{\sigma}$ corresponds to the Mahalanobis distance, which is an euclidean distance normalized by the standard deviation of the distribution, thus obtaining a distance in terms of standard deviations to the center of the distribution.

The 1/2 in the exponent makes the "width" of the curve (measured as half the distance between the inflection points) equal to σ.

A.2 Multivariate Gaussian Distribution

When working with multi-dimensional spaces, we will need to use the Multivariate Gaussian distribution, which is a generalization of the one-dimensional (univariate) Gaussian distribution to higher dimensions. It is parametrized by $\Theta = (\mu, \Sigma)$, where $\mu \in \mathbb{R}^n$ is the mean, and $\Sigma \in \mathbb{R}^{n \times n}$ is the covariance matrix. The probability density function is written as:

$$P(v|\mu, \Sigma) = \frac{1}{(2\pi)^{n/2}|\Sigma|^{1/2}} \quad \exp\left[-\frac{1}{2}(v-\mu)^T\Sigma^{-1}(v-\mu)\right], \tag{A.3}$$

where v is the n-dimensional input data vector.

The covariance matrix Σ deserves special attention because it gives us information about the linear dependence that appears among the different domains of the Gaussian distribution and, therefore, will determine its shape. The covariance matrix is symmetric positive semidefinite.

If the n dimensions are independent, the Covariance Matrix will present a diagonal form:

$$\begin{pmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_n^2 \end{pmatrix} \tag{A.4}$$

we can express the Multivariate Probability Density Function as the product of n independent univariate Gaussian distributions with mean μ_i and variance σ_i^2:

$$\begin{aligned} P(v|\mu,\Sigma) = & \frac{1}{\sigma_1\sqrt{(2\pi)}} \exp\left[-\frac{(v_1-\mu_1)}{2\sigma_1}\right]^2 \cdot \frac{1}{\sigma_2\sqrt{(2\pi)}} \exp\left[-\frac{(v_2-\mu_2)}{2\sigma_2}\right]^2 \cdot \cdots \\ & \cdot \frac{1}{\sigma_n\sqrt{(2\pi)}} \exp\left[-\frac{(v_n-\mu_n)}{2\sigma_n}\right]^2, \end{aligned} \tag{A.5}$$

A.3 GMM Formulation

When we need to model a complex multi-modal distribution, one Gaussian function is not sufficient to give enough fidelity to the model. One possible option to achieve a representation of the multi-modal surface consists in using a Combination of several Gaussian distributions. This approach is called Gaussian Mixture Model (GMM). A Gaussian Mixture Model (GMM) is a parametric probability density function represented as a weighted sum of Gaussian component densities. The pdf of the overall model is formulated as follows:

$$\begin{aligned} P(v|\Theta) &= \sum_{k=1}^{K} \omega_k G_{\text{fg}}(v, \mu_k, \Sigma_k) \\ &= \sum_{k=1}^{K} \omega_k \frac{1}{(2\pi)^{n/2}|\Sigma_k|^{1/2}} \exp\left[-\frac{1}{2}(v_k-\mu_k)^T\Sigma_k^{-1}(v_k-\mu_k)\right], \end{aligned} \tag{A.6}$$

where K is the number of Gaussian distributions that compound the model, w_k is the mixture coefficient of the k-th Gaussian distribution where $\sum_{w_k} = 1$, $\mu_k \in \mathbb{R}^n$ and $\Sigma_k \in \mathbb{R}^{n\times n}$ are, respectively, its mean and covariance matrix, $|\Sigma_k|$ is the determinant of matrix Σ_k.

GMM parameters can be estimated from training data using the iterative Expectation-Maximization (EM) algorithm or Maximum A Posteriori (MAP) estimation from a well-trained prior model.

A.4 Expectation Maximization

Expectation-Maximization algorithm (EM) [DLR+77] is a well established maximum likelihood algorithm for fitting a mixture model to a set of given data. EM is an iterative algorithm that requires an a priori configuration to define the number of K components to be incorporated into the model. Often a suitable number may be selected by a user, roughly corresponding to the number of distinct colors appearing in an object to be modeled. The EM iteration alternates between performing an expectation (E) step, which creates a function for the expectation of the log-likelihood evaluated using the current estimate for the parameters, and a maximization (M) step, which computes parameters maximizing the expected log-likelihood found on the E step. These parameter-estimates are then used to determine the distribution of the latent variables in the next E step. The algorithm is defined as follows:

1.- E step: calculate the Gaussian component assignment probability for each pixel z:

$$P^{(i)}(k|v) = \frac{\omega_k^{(i)} \cdot P(v|\theta_k^{(i)})}{\sum_{k=1}^{K} \omega_k^{(i)} P(v|\theta_k^{(i)})}, \tag{A.7}$$

where i denotes the number of iteration, K is the number of mixture components involved in the process and $\theta_k = \{\mu_k, \Sigma_k\}$.

2.- M step: update the spatial and color means and variances, and the weight of each Gaussian component as:

$$\mu_k^{(i+1)} = \frac{\sum_V P^{(i)}(k|v) \cdot v}{\sum_V P^{(i)}(k|v)}, \tag{A.8}$$

$$\Sigma_k^{(i+1)} = \frac{\sum_V P^{(i)}(k|v) \cdot (v - \mu_k^{(i+1)}) \cdot (v - \mu_k^{(i+1)})^T}{\sum_V P^{(i)}(k|v)}, \tag{A.9}$$

$$\omega_k^{(i+1)} = \frac{\sum_V P^{(i)}(k|v)}{\sum_{k=1}^{K} P^{(i)}(k|v)}, \tag{A.10}$$

where V denotes all the input data samples under analysis.

Appendix B

Energy Minimization Via Graph Cuts

Many vision problems, especially in early vision, can naturally be formulated in terms of energy minimization. The classical use of energy minimization is to solve the pixel-labeling problem, which is a generalization of such problems as stereo, motion, and image restoration . The input is a set of pixels $I = \{I_1, I_2, ...I_i...I_N\}$ and a set of labels l. The goal is to find a labeling f (i.e., a mapping from I to l) which minimizes some energy function [SS05].

Hence, for a video sequence taken by a fixed camera, the foreground segmentation can be formulated as follows [SS05, GPS89].

Each frame image contains N pixels. Let S be the set of indices referring to each of the N pixels. Given a set of pixels I, S of current frame at time-step t, the task of object detection is to assign a label $l_i \in \{\text{background}(=0), \text{foreground}(=1)\}$ to each pixel $i \in S$, and obtain $l = \{l_1, l_2, \ldots l_i \ldots l_N\}$.

In most of the work in the literature, object detection was attempted by first modeling the conditional distribution $P(I_i|l_i)$ of feature value I_i at each pixel i independently. The model used can be either parametric [WADP02, SG00] or non-parametric [EHD00, SS05] based on a past window of observed feature values at the given pixel. The background and foreground model will be detailed presently. Assume the observed feature value of image pixels are conditionally independent given l, thus:

$$P(I|l) = \prod_{i=1}^{N} P(I_i|l_i). \tag{B.1}$$

However, it is clear that neighboring labels are strongly dependent on each other. The neighborhood consistency can be modeled with a Markov Random Field prior on the labels:

$$P(l) \propto \prod_{i=1}^{N} \prod_{j \in = \epsilon_i} \psi(i,j), \tag{B.2}$$

$$\psi(i,j) = \exp\left[\lambda\left(l_i l_j + (1-l_i)(1-l_j)\right)\right], \tag{B.3}$$

where λ determines the pair-wise interaction strength among neighbors and ϵ_i neighborhood of pixel i.

Given the Markov Random Fields prior and the likelihood model above, moving object detection in a given frame reduces to maximum a posterior $P(l|I)$ solution. According to the Bayes rule, the posterior is equivalent to:

$$\begin{aligned} P(l|z) &= \frac{P(z|l)P(l)}{P(I)} = \\ &= \frac{\prod_{i=1}^{N} P(z_i|l_i) \cdot \prod_{i=1}^{N} P(l_i) \cdot \exp\left[\sum_{i=1}^{N} \sum_{j \in Nb_i} \lambda(l_i l_j + (1-l_i)(1-l_j))\right]}{P(I)} = \\ &= \frac{\left(\prod_{i=1}^{N} P(z_i|l_i)\ P(l_i)\right) \cdot \exp\left[\sum_{i=1}^{N} \sum_{j \in Nb_i} \lambda(l_i l_j + (1-l_i)(1-l_j))\right]}{P(I)}, \end{aligned} \tag{B.4}$$

where $P(I)$ is the density of I which is a constant when I is given.

Finally, the MAP estimate is the binary image that maximizes the following Equation:

$$\begin{aligned} \arg\max_l P(I|l)P(l) &= \arg\min_l \left[-ln(P(I|l)P(l))\right] = \\ &= \arg\min_l \left[-ln(P(I|l)) - ln(P(l))\right], \end{aligned} \tag{B.5}$$

The discrete cost function (Equation B.6) leads to an standard form of the energy function that can be solved for global optimum using standard graph-cut algorithms [MJDW00]:

$$E(f) = E_{\text{data}}(f) + E_{\text{smooth}}(f) = \sum_{p \in P} D_i(f_p) + \lambda \sum_{\{p,q\} \in \psi} V_{p,q}(f_p, f_q), \tag{B.6}$$

where ψ is a defined neighborhood in pixels. $D_i(f_p)$ is a function derived from the observed data that measures the cost of assigning the label f_p to the pixel p (How appropriate a label is for the pixel). $V_{p,q}(f_p, f_q)$ measures the cost of assigning the labels f_p, f_q to the adjacent pixels p, q and is used to impose spatial smoothness. The role of λ is to balance the data $D_i(f_p)$ and smooth cost $V_{p,q}(f_p, f_q)$.

At the borders of objects, adjacent pixels should often have very different labels and it is important that E not over-penalize such labeling. This requires V to be a non-convex function of $|f_p - f_q|$. Such an energy function is called discontinuity-preserving.

Energy functions like E are extremely difficult to minimize, however, as they are non-convex functions in a space with many thousands of dimensions. They have traditionally been minimized with general-purpose optimization techniques (such as simulated annealing) that can minimize an arbitrary energy function. As a consequence of their generality, however, such techniques require exponential time and are extremely slow in practice. In the last few years, however, efficient algorithms have been developed for these problems based on graph cuts.

B.1 Graph Cuts

Suppose χ is a directed graph with non negative edge weights that has two special vertices (terminals), namely, the source s and the sink t. An $s-t$-cut (which we will refer to informally as a cut) $C = S;T$ is a partition of the vertices in Y into two disjoint sets S and T such that and $s \in S$ and $t \in T$. The cost of the cut is the sum of costs of all edges that go from S to T:

$$c(S,T) = \sum_{u \in S, v \in T, (u,v) \in \epsilon} c(u,v), \tag{B.7}$$

The minimum s-t-cut problem is to find a cut C with the smallest cost. Due to the theorem of [FF56], this is equivalent to compute the maximum flow from the source to sink. There are many algorithms that solve this problem in polynomial time with small constants.

It is convenient to note a cut $C = S, T$ by a labeling f mapping from the set of the vertices $Y - \{S, T\}$ to $\{0, 1\}$, where $f(v) = 0$ means that $v \in S$ and $f(v) = 1$ means that $v \in T$.

Note that a cut is a binary partition of a graph viewed as a labeling; it is a binary-valued labeling. While there are generalizations of the minimum $s-t$-cut problem that involve more than two terminals (such as the multi-way cut problem),

such generalizations are NP-hard.

B.2 Energy Minimization Via Graph Cuts

In order to minimize E using graph cuts, a specialized graph is created such that the minimum cut on the graph also minimizes E (either globally or locally). The form of the graph depends on the exact form of V and on the number of labels. In certain restricted situations, it is possible to efficiently compute the global minimum. This is also possible for an arbitrary number of labels as long as the labels are consecutive integers and V is the $L1$ distance.

However, a convex V is not discontinuity preserving and optimizing an energy function with such a V leads to over-smoothing at the borders of objects. The ability to find the global minimum efficiently, while theoretically of great value, does not overcome this drawback.

Moreover, efficient global energy minimization algorithms for even the simplest class of discontinuity-preserving energy functions almost certainly do not exist. Consider $V_{p,q}(f_p, f_q) = T[f_p \neq f_q]$, where the indicator function $T[\cdot]$ is 1 if its argument is true and otherwise 0. This smoothness term, sometimes called the Potts model, is clearly discontinuity-preserving.

However, graph cut algorithms have been developed that compute a local minimum in a strong sense. These methods minimize an energy function with non-binary variables by repeatedly minimizing an energy function with binary variables.

Bibliography

[AMYT00] S. Araki, T. Matsuoka, N. Yokoya, and H. Takemura. Real-time tracking of multiple moving object contours in a moving camera image sequence. *IEICE TRANSACTIONS on Information and Systems*, 83(7):1583–1591, 2000.

[AP09] Marcel Alcoverro and M Pardas. Voxel occupancy with viewing line inconsistency analysis and spatial regularization. *Int. Conf. on Computer Vision Theory and Applications*, pages 464–469, 2009.

[Bau74] Bruce Guenther Baumgart. Geometric modeling for computer vision. Technical report, Ph.D. thesis, CS Stanford University Document, 1974.

[Ber77] R. Beran. Minimum hellinger distance estimates for parametric models. *The Annals of Statistics*, 5(3):445–463, 1977.

[BL03] Andrea Bottino and Aldo Laurentini. Introducing a new problem: Shape-from-silhouette when the relative positions of the viewpoints is unknown. *IEEE Trans. on, Pattern Analysis and Machine Intelligence*, 25(11):1484–1493, 2003.

[BS10] Liefeng Bo and Cristian Sminchisescu. Twin gaussian processes for structured prediction. *Int. Journal of Computer Vision*, 87(1-2):28–52, 2010.

[BVZ01] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, 2001.

[BW09] A. Bleiweiss and M. Werman. Fusing time-of-flight depth and color for real-time segmentation and tracking. *Dynamic 3D Imaging*, pages 58–69, 2009.

[CFBM10] M. Cristani, M. Farenzena, D. Bloisi, and V. Murino. Background subtraction for automated multisensor surveillance: a comprehensive

review. *EURASIP Journal on Advances in Signal Processing*, 2010:43, 2010.

[CHB+05] TP Chen, H. Haussecker, A. Bovyrin, R. Belenov, K. Rodyushkin, A. Kuranov, and V. Eruhimov. Computer vision workload analysis: case study of video surveillance systems. *Intel Technology Journal*, 9(02):109–118, 2005.

[CKBH00] G.K.M. Cheung, T. Kanade, J.Y. Bouguet, and M. Holler. A real time system for robust 3D voxel reconstruction of human motions. In *Proc. of Int. Conf. of Computer Vision and Pattern Recognition*, page 2714, 2000.

[CPV04] R. Cucchiara, A. Prati, and R. Vezzani. Real-time motion segmentation from moving cameras. *Real-Time Imaging*, 10(3):127–143, 2004.

[CR03] D. Comaniciu and V. Ramesh. Real-time tracking of non-rigid objects using mean shift, July 8 2003. US Patent 6,590,999.

[CTPD08] R. Crabb, C. Tracey, A. Puranik, and J. Davis. Real-time foreground segmentation via range and color imaging. In *IEEE Workshop of Computer Vision and Pattern Recognition*, pages 1–5. IEEE, 2008.

[Day90] A.M. Day. The implementation of an algorithm to find the convex hull of a set of three-dimensional points. *ACM Trans. on Graphics*, 9(1):105–132, 1990.

[DLR+77] A.P. Dempster, N.M. Laird, D.B. Rubin, et al. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977.

[DMMM10] L. Díaz, R. Muñoz, FJ. Madrid, and R. Medina. Shape from silhouette using dempster–shafer theory. *Pattern Recognition*, 43(6):2119–2131, 2010.

[DR01] Tax D.M.J. and Duin R.P.W. Combining one-class classifiers. *Proc. workshop Multiple Combining Systems*, 2096:299–308, 2001.

[EBBN05] A. Erol, G. Bebis, R.D. Boyle, and M. Nicolescu. Visual hull construction using adaptive sampling. In *Proc. IEEE Workshop on Application of Computer Vision*, volume 1, pages 234–241. Citeseer, 2005.

[EHD00] A. Elgammal, D. Harwood, and L. Davis. Non-parametric model for background subtraction. *Proc. European Conf. on Computer Vision*, pages 751–767, 2000.

[FB03] J.S. Franco and E. Boyer. Exact polyhedral visual hulls. In *British Machine Vision Conference*, volume 1, pages 329–338. Citeseer, 2003.

[FB05] J.S. Franco and E. Boyer. Fusion of multi-view silhouette cues using a space occupancy grid. 2005.

[FF56] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8(3):399–404, 1956.

[FF87] G. Fasano and A. Franceschini. A multidimensional version of the Kolmogorov-Smirnov test. *Royal Astronomical Society, Monthly Notices (ISSN 0035-8711)*, 225:155–170, 1987.

[FFK11] A. Frick, M. Franke, and R. Koch. Time-consistent foreground segmentation of dynamic content from color and depth video. *Pattern Recognition*, pages 296–305, 2011.

[FLB07] J.S. Franco, M. Lapierre, and E. Boyer. Visual shapes of silhouette sets. In *IEEE Symposium on 3D Data Processing, Visualization, and Transmission*, pages 397–404, 2007.

[FS77] Preparata F.P. and Hong S.J. Convex hulls of finite sets of points in two and three dimensions. *Communications of the ACM*, 20(2):87–93, 1977.

[GFBP10] Li Guan, J-S Franco, Edmond Boyer, and Marc Pollefeys. Probabilistic 3d occupancy flow with latent silhouette cues. pages 1379–1386, 2010.

[GLA+08] S.A. Guomundsson, R. Larsen, H. Aanaes, M. Pardas, and J.R. Casas. Tof imaging in smart room environments towards improved people tracking. In *IEEE Workshop of Computer Vision and Pattern Recognition*, pages 1–6, 2008.

[GPH09] J. Gallego, M. Pardas, and G. Haro. Bayesian foreground segmentation and tracking using pixel-wise background model and region based foreground model. In *IEEE Proc. Int. Conference on Image Processing*, pages 3205–3208. IEEE, 2009.

[GPL08] J. Gallego, M. Pardas, and J.L. Landabaso. Segmentation and tracking of static and moving objects in video surveillance scenarios. *Proc. IEEE Int. Conf. on Image Processing*, pages 2716–2719, 2008.

[GPS89] DM Greig, BT Porteous, and Allan H Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989.

[GRBS10] Juergen Gall, Bodo Rosenhahn, Thomas Brox, and Hans-Peter Seidel. Optimization and filtering for human motion capture. *International Journal of Computer Vision*, 87(1-2):75–92, 2010.

[GT01] I. Grinias and G. Tziritas. A semi-automatic seeded region growing algorithm for video object localization and tracking. *Signal Processing: Image Communication*, 16(10):977–986, 2001.

[GVPG03] P.F. Gabriel, J.G. Verly, J.H. Piater, and A. Genon. The state of the art in multiple object tracking under occlusion in video sequences. *Advanced Concepts for Intelligent Vision Systems*, pages 166–173, 2003.

[HHCC03] J.W. Hsieh, W.F. Hu, C.J. Chang, and Y.S. Chen. Shadow elimination for effective moving object detection by Gaussian shadow modeling. *Image and Vision Computing*, 21(6):505–516, 2003.

[HHD99] T. Horprasert, D. Harwood, and L.S. Davis. A statistical approach for real-time robust background subtraction and shadow detection. In *Proc. IEEE Int. Conf. on Computer Vision Frame-Rate Workshop*, 1999.

[HHD02] I. Haritaoglu, D. Harwood, and L.S. Davis. W¡ sup¿ 4¡/sup¿: real-time surveillance of people and their activities. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 22(8):809–830, 2002.

[HZ03] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge Univ Pr, 2003.

[IB98] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *International journal of computer vision*, 29(1):5–28, 1998.

[INR] INRIA. 4D Repository. http://4drepository.inrialpes.fr/.

[JDWR00] S. Jabri, Z. Duric, H. Wechsler, and A. Rosenfeld. Detection and location of people in video images using adaptive fusion of color and edge information. *Proc. Int. Conf. on Pattern Recognition.*, 4:627 –630 vol.4, 2000.

[JTD+08] Y. Jin, L. Tao, H. Di, N.I. Rao, and G. Xu. Background modeling from a free-moving camera by multi-layer homography algorithm. In *IEEE Proc. Int. Conference on Image Processing*, pages 1572–1575, 2008.

[KBKL09] A. Kolb, E. Barth, R. Koch, and R. Larsen. Time-of-flight sensors in computer graphics. *Eurographics State of the Art Reports*, pages 119–134, 2009.

[KS00a] S. Khan and M. Shah. Tracking people in presence of occlusion. *Asian Conf. on Computer Vision*, 5, 2000.

[KS00b] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000.

[Kul87] S. Kullback. The kullback-leibler distance. *The American Statistician*, 41:340–341, 1987.

[Kun04] L.I. Kuncheva. *Combining Pattern Classifiers: Methods and Algorithms*. Wiley-Interscience, 2004.

[KWH+02] D. Koller, J. Weber, T. Huang, J. Malik, G. Ogasawara, B. Rao, and S. Russell. Towards robust automatic traffic scene analysis in real-time. In *Proc. Int. Conf. on Pattern Recognition, Computer Vision and Image Processing.*, volume 1, pages 126–131, 2002.

[KZ02] V. Kolmogorov and R. Zabih. Multi-camera scene reconstruction via graph cuts. *Proc. European Conference on Computer Vision*, pages 8–40, 2002.

[Lau91] Aldo Laurentini. The visual hull: A new tool for contour-based image understanding. *Proc. 7th. Scandinavian Conf. on Image Analysis*, pages 993–1002, 1991.

[Lau94] A. Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 150–162, 1994.

[Lau95] Aldo Laurentini. How far 3d shapes can be understood from 2d silhouettes. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995.

[LC87] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *ACM Siggraph Computer Graphics*, volume 21, pages 163–169. ACM, 1987.

[LE10] C-S Lee and A Elgammal. Coupled visual and kinematic manifold models for tracking. *Int. Journal of Computer Vision*, 87(1-2):118–139, 2010.

[LFP07] Svetlana Lazebnik, Yasutaka Furukawa, and Jean Ponce. Projective visual hulls. *International Journal of Computer Vision*, 74(2):137–165, 2007.

[LHGT04] L. Li, W. Huang, I.Y.H. Gu, and Q. Tian. Statistical modeling of complex backgrounds for foreground object detection. *Trans. IEEE on Image Processing*, 13(11):1459–1472, 2004.

[LLR08] I. Leichter, M. Lindenbaum, and E. Rivlin. Bittracker. A Bitmap Tracker for Visual Tracking under Very General Conditions. *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, 30(9):1572–1588, 2008.

[LP06a] JL Landabaso and M. Pardàs. Cooperative background modelling using multiple cameras towards human detection in smart-rooms. In *In Proc. of European Signal Processing Conference*, 2006.

[LP06b] José Luis Landabaso and Montse Pardàs. Foreground regions extraction and characterization towards real-time object tracking. In *Machine Learning for Multimodal Interaction*, pages 241–249. Springer, 2006.

[LPAM+09] JL Landabaso, JC Pujol-Alcolado, T Montserrat, et al. A global probabilistic framework for the foreground, background and shadow classification task. In *Proc. IEEE Int. Conf. on Image Processing*, 2009.

[LV02] BPL Lo and SA Velastin. Automatic congestion detection system for underground platforms. In *IEEE Int. Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158–161, 2002.

[MBM01] W. Matusik, C. Buehler, and L. McMillan. Polyhedral visual hulls for real-time rendering. In *Eurographics Workshop on Rendering*, volume 1, pages 115–125. Citeseer, 2001.

[MD03] A. Mittal and L.S. Davis. M 2 Tracker: a multi-view approach to segmenting and tracking people in a cluttered scene. *International Journal of Computer Vision*, 51(3):189–203, 2003.

[MG01] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding*, 81(3):231–268, 2001.

[MHK06] T.B. Moeslund, A. Hilton, and V. Kruger. A survey of advances in vision-based human motion capture and analysis. *Computer vision and image understanding*, 104(2-3):90–126, 2006.

[MJDW00] S.J. McKenna, S. Jabri, Z. Duric, and H. Wechsler. Tracking interacting people. *Int. Conf. on Face and Gesture Recognition*, 2000.

[MKKJ96] Saied Moezzi, Arun Katkere, Don Y Kuramura, and Ramesh Jain. Reality modeling and visualization from multiple video sequences. *Computer Graphics and Applications, IEEE*, 16(6):58–63, 1996.

[MR93] X.L. Meng and D.B. Rubin. Maximum likelihood estimation via the ECM algorithm: A general framework. *Biometrika*, 80(2):267, 1993.

[MRG99] S.J. McKenna, Y. Raja, and S. Gong. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4):225–231, 1999.

[MTG97] Saied Moezzi, Li-Cheng Tai, and Philippe Gerard. Virtual view generation for 3d digital video. *Multimedia, IEEE*, 4(1):18–26, 1997.

[Pal11] Montse Solano Pallarol. Seguimiento y egmentación de objetos en secuencias de vídeo para aplicaciones interactivas. *Telecommunications Engineering, Final Project*, 2011.

[Pic05] M. Piccardi. Background subtraction techniques: a review. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 4, pages 3099–3104. Ieee, 2005.

[PL04] P. Pinheiro and P. Lima. Bayesian sensor fusion for cooperative object localization and world modeling. In *Conference on Intelligent Autonomous Systems*. Citeseer, 2004.

[PMTC03] A. Prati, I. Mikic, MM Trivedi, and R. Cucchiara. Detecting moving shadows: Algorithms and evaluation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(7):918–923, 2003.

[PR03] Juszczak P. and Duin R.P.W. Uncertainty sampling methods for one-class classifiers. *Proc. of ICML Workshop on Learning from Imbalanced Data Sets*, 2003.

[PT03] F. Porikli and O. Tuzel. Human body tracking by adaptive background models and mean-shift analysis. In *IEEE Int. Workshop on Performance Evaluation of Tracking and Surveillance*, 2003.

[PT05] F. Porikli and J. Thornton. Shadow flow: A recursive method to learn moving cast shadows. *Proc. IEEE Int. Conf. on Computer Vision*, 1:891–898, 2005.

[SA99] S. Shah and JK Aggarwal. Statistical decision integration using fisher criterion. *Proc. Int. Conf. on Information Fusion*, pages 722–729, 1999.

[SA02] H.S. Sawhney and S. Ayer. Compact representations of videos through dominant and multiple motion estimation. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 18(8):814–830, 2002.

[SB02] SM Smith and JM Brady. ASSET-2: Real-time motion segmentation and shape tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17(8):814–820, 2002.

[SBB10] Leonid Sigal, Alexandru O Balan, and Michael J Black. Humaneva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion. *International journal of computer vision*, 87(1-2):4–27, 2010.

[SBF00] Hedvig Sidenbladh, Michael J Black, and David J Fleet. Stochastic tracking of 3d human figures using 2d image motion. In *Eur. Conf. on Computer Vision*, pages 702–718. Springer, 2000.

[SD97] Steven M Seitz and Charles R Dyer. Photorealistic scene reconstruction by voxel coloring. In *IEEE Proc. Computer Vision and Pattern Recognition,*, pages 1067–1073. IEEE, 1997.

[SG00] C. Stauffer and W.E.L. Grimson. Learning patterns of activity using real-time tracking. *IEEE Trans. on, Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000.

[SHG+11] Carsten Stoll, Nils Hasler, Juergen Gall, H Seidel, and Christian Theobalt. Fast articulated motion tracking using a sums of gaussians body model. In *Proc. IEEE Int. Conf. Computer Vision*, pages 951–958, 2011.

[SK11] I. Schiller and R. Koch. Improved video segmentation by adaptive combination of depth keying and mixture-of-gaussians. *Image Analysis*, pages 59–68, 2011.

[SM11] Jordi Salvador Marcos. Surface reconstruction for multi-view video. Technical report, Ph.D. thesis, Universitat Politècnica de Catalunya (UPC), 2011.

[SP05] Sudipta N Sinha and Marc Pollefeys. Multi-view reconstruction using photo-consistency and exact silhouette constraints: A maximum-flow formulation. In *IEEE Proc. Int. Conf. on Computer Vision,*, volume 1, pages 349–356, 2005.

[SS05] Y. Sheikh and M. Shah. Bayesian modeling of dynamic scenes for object detection. *IEEE trans. on Pattern Analysis and Machine Intelligence*, 27(11):1778–1792, 2005.

[SS11] E.E. Stone and M. Skubic. Evaluation of an inexpensive depth camera for passive in-home fall risk assessment. In *Proc. IEEE Conf. Pervasive Computing Technologies for Healthcare*, pages 71–77, 2011.

[SSC10] Jordi Salvador, Xavier Suau, and Josep R Casas. From silhouettes to 3d points to mesh: towards free viewpoint video. In *Proc. Int. workshop on 3D Video Processing*, pages 19–24. ACM, 2010.

[TEBM08] F. Tiburzi, M. Escudero, J. Bescós, and J.M. Martínez. A ground-truth for motion-based video-object segmentation. *IEEE Int. Conf. on Image Processing Workshop on Multimedia Information Retrieval: New Trends and Challenges*, pages 17–20, 2008.

[TM98] C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. *Proc. Int. Conf. on COmputer Vision*, pages 839–846, 1998.

[VM08] V. Vilaplana and F. Marques. Region-based mean shift tracking: application to face tracking. In *IEEE Int. Conf. on Image Processing*, pages 2712–2715. IEEE, 2008.

[Vo09] Sergio Velastin and others. Multicamera Human Action Video Data (MUHAVI). *Kingston University's Digital Imaging Research Centre. http://dipersec.king.ac.uk/MuHAVi-MAS/*, 2009.

[WADP02] C.R. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. on Pattern Analysis and Machine Intelligence,*, 19(7):780–785, 2002.

[WBB08] Q. Wu, P. Boulanger, and W.F. Bischof. Robust real-time bi-layer video segmentation using infrared video. In *Computer and Robot Vision, 2008. CRV'08. Canadian Conference on*, pages 87–94. IEEE, 2008.

[WZYZ10] L. Wang, C. Zhang, R. Yang, and C. Zhang. Tofcut: towards robust real-time foreground extraction using a time-of-flight camera. In *Proc. of 3DPVT*, 2010.

[XCA11] L. Xia, C.C. Chen, and JK Aggarwal. Human detection using depth information by kinect. In *Workshop on Human Activity Understanding from 3D Data in Conjunction with CVPR (HAU3D)*, 2011.

[XLL04] LQ Xu, JL Landabaso, and B. Lei. Segmentation and tracking of multiple moving objects for intelligent video analysis. *BT Technology Journal*, 22(3):140–150, 2004.

[XLP05] L.Q. Xu, JL Landabaso, and M. Pardas. Shadow removal with blob-based morphological reconstruction for error correction. In *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing.*, volume 2, 2005.

[YZC+07] T. Yu, C. Zhang, M. Cohen, Y. Rui, and Y. Wu. Monocular video foreground/background segmentation by tracking spatial-color Gaussian mixture models. In *IEEE Workshop on Motion and Video Computing*, pages 5–5, 2007.

[ZA01] Q. Zhou and J.K. Aggarwal. Tracking and classifying moving objects from video. *Proc. IEEE Workshop on Performance Evaluation of Tracking and Surveillance*, pages 52–59, 2001.