UNIVERSITAT POLITÈCNICA DE CATALUNYA

A Unified Framework for Consistent 2D/3D Foreground Object Detection

José Luis Landabaso Díaz

Advisor: Montse Pardàs i Feliu

Department of Signal Theory and Communications

Ph. D. Dissertation

Barcelona, December 2007

ACTA DE QUALIFICACIÓ DE LA TESI DOCTORAL

Reunit el tribunal integrat pels sota signants per jutjar la tesi doctoral:

Títol de la tesi: ....

Autor de la tesi: ....

Acorda atorgar la qualificació de:

□ No apte
□ Aprovat
□ Notable
□ Excel·lent
□ Excel·lent Cum Laude

Barcelona, …………… de/d’…................…………….. de ....….

El President

....
(nom i cognoms)

El Secretari

....
(nom i cognoms)

El vocal

....
(nom i cognoms)

El vocal

....
(nom i cognoms)

El vocal

....
(nom i cognoms)

A Cristina.

Abstract

This Ph.D. dissemination addresses two-dimensional (2D) and three-dimensional (3D) active entity detection in video scenes. Active entities are the foreground parts in a stationary background scene and they typically correspond to the regions of interest in many applications such as automated video surveillance, object and person tracking and suspicious object detection, among others.

This dissemination presents a novel framework that permits obtaining 2D and 3D active entities as an inter-dependent probabilistic procedure. In the process of creating this framework, a study has been conducted to explore ways to generalize existing activity detection techniques to a Bayesian form. Some of the techniques, specially those which were closer to planar foreground detection, can be usually extended. With regard to volumetric activity detection, the literature reveals that very little work has been done in the field of Bayesian classification. Thus, in order to support the framework previously outlined, a new Bayesian 3D activity detection technique has been developed.

A probabilistic analysis only accounts for half of the problem. The Bayesian framework gives a unified manner to interact between the planar and the volumetric detection tasks and helps to prevent the propagation of noisy pixel observations to the 3D space. However, when large systematic errors occur in the 2D detection level, a different approach has to be taken to correct them. In this respect, 2D/3D geometric relations can be exploited to detect systematic errors. Errors in the planar detection task often produce a set of incompatible foreground planar regions in the sense that they cannot be globally explained as the projection of the detected 3D volume. This is a key issue with significant implications that is not considered in most of current approaches. Classical 3D reconstruction methods simply assume that errors do not occur in the 2D plane. Instead, this dissemination presents a new 3D foreground detection scheme that is able to correct errors in 2D planar detections by checking the consistency between 3D foreground detections and the set of corresponding 2D foreground regions.

Resumen

Esta tesis aborda el problema de la deteccion bi-dimensional (2D) y tri-dimensional (3D) de entidades activas en escenas de vídeo. Las entidades activas son aquellos elementos del primer plano de una imagen, que generalmente se corresponden con las regiones de interés de muchas aplicaciones, como por ejemplo la vídeo vigilancia automática, el seguimiento de personas y objetos o la detección de objetos sospechosos.

En esta tesis se presenta un novedoso marco de trabajo que permite obtener entidades activas 2D y 3D de forma inter-dependiente y probabilística. Para ello se ha realizado un estudio con el objetivo de explorar posibles extensiones Bayesianas de los detectores de actividad. Las técnicas más cercanas a la detección 2D son las que se pueden extender con mayor facilidad. En cuanto a la detección volumétrica, la literatura revela que se ha relizado muy poco trabajo en el campo de la clasificación Bayesiana, y por tanto se ha desarrollado una nueva técnica probabilística de detección 3D, con el objetivo de integrarla en el marco de trabajo mencionado anteriormente.

Un análisis probabilístico sólo resuelve la mitad del problema. El marco de trabajo Bayesiano proporciona una manera unificada de interacción entre las técnicas de detección 2D y 3D, y ayuda a prevenir la propagación del ruido de observación en los píxeles al espacio tri-dimensional. Sin embargo, cuando ocurren errores más graves de detección 2D que son sistemáticos, el problema debe abordarse desde otro punto de vista: se pueden explotar ciertas relaciones geométricas entre los espacios 2D y 3D. Los errores en la detección bi-dimensional producen frecuentemente un conjunto de regiones 2D incompatibles, es decir, que dichas regiones no pueden ser justificadas mediante la proyección de un volumen 3D. Este es un aspecto crítico con implicaciones importantes que la mayoría de propuestas actuales no contemplan. Los métodos clásicos de reconstrucción 3D asumen simplemente que no ocurren errores en el nivel bi-dimensional. Por contra, esta tesis presenta un nuevo esquema de detección 3D que permite corregir errores en las imágenes mediante la comprobación de la consistencia entre las detecciones volumétricas y su correspondiente conjunto de regiones de primer plano bi-dimensionales.

Agradecimientos

En primer lugar, quiero agradecer a Montse Pardàs sus consejos y ayuda a lo largo de estos últimos cinco años. Gracias sobretodo por *perseguirme* en esta última etapa de la tesis.

Gracias también al grupo de imagen y en especial a Ferran Marqués por la revisión minuciosa que hizo de la tesis, a Gloria Haro por sus consejos y correciones en el capítulo 4, a Li-Qun Xu por introducirme en las técnicas relacionadas con los capítulos 4 y 5, a Josep Ramon Casas por su ayuda en el capítulo 7 y a Joel Solé por ayudarme a formalizar las matemáticas en varios capítulos. En general, gracias a todos los doctorandos por su ayuda y por los buenos momentos, incluyendo las tardes de fútbol.

Finalmente, agradezco a todos los que desde fuera de la universidad siempre han estado ahí; amigos, familia y especialmente a Cristina que ha tenido todo el cariño y toda la paciencia del mundo conmigo.

José Luis

Barcelona, Diciembre 2007

Contents

Notation

Boldface upper-case letters denote matrices, boldface lower-case letters denote vectors and lower-case italics denote scalars.

$\mathbb{R}$, $\mathbb{Z}$	The set of real and integer numbers, respectively.
$\mathbf{X}^T$	Transpose of the matrix **X**.
$\mathbf{X}^{-1}$	Inverse of the matrix **X**.
$[\mathbf{x}]_i$	$(i)^{\text{th}}$ component of vector **x**.
$\lvert x \rvert$	Absolute value of the scalar x.
$\lVert\mathbf{x}\rVert$	Euclidean norm of the vector **x**: $\lVert\mathbf{x}\rVert = \sqrt{\mathbf{x}^T\mathbf{x}}$.
arg	Argument.
max, min	Maximum and minimum.
$(\cdot)^\star$	Optimal value.
$\bigcap$, $\bigcup$	Intersection and union.
$A \oplus B$	Morphological dilation of an image A by structuring element B.
$A \subseteq B$	A is subset of B.
$[a,b]$, (a,b)	Closed interval ($a \le x \le b$) and open interval ($a < x < b$), respectively.
$P(\cdot)$	Probability.
$E_{\mathcal{Y}}[\cdot]$	Mathematical expectation with respect to unknown data $\mathcal{Y}$.
$\mathcal{L}(\cdot)$	Likelihood.
$\propto$	Equal up to a scaling factor (proportional).

$\simeq$	Approximately equal.
$\log(\cdot)$	Natural logarithm.
$\binom{n}{k}$	Binomial coefficient: “n choose k”.
$H(\cdot)$	Heaviside step function.
$\delta f[x]$	Central difference of function $f[x]$.

Acronyms

2D, 3D	2-Dimensional and 3-Dimensional, respectively.
BDR	Background Detection Records.
BPT	Binary Partition Tree.
CCA	Connected Components Analysis.
EM	Expectation Maximization.
FA	False Alarm.
e.g.	for example.
JPEG	Joint Photographic Experts Group (image standard).
LUT	Look-Up Table.
M	Miss.
MAP	Maximum A Posteriori.
ML	Maximum Likelihood.
MoG	Mixture of Gaussians.
MPEG	Moving Picture Experts Group.
OpenGL	Open Graphics Library (standard specification).
pdf	probability density function.
RGB	Red Green Blue color space.
SfIS	Shape from Inconsistent Silhouette.
SfS	Shape from Silhouette.
S&G	Chris Stauffer and W. Eric L. Grimson (authors of [SG00b]).
i.e.	that is to say.
IH	Inconsistent Hull.
PT	Projection Test.
UH	Unbiased Hull.

VH	Visual Hull.
w.r.t.	with respect to.
YUV	YUV color space: luma (Y) and color (U and V)

Chapter 1

Introduction

1.1 Motivation

Detecting active entities in video scenes has been one of the most studied topics in computer vision. Active entities are the foreground regions in a stationary background scene. They typically correspond to persons and objects that move over static elements of the environment. Thus, areas of activity concur with the regions of interest in many applications, such as automated video surveillance, object tracking, human behavior modeling, immersive video-conferencing, suspicious object detection, etc. The great number of applications has motivated a large number of research works and led to many significant achievements.

Two decades ago, the first active entity detectors operating with one fixed camera were developed. In the following decade, as computing hardware became more powerful, the interest started to shift towards volumetric active entity detection using multiple cameras. In recent years the growth of 3D detection applications has been particularly noticeable.

Several of the volumetric foreground detection techniques are built on top of planar foreground detectors. In practice, most volumetric detectors simply take planar detections as an input source, without considering which was the process that yielded the planar foreground regions. These systems have been used in recent years with great success. However, the dependency between the planar and volumetric approaches now needs to be understood more deeply and improved further in order to bridge the gap between both techniques. This thesis is focused towards this challenging field. In particular, this thesis addresses the problem of precise 3D active entity detection by analyzing the 2D-3D interaction process.

In this thesis we present a novel framework that permits obtaining 2D and 3D active entities as an inter-dependent procedure. In order to create this framework, we have taken a Bayesian approach that unifies our new findings with many of the most successful current approaches.

In the process of creating this framework, we have studied how existing techniques of relevance could be generalized to a Bayesian form. We have found that some of them, specially those which were closer to planar foreground detection, can be usually extended. Not surprisingly, providing probabilistic justification to *engineered* approaches significantly improves their performance. With regard to volumetric activity detection, the literature reveals that very little work has been done in the field of Bayesian classification. Thus, in order to support the framework previously outlined, we have developed a new Bayesian 3D activity detection technique that better exploits the redundancy present in a multi-camera environment.

A probabilistic analysis only accounts for half of the problem. The Bayesian framework proposed gives a unified manner to interact between the planar and the volumetric detection subsystems. In addition, an outlier model in the probabilistic framework prevents noisy pixel observations from propagating to the rest of the views through a bogus reconstruction. However, when large systematic errors occur in the 2D detection plane, then outlier models cannot help.

As it will be shown throughout this thesis, errors in the planar detection task often produce a set of incompatible foreground planar regions in the sense that they cannot be globally explained as the projection of the detected 3D volume. This is a problem with significant implications that is not considered in most of current approaches. Current 3D reconstruction methods simply assume that errors do not occur in the 2D plane. Instead, we present a new three-dimensional foreground detection scheme that is able to correct errors in 2D planar detections by checking the consistency between 3D foreground detections and the set of corresponding 2D foreground regions.

This consistency-aware reconstruction method is one of the main contributions of this thesis. The technique allows to obtain accurate 3D models that provide the most reasonable explanation of a 3D detection based on 2D observations. In addition, reassigned classifications are incorporated back into our Bayesian framework so that the 3D probabilistic map reflects them.

1.2 Contributions

The main contributions of this thesis are consequence of the cooperative, Bayesian, consistency-aware approach proposed. Some of the key contributions are:

- In the image level we have contributed a theoretical framework in which pixels are both classified and maintained along the time in a Bayesian way. Indeed, there already existed some approaches in which pixels where segmented based on probabilistic classification. However, note that classification is only a part of the problem. Moreover, the easy one. Pixel models also have to be maintained along the time as new pixel colors are observed along the time. In the proposed framework, the classification and update stages can be fully explained as a procedure in which one stage is probabilistically related to the other. We show that the speed of adaptation of a pixel model is a function of the foreground/background probability. Furthermore, we show that some of the most commonly used systems in the past can be adapted to the presented classification/update setting.

In addition, the presented scheme is able to estimate the error and classification probabilities of the system, which was not possible in many of the current approaches. The ability to provide classification and error probabilities is of great importance so that they can be employed as confidence values in higher-level systems, such as the 3D probabilistic reconstruction method described later in this text.

- In the volumetric level, we have presented a new activity detection method in which 2D foreground classifications in a view are achieved in accordance with the rest of views in a Bayesian framework. In this novel approach, the 3D space is simultaneously reconstructed and classified, in contrast with current approaches that first classify images and then reconstruct the volume. The interaction with image foreground detection is as follows: The speed of adaptation of a pixel model depends on the probability of its classification. Based on 2D probabilistic values, it is possible to obtain the foreground and background probabilities of any 3D point. The projection of the 3D probabilistic points is finally used as the new speed of adaptation. Thus, we assure that pixel models are maintained making use of multi-camera information.
- The main contribution of this thesis is in the area of consistency-aware 3D reconstruction. The most distinguishing aspect of the presented method is that it assesses the coherence of the volumetric and the planar foreground regions before classifying. Current techniques reconstruct only the part of the volume that projects consistently in all the planar foreground regions. In contrast, we propose a fast technique for estimating that part of the volume that projects inconsistently and define a criteria for classifying it by minimizing the probability of 3D misclassification.
- Finally, we show that it is possible to integrate the consistency-aware reconstruction method back into our Bayesian framework giving results for video sequences that are over other state-of-the-art techniques.

A complete list of the contributions of this thesis, as well as the references to journal, book chapter, patents, conference proceedings, contributions to projects and publications in the course of preparation have been compiled in the last chapter.

1.3 Thesis Organization

This thesis is structured as follows. Next chapter is devoted to review the state-of-the-art of both planar and volumetric foreground detection techniques. Chapter 3 states the problem we want to solve, which can only be fully understood after the initial review of current techniques and their flaws. Chapter 4 presents our Bayesian framework for the planar foreground detection task. In the chapter, we present the classification problem as a maximum a posteriori task between foreground and background modeled classes and show how some of the current most successful approaches benefit from being adapted to our framework. In addition, we derive the equations for on-line model update with probabilistic justification. These equations will be constantly revisited through the rest of the text. In chapter 5, several applications making use of planar activity detections are presented showing the usefulness of such techniques. Chapter 6 is devoted to a theoretical and experimental study of a Bayesian volumetric foreground detection technique. The chapter bridges the gap with the planar probabilistic method in Chapter 4 since it provides more-robust 3D probabilistic values for updating planar models. Chapter 7 presents another contribution of this thesis. In this chapter, we first describe a fast technique for estimating that part of the volume which projects inconsistently and propose a criteria for classifying it by minimizing the probability of 3D misclassification. In addition, we describe the interaction between this technique and the cooperative Bayesian method presented in Chapter 6. Finally, Chapter 8 provides some conclusions and proposes future extensions to this work.

Chapter 2

The Foreground Detection Challenge

THIS CHAPTER is devoted to discuss the two major groups in which the foreground classification task can be divided, depending on the dimensional space where the detection is being done. The first group corresponds to the set of approaches in which the detection of the moving entities (objects or persons) is attained in two dimensions, usually employing a single camera. The second group corresponds to the techniques that obtain a foreground volume in the three-dimensional space, normally using several cameras.

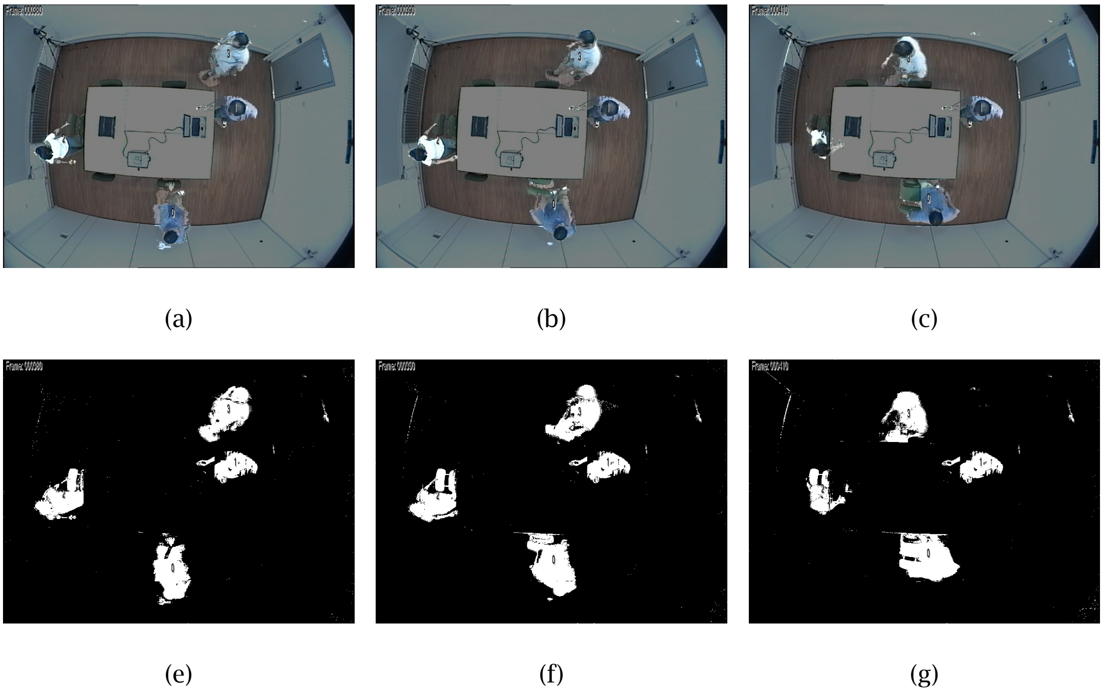
Each type of foreground detection has its own approaches, problems and applications. An overview on current trends of foreground extraction is given in this chapter.

2.1 Planar Foreground Detection

Planar foreground detection is the first group of foreground extraction techniques which are presented. Two-dimensional foreground detections can be used in many computer vision applications, such as video surveillance, 2D-trackers (see Figure 2.1), human-computer interaction, object oriented encoding as in MPEG-4, suspicious object detection and to obtain volumetric foreground representations using a technique known as Shape from Silhouette (which will be deeply discussed in the text). Some applications based on planar segmentation are presented in chapter 5.

There are many different approaches for the planar foreground segmentation process described in the literature. These techniques can be grouped in two main types:

- In the first approach, 2D regions are first segmented based on the feature homogeneity of the pixels, such as color or texture, and tracked later [CSE05, ML98, TA02]. Usually in $t+1$ a segmentation is done and new regions are matched with those which were segmented



(a) (b) (c)

(e) (f) (g)

Figure 2.1: Images obtained from the zenithal camera of the Smart Room of our lab at the UPC showing a set of labeled blobs, i.e., grouped pixels, in frames 380, 390 and 410, corresponding to a sequence recorded at 25fps.

in t. In this approach, all the moving regions are considered to be the foreground regions of a scene.

- Another approach 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, either as an exception to the modeled background [EDHD99, HHD99, HHD00, MJD$^+$00, SG00b, WADP97], or in a Bayesian framework, using MAP (*maximum a posteriori*) classification, if there exist foreground models [KS00a, LHGT04, MD02].

  In a temporal perspective, these methods were originally referred as Background subtraction techniques, since the first methods detected the moving objects by subtracting the current image from a background (also called reference) image. The concept of Background *subtraction* was gradually changed to that one of Background *modeling*, where the subtraction step is replaced by a classification step using a Background model.

The latter is the common approach taken when working with stationary cameras as in video surveillance applications or smart-room environments, which are the scenarios of interest in which this thesis is focused on. Since the cameras are stationary, the pixel features of the background process can be statistically modeled so that the fore/background classification is tolerant to slow changes of illumination, the cameras thermal noise, shadows, etc. Later, the foreground regions which are extracted can also be modeled to help during the classification process. A good review of the state-of-the art on Background modeling can be found at [Pic04].

One of the most successful approaches of background modeling was introduced by Stauffer and Grimson (S&G) [ISBG99, SG00b]. The algorithm, which has been extensively implemented and referenced in the literature, is successful for a combination of factors, including implementation easiness, fast operation, robustness to slow and sudden illumination changes, and for being able to model repetitive background changes, such as in waving flags or moving trees. In the following, we give a brief overview of the algorithm, detailing which are the common steps shared with other similar works, and pointing out which are the main differences w.r.t. other approaches.

2.1.1 Background Modeling towards Foreground Segmentation

The main idea of the algorithm of S&G is to model the photometric variations ($\mathbf{I}$) of each pixel along the time course by a mixture of K Gaussian distributions. Modeling the color or gray-scale information is the most common approach taken in the literature [EDHD99, FR97, HHD00, JDWR00, SG00b, WADP97]. It is also not uncommon to model texture-based information [JS02, LL02, LPX05b, XLL04], and the temporal information [LHGT02, Wix00] associated with interframe changes at the pixels.

In the method of S&G, different Gaussians are assumed to characterize different color appearances in each pixel, and each Gaussian is weighted (w) depending on how often the Gaussian has explained the same appearance. Mixtures of Gaussians (MoG) have been used in the literature [FR97, HHD00, SG00b] to ensure that repetitive moving background can be represented by different probabilistic functions.

However, a single Gaussian is enough to model the background process [JDWR00, WADP97] when the scene of interest consists of a relatively static situation. Gaussians have also been used as the kernels of nonparametric models in [EDHD99], for instance. In this approach, a kernel-based function is employed to represent the color distribution of each pixel. The background pdf is then given as a sum of Gaussian kernels centered in the most recent background values. Thus, the method can be understood as a smoothed continuous version of the histogram. However, note that this is different from an MoG, since in this method each Gaussian describes just a single sample of data instead of a mode of the pdf.

Background models are usually expressed as the likelihood $p(\mathbf{I_x}|\beta)$ that an observation $\mathbf{I_x}$ in a pixel $\mathbf{x}$ (indicating its spatial coordinates) belongs to the background process (β). The mathematical expressions of the likelihoods of the mentioned Gaussian-based models are somewhat similar among them; and the computational complexity rises as the number of Gaussians is increased. Thence, a single Gaussian model ($\mathrm{G}_\mathbf{x}(\mathbf{I_x})$) is faster to operate than an MoG model ($\mathrm{MoG}_\mathbf{x}(\mathbf{I_x})$), which is in turn faster than a nonparametric model with a Gaussian kernel ($\mathrm{nonPrm}_\mathbf{x}(\mathbf{I_x})$).

If we choose to model the background process of each pixel as a single Gaussian distribution, the likelihood of observing a certain value $\mathbf{I_x}$ in pixel $\mathbf{x}$ is

$$\mathrm{G}_\mathbf{x}(\mathbf{I_x}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_\mathbf{x}|}} e^{-\frac{1}{2}(\mathbf{I_x}-\mu_\mathbf{x})^T \Sigma_\mathbf{x}^{-1}(\mathbf{I_x}-\mu_\mathbf{x})}, \tag{2.1}$$

where $\mu_\mathbf{x}$ and $\Sigma_\mathbf{x}$ are the mean vector and covariance matrix, respectively, of the Gaussian corresponding to pixel $\mathbf{x}$, and where D denotes the number of features considered in vector $\mathbf{I_x}$.

The background model using an MoG distribution is

$$\begin{aligned}\mathrm{MoG}_\mathbf{x}(\mathbf{I_x}) &= \sum_{k=1}^{K} w_{\mathbf{x},k}\mathrm{G}_{\mathbf{x},k}(\mathbf{I_x}) = \\ &\sum_{k=1}^{K} \frac{w_{\mathbf{x},k}}{(2\pi)^{D/2}\sqrt{|\Sigma_{\mathbf{x},k}|}} e^{-\frac{1}{2}(\mathbf{I_x}-\mu_{\mathbf{x},k})^T \Sigma_{\mathbf{x},k}^{-1}(\mathbf{I_x}-\mu_{\mathbf{x},k})},\end{aligned} \tag{2.2}$$

where K is the total number of Gaussians used in each pixel, and where $w_{\mathbf{x},k}$ is the prior probability that a background pixel is represented by a certain mode k of the mixture ($\sum_{k=1}^{K} w_{\mathbf{x},k} = 1$). These priors are often referred as the weights of the Gaussians. Also note that the means and covariances are indexed w.r.t. a Gaussian k of the MoG in $\mathbf{x}$: $\Sigma_{\mathbf{x},k}$ and $\mu_{\mathbf{x},k}$.

And finally, the probability density function, considering the nonparametric model is

$$\begin{aligned}\mathrm{nonPrm}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) =& \frac{1}{N}\sum_{n=1}^{N}\mathbf{K}_{\Sigma}(\mathbf{I}_{\mathbf{x}} - \mathbf{I}_{\mathbf{x}}[n]) = \\ & \frac{1}{N}\sum_{n=1}^{N}\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma|}}e^{-\frac{1}{2}(\mathbf{I}_{\mathbf{x}}-\mathbf{I}_{\mathbf{x}}[n])^{T}\Sigma^{-1}(\mathbf{I}_{\mathbf{x}}-\mathbf{I}_{\mathbf{x}}[n])},\end{aligned} \tag{2.3}$$

where $\mathbf{K}_{\Sigma}$ is a Gaussian kernel function with bandwidth Σ; and where $\mathbf{I}_{\mathbf{x}}[n]$ denotes the value of pixel $\mathbf{x}$ in one of the N samples observed in the past.

In (2.1), (2.2) and (2.3), $\mathbf{I}_{\mathbf{x}}$ is a vector of features that may include spectral, spatial or temporal information, among others. If this is the case, it is important to keep the covariance matrices as they are since the nature of the modeled features is very different. However, if $\mathbf{I}_{\mathbf{x}}$ is used to represent the red, green and blue, or YUV values of the pixel, then it is safe to assume that all the channels are statistically independent, leaving

$$\Sigma_{\mathbf{x}} = \begin{pmatrix} [\sigma^2_{\mathbf{x}}]_1 & 0 & 0 \\ 0 & [\sigma^2_{\mathbf{x}}]_2 & 0 \\ 0 & 0 & [\sigma^2_{\mathbf{x}}]_3 \end{pmatrix} \tag{2.4}$$

$$\Sigma_{\mathbf{x},k} = \begin{pmatrix} [\sigma^2_{\mathbf{x},k}]_1 & 0 & 0 \\ 0 & [\sigma^2_{\mathbf{x},k}]_2 & 0 \\ 0 & 0 & [\sigma^2_{\mathbf{x},k}]_3 \end{pmatrix} \tag{2.5}$$

$$\Sigma = \begin{pmatrix} [\sigma^2]_1 & 0 & 0 \\ 0 & [\sigma^2]_2 & 0 \\ 0 & 0 & [\sigma^2]_3 \end{pmatrix}. \tag{2.6}$$

And then,

$$\mathrm{G}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \prod_{d=1}^{D}\frac{1}{\sqrt{2\pi[\sigma^2_{\mathbf{x}}]_d}}e^{-\frac{1}{2[\sigma^2_{\mathbf{x}}]_d}([\mathbf{I}_{\mathbf{x}}]_d-[\mu_{\mathbf{x}}]_d)^2}, \tag{2.7}$$

which is equivalent to consider D unidimensional Gaussians, being D equal to 3 in the RGB and YUV color spaces.

Expressions of $\mathrm{MoG}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}})$ and $\mathrm{nonPrm}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}})$ assuming channel independence are,

$$\mathrm{MoG}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \sum_{k=1}^{K}\left(w_{\mathbf{x},k}\prod_{d=1}^{D}\frac{1}{\sqrt{2\pi[\sigma^2_{\mathbf{x},k}]_d}}e^{-\frac{1}{2[\sigma^2_{\mathbf{x},k}]_d}([\mathbf{I}_{\mathbf{x}}]_d-[\mu_{\mathbf{x},k}]_d)^2}\right) \tag{2.8}$$

$$\mathrm{nonPrm}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \frac{1}{N}\sum_{i=1}^{N}\prod_{d=1}^{D}\frac{1}{\sqrt{2\pi[\sigma^2]_d}}e^{-\frac{1}{2[\sigma^2]_d}([\mathbf{I}_{\mathbf{x}}]_d-[\mathbf{I}_{\mathbf{x}}[n]]_d)^2}, \tag{2.9}$$

where the kernel bandwidth $[\sigma^2]_d$ for a given pixel and channel can be estimated by computing the median absolute deviation over a sample for consecutive intensity values of the pixel. That is, the median m of $|\mathbf{I}_{\mathbf{x}}[t] - \mathbf{I}_{\mathbf{x}}[t-1]|$ for each consecutive pair $(\mathbf{I}_{\mathbf{x}}[t], \mathbf{I}_{\mathbf{x}}[t-1])$ in the sample is calculated independently for each color channel [EDHD99].

Furthermore, in the RGB color-space it is also common to consider that all the channels have the same variances leaving $\Sigma_{\mathbf{x}} = \sigma_{\mathbf{x}}^2 \begin{pmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{pmatrix}$, $\Sigma_{\mathbf{x},k} = \sigma_{\mathbf{x},k}^2 \begin{pmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{pmatrix}$ and $\Sigma = \sigma^2 \begin{pmatrix} 1&0&0 \\ 0&1&0 \\ 0&0&1 \end{pmatrix}$, and therefore:

$$\mathrm{G}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{\mathbf{x}}^2}} e^{-\frac{1}{2\sigma_{\mathbf{x}}^2}([\mathbf{I}_{\mathbf{x}}]_d - [\mu_{\mathbf{x}}]_d)^2} \tag{2.10}$$

$$\mathrm{MoG}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \sum_{k=1}^{K} \left(w_{\mathbf{x},k} \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma_{\mathbf{x},k}^2}} e^{-\frac{1}{2\sigma_{\mathbf{x},k}^2}([\mathbf{I}_{\mathbf{x}}]_d - [\mu_{\mathbf{x},k}]_d)^2} \right), \tag{2.11}$$

which is the approach taken by S&G.

Finally, the expression of the nonparametric model assuming channel independence and equal variances across channels is

$$\mathrm{nonPrm}_{\mathbf{x}}(\mathbf{I}_{\mathbf{x}}) = \frac{1}{N} \sum_{i=1}^{N} \prod_{d=1}^{D} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{1}{2\sigma^2}([\mathbf{I}_{\mathbf{x}}]_d - [\mathbf{I}_{\mathbf{x}}[n]]_d)^2}. \tag{2.12}$$

Even though Gaussian-based background modeling is the most common approach taken in the literature, other models have also been successfully used including eigenvalue decompositions of image blocks [SWFS03]. Decompositions of whole images have also been proposed in [ORP00]. And finally, there have been simpler attempts with reasonable results, such as computing the median over the last n frames [CGPP03, LV00].

Independently of the modeled features and model type, the Background modeling process in all the most relevant approaches is accomplished in the two following general steps:

Pixel Classification: The main objective of any foreground detection method is to classify a pixel value into the Background or Foreground process. A correct classification is important not only for being the output result of the algorithm but also because background models are updated based on the classification decision. There are two main types of classifiers. (1) Pixels can be classified into foreground as an exception to the background model [EDHD99, HHD00, MJD+00, SG00b], or (2) as the result of a Bayesian MAP classification [KS00a, LHGT04, MD02] when there exist models of the foreground entities.

Model Update: Once the pixels are classified, Background models are updated. There are two main ways of updating the models. (1) Model parameters can be simply maintained by incorporating new elements of information at every instant [SG00b, WADP97], or (2) can be completely recomputed [EDHD99, LV00] at each instant considering the last n samples.

Hereafter, we outline the approaches taken by different authors to accomplish the classification/update steps. The techniques used by S&G for the classification and update steps are presented and compared to the most common approaches taken by other authors. Later, we devise an extension of the planar foreground detection process for multi-camera scenarios, showing the improvements over single-camera systems and expressing the conditions under where a planar foreground can be extracted using a set of cameras.

2.1.1.1 Classification Step

Similarly as in [WADP97], in the approach of S&G a pixel $\mathbf{x}$ is assigned to a Gaussian k when the pixel's color value ($\mathbf{I_x}$) is within 2.5 standard deviations of the distribution mean ($\mu_{\mathbf{x},k}$). Then, in order to determine which Gaussians represent the foreground or background processes, the Gaussians of each pixel are reordered according to $\frac{w_{\mathbf{x},k}}{\sigma_{\mathbf{x},k}}$ in descending order. The first few Gaussians in the list correspond to the ones with more supporting evidence (more times explaining incoming pixels) at the lowest variance (explained incoming pixels are always very similar). In other words, the first few Gaussians represent the background process since the background is often very static (low variance ($\sigma_{\mathbf{x},k}$)) and it is seen during most of the time (high weight ($w_{\mathbf{x},k}$)). On the contrary, unassigned pixel values and the pixel values corresponding to the last Gaussians of the list are classified into the foreground. Formally, in the method of S&G a pixel is classified into the background process if its value matches one of the first B distributions decided by (2.13); and it is classified into the foreground, otherwise.

$$B = \underset{b}{\operatorname{argmin}}\left(\sum_{k=1}^{b} w_{\mathbf{x},k} > T\right), \tag{2.13}$$

where k expresses the index of the Gaussians in the reordering ($\frac{w_{\mathbf{x},k}}{\sigma_{\mathbf{x},k}}$) previously mentioned.

The process of new Gaussian modes creation is as follows. If none of the distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

The S&G approach is specially appealing for the very fast, simple and efficient classification process explained above, which works remarkably well despite it does not have a Bayesian justification.

In single-Gaussian model systems, often it is also considered that a pixel value corresponds to the foreground process if it does not fall within 2.5 standard deviations of the mean of the distribution, which serves to unequivocally model the background. Since there is only one Gaussian, the reordering does not have to be performed and the method is even faster than [SG00b]. And in systems using nonparametric models, a threshold is set on the likelihood of the background model to decide whether a pixel value belongs to the foreground or background process. In this case, although not explicitly mentioned, one could assume that a uniform foreground model is used. However, this assumption should be considered with caution. Note that it is important to guarantee that the integral of the uniform distribution over all the possible observable values remains one. Thus, just setting a threshold does not fully have a Bayesian interpretation either. A more detailed description of the classification task making use of uniform models will be given in §4.2.2.

Bayesian Classifiers A Bayesian approach for foreground segmentation is important because it provides a natural classification framework supported on probabilistic models. However, in the S&G and similar approaches, a Bayesian formulation is not possible since the foreground process is not modeled or it is only partially modeled. For instance, in the method of S&G all incoming pixels which are not clustered with any Gaussian are classified into the foreground even though there does not exist a model for them.

If there only exists a complete model of the background class, then the foreground segmentation task is a problem of one-class classification [JD03, TD01], which is harder than a standard two-class classification problem. In two-class classification, a decision boundary is supported from both sides by models of each of the classes (foreground and background). Because in case of one-class classification only the target Background class is available, just one side of the boundary is supported. Based on the model of one class only, it is hard to decide how tight the boundary ($2.5\sigma_{\mathbf{x},k}$ in MoG and $2.5\sigma_{\mathbf{x}}$ in single-Gaussian approaches) should fit around the target class.

The absence of foreground models makes it also very hard to estimate the classification error. The expected rate of false alarms, corresponding to the pixel values that are wrongly classified as foreground, can be estimated on the available data. However, a second kind of error, referring to the foreground pixels that are erroneously classified as background, cannot be estimated.

A Bayesian classifier is also useful because it does not only perform the classification task, but it also provides the probability that a pixel sample belongs to the chosen class. Pixel probability of belonging to a certain class is an important source of information for combining several pixel classifiers to obtain a Bayesian 3D classification, as will be shown in chapter 6.

In order to introduce the more general Bayesian classification approach with the previously mentioned advantages, both the foreground and background models (likelihoods) of a pixel have to be available. Then, the probability that a pixel $\mathbf{x}$ belongs to the foreground class (ϕ), given an observation $\mathbf{I}_{\mathbf{x}}$, can be expressed in terms of the likelihoods of the foreground and background processes as follows:

$$P(\phi|\mathbf{I}_{\mathbf{x}}) = \frac{P(\phi)p(\mathbf{I}_{\mathbf{x}}|\phi)}{p(\mathbf{I}_{\mathbf{x}})}. \tag{2.14}$$

In order to compute (2.14), the unconditional joint probability density ($p(\mathbf{I}_{\mathbf{x}})$) can be expressed in terms of the conditional distributions as:

$$p(\mathbf{I}_{\mathbf{x}}) = P(\phi)p(\mathbf{I}_{\mathbf{x}}|\phi) + P(\beta)p(\mathbf{I}_{\mathbf{x}}|\beta), \tag{2.15}$$

where $P(\phi)$ and $P(\beta)$[1] are prior probabilities of foreground and background, respectively. Then, it follows that

$$P(\phi|\mathbf{I}_{\mathbf{x}}) = \frac{P(\phi)p(\mathbf{I}_{\mathbf{x}}|\phi)}{P(\phi)p(\mathbf{I}_{\mathbf{x}}|\phi) + P(\beta)p(\mathbf{I}_{\mathbf{x}}|\beta)}. \tag{2.16}$$

[1] Foreground and background priors depend on the application. However, approximate values 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.

Thus, a pixel is classified into the foreground class with a Bayesian justification if $P(\phi|\mathbf{I}_\mathbf{x}) > \frac{1}{2}$ is satisfied.

Alternatively, the following test can also be used:

$$P(\phi)p(\mathbf{I}_\mathbf{x}|\phi) > P(\beta)p(\mathbf{I}_\mathbf{x}|\beta), \tag{2.17}$$

which is faster, since the denominator in (2.14) does not have to be computed.

As it has been previously mentioned, Bayesian classification [KS00a, LHGT04, MD02] can only be performed when there exists 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.

Several foreground models have been proposed in the past [EDHD99, HHD00, KS00a, LHGT04, MD02, MJD$^+$00, MRG99] for different purposes including the mentioned foreground segmentation task [KS00a, LHGT04, MD02] and also in object and person trackers where the foreground has been previously segmented [EDHD99, HHD00, MJD$^+$00, MRG99]. If foreground models are modeled as $p(\mathbf{I}_\mathbf{x}|\phi)$, then expression (2.16) can be used straight away. However, when foreground models are designed to represent the likelihood of a certain pixel to belong to a certain foreground entity (e): $p(\mathbf{I}_\mathbf{x}|e)$, then the likelihood that the pixel belongs to the foreground process is obtained by summing up all possible cases $p(\mathbf{I}_\mathbf{x}|\phi) = \sum_e P(e)p(\mathbf{I}_\mathbf{x}|e)$, where $P(e)$ expresses the prior probability that a certain foreground entity is observed.

Similarly as with background models, foreground models are Gaussian-based in most of the cases. For instance, single-Gaussians have been used in [WADP97], MoGs have been used in [KS00a, MJD$^+$00, MRG99] and nonparametric models with Gaussian kernels, in [EDHD99, MD02]. 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. In chapter 4, we propose a Bayesian classification method that can be used with any background model that is expressed as a likelihood function such as (2.1), (2.2) and (2.3), proposed in [WADP97], [SG00b] and [EDHD99], respectively. In addition, in §4.4 we develop the equations that link pixel classification with model update in a Bayesian way. This contrasts with current approaches, that usually link classification and update stages using heuristic rules, as we following show.

2.1.1.2 Update Step

Similarly as in the last section, in the following we review the model update mechanism employed in the approach of S&G, comparing it with some of the other most relevant approaches taken so far.

In [SG00b], every time a new value for pixel $\mathbf{x}$ is observed, the weight ($w_{\mathbf{x},k}$) of the k-th Gaussian that explains this observation is updated as in (2.18):

$$w_{\mathbf{x},k}[t] = \begin{cases} w_{\mathbf{x},k}[t-1] + \alpha(1 - w_{\mathbf{x},k}[t-1]); & \text{if matched} \\ (1-\alpha) w_{\mathbf{x},k}[t-1]; & \text{if not matched.} \end{cases} \tag{2.18}$$

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 variance ($\sigma^2_{\mathbf{x},k}$) and mean ($\mu_{\mathbf{x},k}$) associated to each Gaussian k are also renewed as in (2.19):

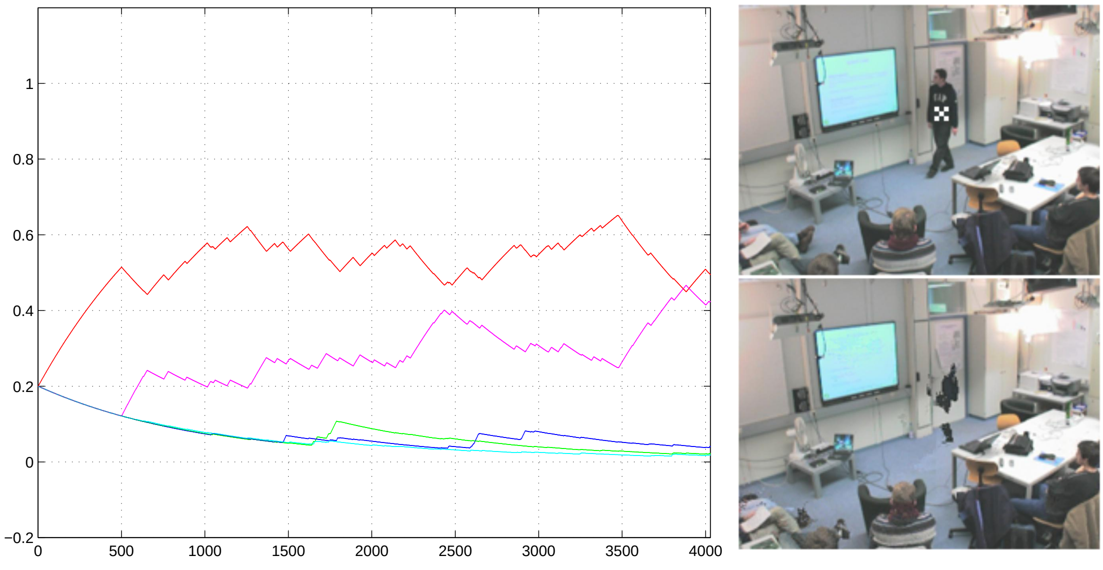
$$\begin{aligned} \mu_{\mathbf{x},k}[t] &= (1-\rho_{\mathbf{x},k})\mu_{\mathbf{x},k}[t-1] + \rho_{\mathbf{x},k}\mathbf{I}_{\mathbf{x}}[t] \\ \sigma^2_{\mathbf{x},k}[t] &= (1-\rho_{\mathbf{x},k})\sigma^2_{\mathbf{x},k}[t-1] + \rho_{\mathbf{x},k}\left(\mathbf{I}_{\mathbf{x}}[t] - \mu_{\mathbf{x},k}[t-1]\right)^T \left(\mathbf{I}_{\mathbf{x}}[t] - \mu_{\mathbf{x},k}[t-1]\right), \end{aligned} \tag{2.19}$$

where $\rho_{\mathbf{x},k}$ is the adaptation learning rate used in Gaussian k and pixel $\mathbf{x}$: $\rho_{\mathbf{x},k} \propto G_{\mathbf{x},k}(\mathbf{I}_{\mathbf{x}})$. Here again, the same type of low-pass filter as the mentioned above is used. In addition, means and variances can be updated faster when their likelihood $G_{\mathbf{x},k}(\mathbf{I}_{\mathbf{x}})$ is higher, or using a constant $\rho_{\mathbf{x},k} = \rho$, if it is important to reduce computation to provide faster Gaussian tracking. Even though the scheme has proved to be very robust, the equations above do not have strict justification. In addition, they do not provide the confidence in classification, but the likelihood of an observation.

By updating the mean and the variance, the system is allowed to adapt to slow illumination changes. Sudden changes are treated as follows: Every time when none of the K distributions matches the current pixel value ($\mathbf{I}_{\mathbf{x}}$), distribution l, with the lowest likelihood, is replaced with a new distribution with mean $\mu_{\mathbf{x},l} = \mathbf{I}_{\mathbf{x}}$, low weight ($w_{\mathbf{x},l}$) and high variance ($\sigma^2_{\mathbf{x},k}$). Then, new Gaussians are repeatedly updated to finally model a new background mode. A more in-depth study of the issues related with initialization of MoGs have been discussed in [KB01].

In the original formulation of the method, every time that a Gaussian k is clustered, its weight ($w_{\mathbf{x},k}$) is increased, assuming that the foreground entities will not remain static in the scene. Thence, the entities which have been stopped for a while are eventually integrated into the background. 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.

Figure 2.2 depicts an example of foreground-to-background integration. In Figure 2.2(a), the temporal evolution of the weights of a mixture of five Gaussians is shown and the pictures in Figure 2.2(b) show the input image and its background reference counterpart in a certain instant. Note that the background has wrongly adopted some parts of the person since the person has been standing at the same place for a long time. A white cross in Figure 2.2(b) marks the pixel for which the evolution of the weights is shown. The red line on the top of the graph shows the evolution of the weight of the Gaussian modeling the real background, while the pink line below corresponds to the weight of the Gaussian which was initially representing the foreground process. Note that after a certain number of frames, the weights tend not to be reliable enough to distinguish the real background from the foreground. In chapter 4, we propose some modifications to the algorithm to adjust it to different working environments.



(a)

(b)

Figure 2.2: Merging into the background process.

The S&G method has proved to work remarkably well in many typical situations, showing outstanding results in outdoors scenarios. However, other practitioners have taken different directions to adapt background models to illumination changes and to any other kind of change in the scene. For instance, there are other methods for online background model update, such as Kalman filtering [KWH$^+$94].

On the contrary, background models can be entirely recomputed at each instant, making use of the last number of observations. For instance, the median over the last N frames is computed in [CGPP03, LV00]. In [EDHD99], Elgammal et al. also recompute background models using the last observed samples. In this case, the background process is modeled making use of the nonparametric pdf in (2.9). In addition, they also provide the equations for an online adaptation scheme.

Also, given a certain number of observations, a standard method consists in maximizing the likelihood of the observed background:

$$p\left(\mathbf{I}_{\mathbf{x}}[1], \mathbf{I}_{\mathbf{x}}[2], \cdots, \mathbf{I}_{\mathbf{x}}[N] | \beta\right) \tag{2.20}$$

To do so, if we consider that the background process is stationary, it is possible to employ the standard expectation maximization method [Bil98, DLR77]. Wayne and Schoones [PS02] propose the mentioned expectation maximization method to recompute the parameters of an MoG model. In §4.4, we take a similar approach. The main difference is that we propose to use expectation maximization with the complete pixel likelihood, i.e., the likelihood of both the foreground and background processes: $p(\mathbf{I}_\mathbf{x}[1], \mathbf{I}_\mathbf{x}[2], \cdots, \mathbf{I}_\mathbf{x}[N])$. As we will show, this has some advantages over the former approach. First, the equations that we derive let us express the update speed of the parameters of the background models as a function of the probability of the background class. Note that this is not the case when (2.20) is maximized. Besides, if (2.20) is used, then the observations have to be pre-classified (*true* or *false*) before performing expectation maximization. In contrast, in our setting both the classification and update stages are inextricably tied one to the other. In addition, we also develop an online version of the equations so that a window with the last N frames does not have to be stored.

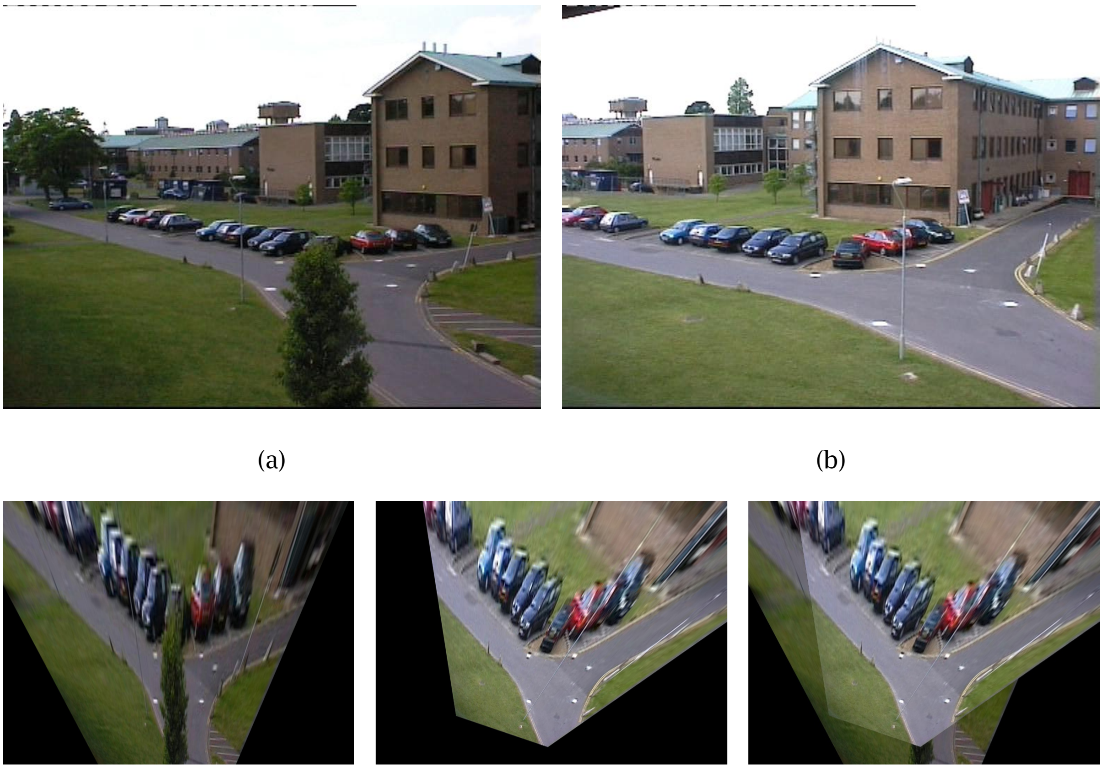
In general, online adaptation offers faster operation and avoids having to store a window of observed samples. However, the mechanisms by which the models are updated have to be carefully designed for each particular application so that the model's speed of adaptation is the optimal one based on the expectation that new observed values ($\mathbf{I}_\mathbf{x}$) form part of the background process. As we have mentioned, in chapter 4 we derive a set of equations where the speed of adaptation of the background models depends on the probability of the background process. Thus, higher-level features, such as the redundancy present in a 3D environments, can be also used if they are expressed as well in a Bayesian form, not having to rely only on the decisions taken at the pixel level.

2.1.2 Multi-Camera Planar Foreground Detection

The planar foreground segmentation process has been traditionally seen as a problem to be solved using a single camera. However, foreground segmentation making use of a number of widely separated cameras has two key advantages over conventional systems. First, the process gains robustness for not having to rely only on one camera. And second, the occlusions present in some views are automatically resolved when the objects do not occlude in other cameras.

Multi-camera planar foreground segmentation can only be achieved under the assumption that the scene develops in flat areas with large distances between the scene of interest and the cameras. In such situations, it is possible to project regions from one camera view into the others to allow a smooth transition between the images [BER02, KCM03, Ste99]. Thus, the foreground segmentation process becomes only a problem of 2D localization in a plane built as a mosaic of regions which provide low noise or lack of occluding objects. See, for instance, the tree in Figure 2.3(a), not present in Figure 2.3(b), therefore leaving this view free of occlusions.

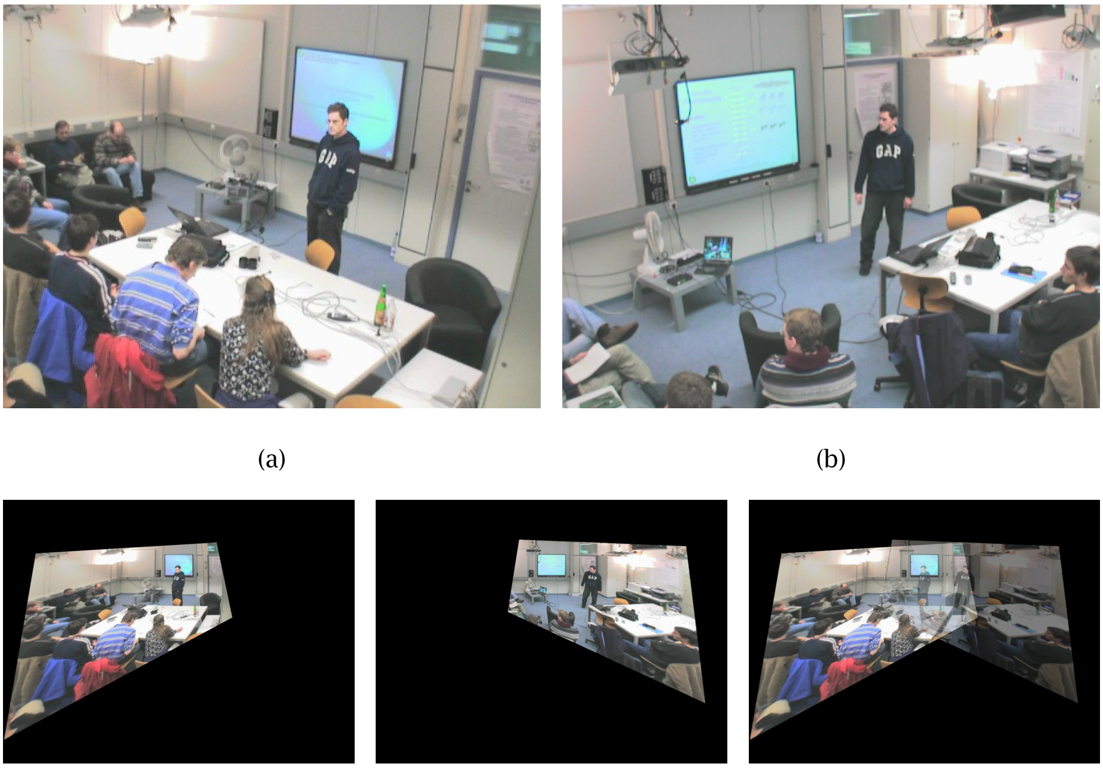
Unfortunately, as it has been mentioned, this technique only results when the foreground segmentation is performed in a plane. For example, in Figure 2.4, the three-dimensional structure of the area of interest arises the problems of using perspective projections.



(a) (b)

(c) (d) (fusion of projected images)

Figure 2.3: Homography in an outdoor scenario. Figures (a) and (b) show the scene from two different cameras. Their projection is done so that the 4 marked points positioned over the road match. The figure in the bottom-right shows the result of choosing the most visible areas of homographies (c) and (d).



(a)

(b)

(c)

(d)

(fusion of projected images)

Figure 2.4: Homography in an indoor scenario, where the cubic structure of the room makes it a difficult task to match and join the perspectives coming from different views.

The technique used to project one camera view into another is known as “homographic transformation between views" [HZ04, chapter 4],[Che02]. Formally, a homography is a planar projective transformation in which three colinear 2D-points lie in the same line after mapping them. Alternatively, a homography can be defined as an invertible mapping of lines to lines, which can be represented in homogenous coordinates[1] as follows:

$$\begin{pmatrix} x' \\ y' \\ 1' \end{pmatrix} = \begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & h_{33} \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix},$$

or more compactly

$$\mathbf{x}' = \mathbf{H}\mathbf{x}. \tag{2.21}$$

Although there are 9 entries in **H**, homographies have 8 degrees of freedom since the common scale factor is not relevant (remind that **x** and **x**′ are expressed in homogeneous coordinates). Thus, to determine the homography we only require 4 pairs of corresponding points. For instance, in Figure 2.3, we have used the 4 marked points over the road, and in Figure 2.4, the 4 corners of the whiteboard.

In practice, correspondence between points through a homography is known up to a noise factor, so many more points are used, forming an over constrained system which is solved by the minimization of a cost function.

Foreground detection using homographic transformations between multiple images has proved to be more robust than the systems using only one camera. In a multi-camera environment the data is redundant and this translates into better foreground segmentation performance, since the cameras with best visibility can be chosen at any moment. However, homographies cannot be used when the assumptions of large distances and flat areas do not hold, such as in the indoor scenario in Figure 2.4.

In order to exploit the redundancy present in a multi-camera system without the mentioned assumption, it is possible to perform a foreground detection process directly at the volume-level. In the next section we present a survey on the different approaches which can be taken for the volumetric foreground detection.

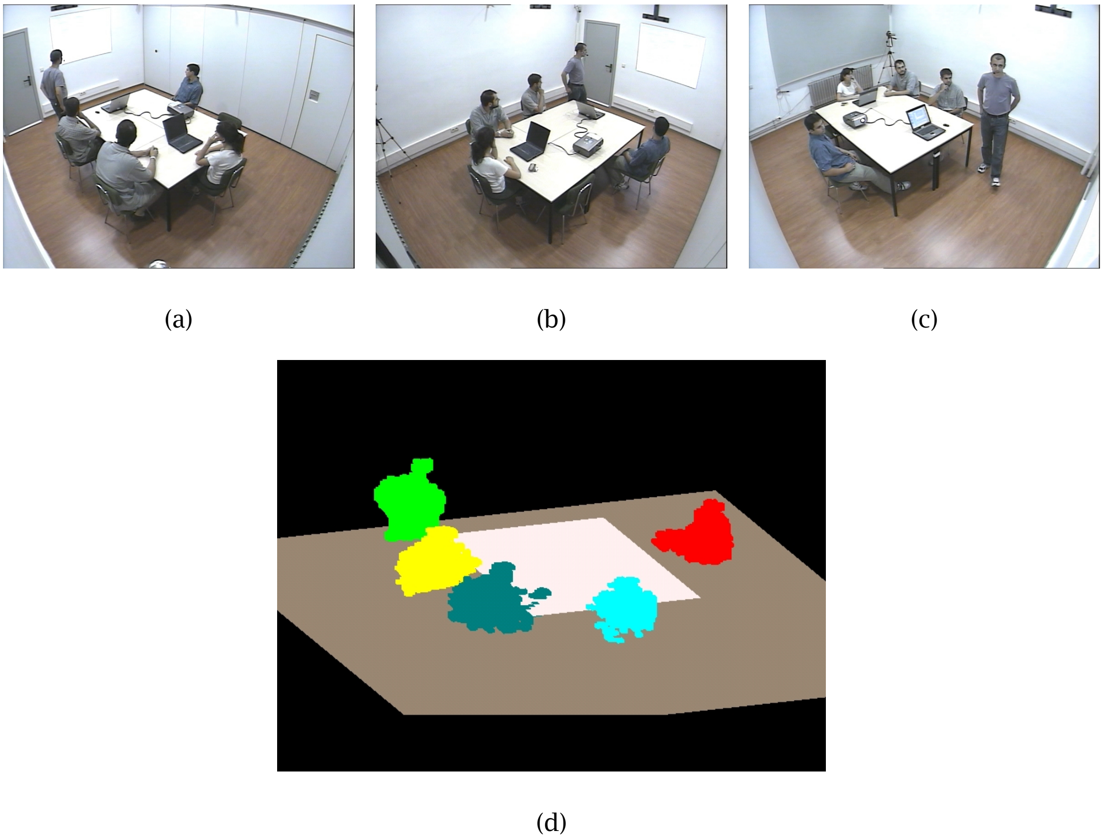
2.2 Volumetric Foreground Detection

The volumetric foreground detection task consists in the segmentation of the three-dimensional regions which do not form part of the background process in the scene of interest. 3D foreground segmentations can be used in many computer applications, such as video surveillance,

[1] Homogeneous coordinates (X, Y, W) of a finite point (x, y) in the plane are any three numbers for which $x = \frac{X}{W}$ and $y = \frac{Y}{W}$. Coordinates $(X, Y, 0)$, describe the point at infinity in the direction of $\frac{Y}{X}$. Thence, parallel lines described by **u** and **v** are found to intersect $(\mathbf{u} \times \mathbf{v})$ at a point in the infinity in the direction $\frac{[\mathbf{u}]_2}{[\mathbf{u}]_1} = \frac{[\mathbf{v}]_2}{[\mathbf{v}]_1}$.

3D trackers (see Figure 2.5), augmented reality (see Figure 2.6), body model fitting (see Figure 2.7), human-computer interaction, and to improve the performance of 2D-foreground detectors by using the techniques which we have developed and that are presented in chapters 6 and 7.

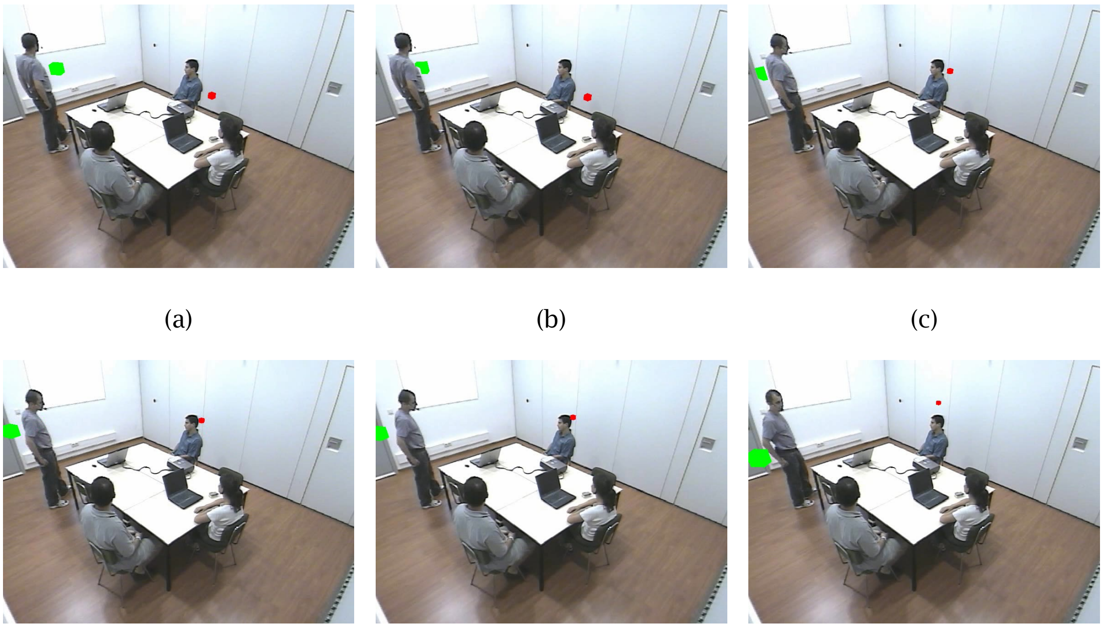
The 3D foreground detection process can be understood as an extension of the planar foreground detection process described in the previous sections. However, the three-dimensional detection process is significantly different from the two-dimensional one in that the cameras are only able to indirectly measure the 3D space. That is, a camera can only indirectly register the volume as its observed projection in the camera's optical plane, and the volumetric estimate can only be obtained after a procedure of triangulation.


(a) (b) (c)

(d)

Figure 2.5: Volumetric reconstruction and labeling of a video sequence recorded at our lab in the UPC.

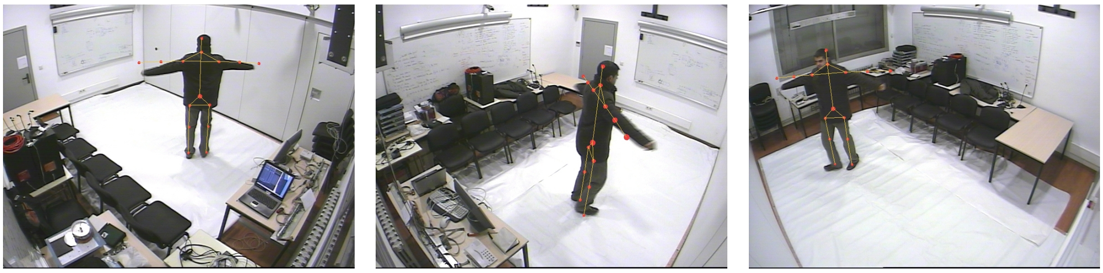
In the following, we introduce the 3D-2D correspondence problem: we first describe the camera model which is assumed in the rest of the dissertation, and then we review the camera calibration method which has been used. Later on, we explore some of the different existing approaches to obtain 3D estimates of the object of interest. Finally, we focus on the Shape from Silhouette approach and its varieties.



(a) (b) (c)

(d) (e) (f)

Figure 2.6: A simple example of augmented reality.
The images have been obtained at the Smart Room of our lab at the UPC during a presentation. The pictures correspond to frames 80, 83, 94, 97, 97, 99 and 125, of a sequence recorded at 25fps. In the scene there are two fictitious objects (a red and a green box) interacting with the persons in the room. Since the location of the entities in the 3D foreground is known, the fictitious objects can be hidden when being occluded. For example, the green box is continuously revolving around the presenter, but it is only shown when it is not behind him. The other fictitious object (the red box) makes use of the location of the foreground entities (the attendees) to move from one to the other.



(a) (b) (c)

Figure 2.7: An example showing a three-dimensional hierarchical structural model of the human body with soft kinematic constraints. These constraints take the form of a priori, stochastic distributions learned from previous configurations of the body exhibited during specific activities [CFCTP05, DBT03].

2.2.1 Pinhole Camera Model

To obtain a volume estimate of the real entity, the correspondences between the 3D points **X** and the 2D pixel coordinates $\mathbf{x_i}$ in each camera view i have to be resolved.

It is possible to model the 2D-3D correspondence task as follows:

$$\mathbf{x_i} = \mathbf{P_i X}. \tag{2.22}$$

In this section, we give a very brief overview of what is $\mathbf{P_i}$ in (2.22), and in the next section we describe the process to obtain it. We advice the reader to consult [Fau93, HZ04] for further details. Another brief and comprehensive review can be found in [Gar04].

A camera can be seen as a mechanism which performs the projection from the 3D real world to the 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 with CCD cameras.

From Figure 2.8, we can express the model in homogenous coordinates ($u = \frac{U}{S}$, $v = \frac{V}{S}$) as

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.23}$$

where f is the focal length of the camera.

The model may be generalized if the image coordinates are not centered at the intersection

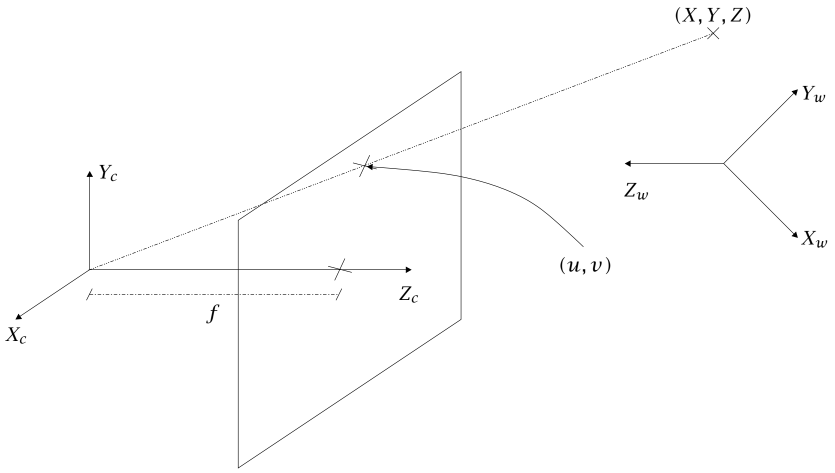


Figure 2.8: Coordinate systems used in the pinhole camera model.
The distance from the focal point to the center of the optical plane is the focal length f. The scene point (X, Y, X), is mapped to the point (u, v) in the optical plane.

of the optical axis with the retinal plane, and if the scaling of each axis is different:

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \underbrace{\begin{pmatrix} fm_u & 0 & u_0 \\ 0 & fm_v & v_0 \\ 0 & 0 & 1 \end{pmatrix}}_{\mathbf{K}} \left.\begin{matrix} 0 \\ 0 \\ 0 \end{matrix}\right) \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.24}$$

where m_u, m_v are the scaling factors of the focal length in each dimension, and u_0 and v_0 are offsets in each dimension. The matrix ($\mathbf{K}$) containing these parameters is known as the calibration matrix of a camera.

If we consider that the optical center of the camera is not the center of the real world coordinate system, then

$$\begin{pmatrix} U \\ V \\ S \end{pmatrix} = \underbrace{\begin{pmatrix} fm_u & 0 & u_0 \\ 0 & fm_v & v_0 \\ 0 & 0 & 1 \end{pmatrix} (\mathbf{R}|\mathbf{t})}_{\mathbf{P}} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix}, \tag{2.25}$$

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

At this point, a perfect projective transformation is assumed in the camera but no real lens works this way, and so we must introduce a non linear distortion model. Typically, only the radial distortion is considered.

The radial distortion model is expressed with the following equation:

$$\frac{r_d}{r} = \frac{u_d - u_0}{u - u_0} = \frac{v_d - v_0}{v - v_0}, \tag{2.26}$$

where (u_d, v_d) are the coordinates of a distorted image point.

Since the Taylor series expansion of (2.26) w.r.t. 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 of (2.26) and the pixel coordinates of the distorted image can be obtained as:

$$\begin{aligned} u_d =& u + (u - u_0)\left(k_1\left(\left(\frac{u-u_0}{fm_u}\right)^2 + \left(\frac{u-u_0}{fm_u}\right)^2\right) + k_2\left(\left(\frac{u-u_0}{fm_u}\right)^2 + \left(\frac{u-u_0}{fm_u}\right)^2\right)^2\right) \\ v_d =& v + (v - v_0)\left(k_1\left(\left(\frac{v-v_0}{fm_v}\right)^2 + \left(\frac{v-v_0}{fm_v}\right)^2\right) + k_2\left(\left(\frac{v-v_0}{fm_v}\right)^2 + \left(\frac{v-v_0}{fm_v}\right)^2\right)^2 .\right) \end{aligned}$$

2.2.2 Camera Calibration

The process of calibration consists in estimating the intrinsic $(\mathbf{K}, k_1, k_2)$ and extrinsic $(\mathbf{R}, \mathbf{t})$ parameters of a camera.

Since **P** has 12 entries, and (ignoring scale) only eleven degrees of freedom in homogeneous coordinates, it is necessary to have 11 equations (i.e., 11 3D/2D pairs of points) to solve **P**. In practice, more points are used, as in the estimation of a homography, to minimize a function of the error [HZ04, page 179]. All these calibration points may be obtained using special calibration devices, such as a chessboard, for instance (see Figure 2.9).

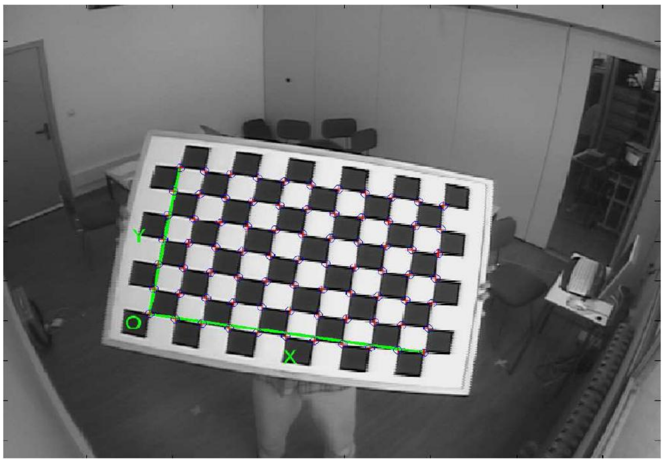


Figure 2.9: Calibration process of the cameras of the smart-room of our lab using a chessboard as the pattern of reference.

In order to complete the calibration process, the parameters defining the radial distortion also have to be estimated. To do so, k_1 and k_2 (assuming a Taylor expansion up to second order) can be included as part of the imaging process. The parameters are then computed together with **P** during an iterative minimization of the error cost function.

The calibration process of the smart-room of our lab has been performed using the software package available in [Bou].

2.2.3 3D Reconstruction Methods

In the following, we outline some of the approaches that can be taken for the 3D foreground segmentation task. We give special emphasis to the Shape from Silhouette approach, that is one of the most successful methods being currently used. Then, we introduce the main ideas that have lead some researchers to use this method instead of others. A more detailed description of the Shape from Silhouette method and its different varieties will be given later in section 2.2.4.

Before an in-depth examination of the different 3D reconstruction methods follows, a preliminary consideration on the working environment appears to be useful at this point. In all the different approaches, 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. Since this condition is satisfied in many of the typical camera setups, in the rest of the dissertation the condition is also assumed to be true.

Under the assumption expressed above, we following describe some of the approaches taken in the literature. The techniques for the volumetric foreground detection task can be sorted into four different groups based on the accuracy of the detected 3D-Shape.

From least to most accurate, the volumetric estimates are:

3D Bounding Box This is the simplest volume estimate of the real object. It is defined by the six faces delimiting the smallest rectangular hexahedron where the object is guaranteed to lie. The 3D Bounding Box is usually deduced from the silhouettes' 2D bounding boxes. The volume estimate that is obtained is only a rough approximation of the real shape. However, the method is fast, and therefore it is suitable as the initialization step of other more accurate reconstruction methods. For instance, 3D Bounding Boxes have been used as the first step of 3D reconstruction in [ES02, SY99], among others.

Convex Hull 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 [PH77] and another proposal taking care of the technical aspects of a practical implementation can be found in [Day90].

Visual Hull A more refined object estimate is the Visual Hull [BL03, Lau91, Lau94, Lau95]. The Visual Hull is obtained with the previously mentioned technique known as Shape from Silhouette:

1. In a first step, a number of images are taken from different positions around the scene of interest.
2. Later, each image is segmented to produce binary masks, also called silhouettes, to delimit the objects of interest in each view.
3. Finally, the volume estimate known as Visual Hull is obtained as the maximal volume which could explain the observed silhouettes.

Photo Hull The Photo Hull, which lies between the Visual Hull and the real object surface, is the most accurate non-invasive estimate of the real object. If a set of color or gray-scale images are available, it is possible to reconstruct the 3D scene model as a minimization process between real images and synthesized images from an hypothesized 3D scene. The process is performed as a photo-consistency test of visible volumetric points w.r.t. each image. Note that there may be many 3D scenes which are consistent with the real images, and therefore the 3D scene is not unique. The Photo Hull is defined as the maximum volume that is photo-consistent [KS00b], and Space Carving is the method used to obtain it.

An entity's shape $\mathcal{S}$, its Photo Hull $PH(\mathcal{S})$, its Visual Hull $VH(\mathcal{S})$, its Convex Hull $CH(\mathcal{S})$ and its Bounding Box $BB(\mathcal{S})$ are such that:

$$\mathcal{S} \subseteq PH(\mathcal{S}) \subseteq VH(\mathcal{S}) \subseteq CH(\mathcal{S}) \subseteq BB(\mathcal{S}), \tag{2.27}$$

assuming that all the different volume estimates are error-free.

From (2.27), it follows that Shape from Silhouette and Space Carving are the most precise techniques, being Space Carving the method that obtains the most accurate volumetric estimate of the shape $\mathcal{S}$. Indeed, color information gives the clue which makes Space Carving more precise than the other methods. For example, concave patches are not observable in the binary silhouettes, while they can be deduced using color consistency information. However, the choice of one method or another depends on the intended application. Space Carving obtains the most accurate volume estimate, but it is not suitable for real-time operation, for instance. Some of the motivations that have lead many researchers to choose Shape from Silhouette are:

First of all, because Shape from Silhouette performs faster than Space Carving. In fact, real-time operation of Shape from Silhouette systems is very common since it is algorithmically simpler than Space Carving. Notice that in Space Carving one needs to control the reflection properties of the objects. This makes the “color consistency check" criterion to be more complex than the “projection test" criterion, which is the measure used in the Shape from Silhouette approach.

Second, because the Visual Hull is a very accurate estimate of the entities' real shapes, even though it is not as accurate as the Photo Hull. The shapes obtained are precise enough so that, for instance, body and gesture analysis and 3D tracking can be performed.

And finally, one of the most important reasons for choosing Shape from Silhouette over other reconstruction methods is that the technique is intimately tied to the Planar Foreground Detection process introduced in section 2.1. The Visual Hull is built making use of 2D foreground detections. Furthermore, in an ideal error-free situation, the projection of the Visual Hull concurs with the two-dimensional detected foreground regions. We will show that this property can be exploited to improve the overall 2D and 3D detection process.

For further reference to the reconstruction problem in general and how to choose one or another approach, refer to [SCMS01] and [Dye01].

2.2.4 Shape from Silhouette (SfS)

In this section we describe the Shape from Silhouette (SfS) 3D reconstruction process in detail. We comment on the different varieties of SfS, show some examples and we finally explain some of its limitations.

SfS operates over the binary silhouette images, where the pixels have been already classified. Each point in the 2D foreground defines a ray in the 3D space that intersects the foreground entity 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.

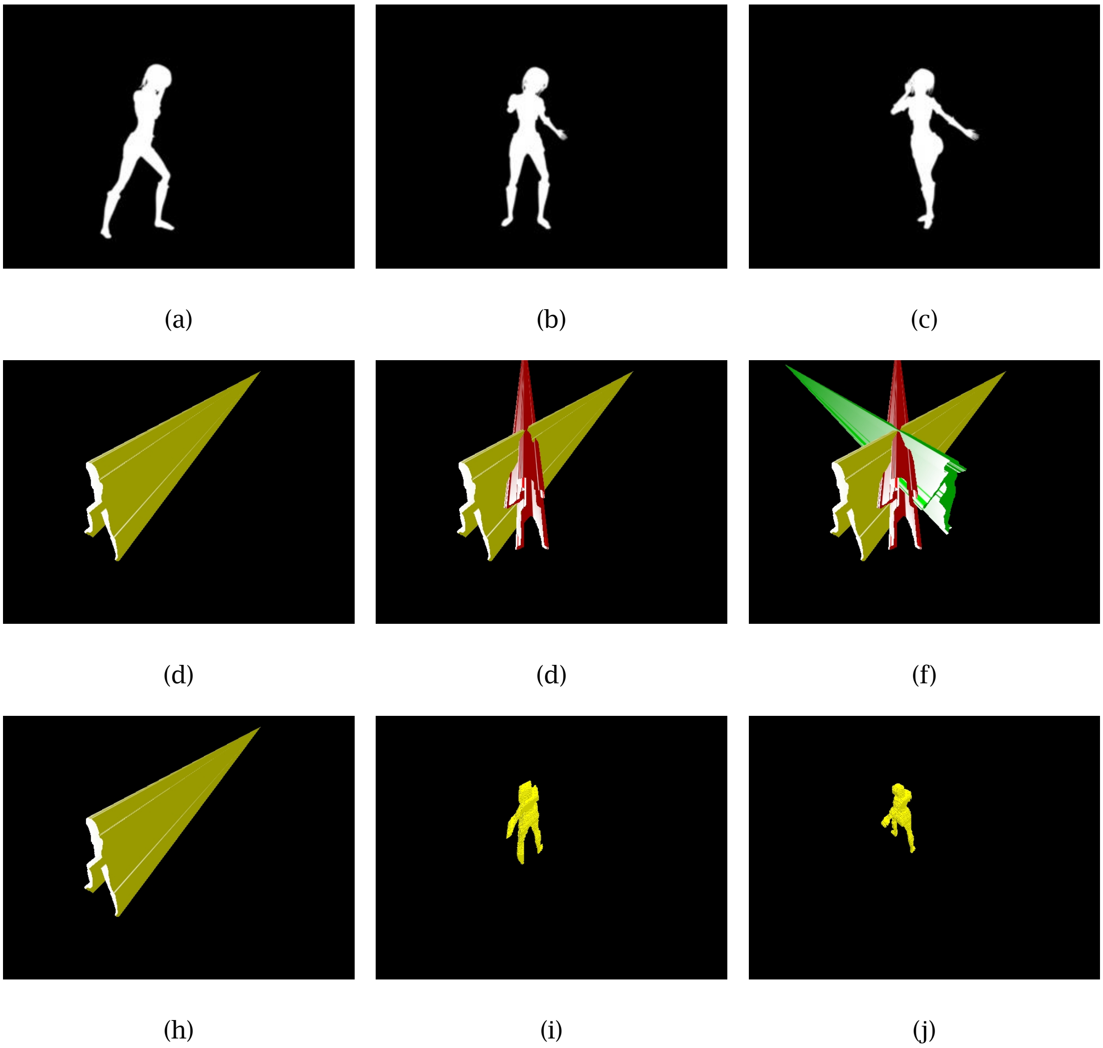
See Figure 2.10 for an example. In the first row, a set of silhouettes is shown. The second row of images depicts the visual cones using one, two and three cameras, from left to right, respectively.

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. The row at the bottom in Figure 2.10 shows the Visual Hull, i.e., the intersected volume, when using one, two and three cameras. Indeed, as the number of cameras increases, the reconstructed shape is more accurate. This effect can be more clearly appreciated by confronting the reconstructed Visual Hulls shown in Figure 2.10 with the more accurate one in Figure 2.11, where eight cameras have been used.

The principal step of the SfS algorithm is the intersection test. There are two main approaches to obtain the intersected volume: (1) the *geometric* approach and (2) the *voxel-based* approach.

2.2.4.1 Geometric Solutions

In the *geometric* approach, the silhouettes are back-projected, creating an explicit set of cones that are then geometrically intersected [GHF86, MBM01, MBR$^+$00, RS97].



(a) (b) (c)

(d) (d) (f)

(h) (i) (j)

Figure 2.10: The visual cone of a silhouette is projected from the camera's center. The first row of images shows three silhouettes of the Kung Fu Girl dataset. The *Kung-Fu Girl* dataset is provided by the *Graphics Optics Vision group* of *Max-Planck-Institut fur Informatik* [Gra]. The second row of images shows the projection of the visual cones. In the row at the bottom, the Visual Hull, that is the intersection of the visual cones, is presented: (h) shows the Visual Hull considering only one visual cone; and (i) and (j) show the resulting volume after the intersection of two and three visual cones, respectively.

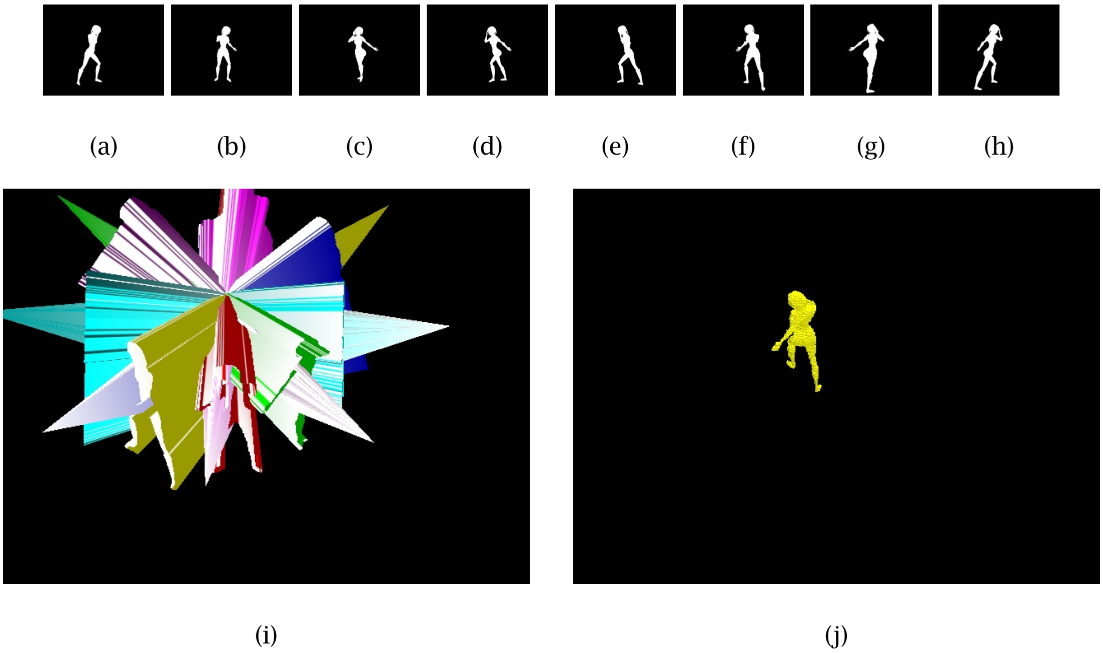


Figure 2.11: With eight views, the visual cones intersect further constraining the shape estimates of the Kung Fu Girl character. From (a) to (h) the silhouettes are shown. (i) depicts the visual cones, and in (h) the intersection of the cones is shown. Note that the resulting volume is more accurate than the one in Figure 2.10.

The geometric approach of the volume intersection problem can be summarized as follows. First, the silhouettes are decomposed into a series of connected edges. Then, for each input silhouette and for each edge, the face of the visual cone is computed. This face is then intersected with the cones of all other input silhouettes. The result of these intersections is a set of polygons that define the surface of the Visual Hull.

Geometric methods describe the intersected volume by a series of 3D Constructive Solid Geometry *CSG* [MBM01, MBR$^+$00] intersections. In these systems, the geometric computations are performed in the image space, eliminating the resampling and quantification artifacts of the voxel-based approaches. However, real-time operation [MBM01, MBR$^+$00] can only be achieved when working with low resolution images. In addition, the silhouettes need to be simplified with coarser polygonal approximations. Therefore, these methods shift the quantization problem from 3D to 2D.

2.2.4.2 Voxel-Based Solutions

Other approaches divide the space into voxels, that is the volume elements representing values in the three-dimensional space (the pixel equivalents for 3D volume data) [CKBH00, LP05, LP06, MKKJ96, MTG97, SVZ00]. Then, each voxel is projected into all the images to test (using a projection test) whether they are contained in every silhouette. More efficient octree-based strategies have also been used to test voxels in a coarse to fine hierarchy [Pot87, Sze93].

The projection test is any function which determines whether the projection of a volume element belongs to a silhouette or not. There are many possible projection tests. Some are faster, and others more robust to noise, for instance. 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. In Chapter 7, an exhaustive analysis of different projection tests is given.

The voxel-based SfS algorithm for any projection test is the one shown in Algorithm 1.

Algorithm 1 Voxel-based SfS algorithm

```
Require: Silhouettes: $S(c)$, a Projection Test Function: $PT_c(voxel, Silhouette)$
for all $voxel$ do
    $voxel$ ← Foreground
    for all $c$ do
        if $PT_c(voxel, S(c))$ is false then
            $voxel$ ← Background
        end if
    end for
end for
```

The main advantages of voxel-based extraction of the Visual Hull are:

- The voxelized Visual Hull is an easy manner to represent the foreground volume. Its simplicity derives from the fact that the volume is labeled as a list of foreground and background voxels. Notice that the exact (geometric) Visual Hull consists of curved and irregular surface patches which are difficult to work with.

- Working with voxels also has the time-constancy key advantage. Given a specific voxelization of the space, the number of voxels is always the same and the voxel classification process always takes the same amount of time, independently of the size of the input silhouette images. The time of operation can be predicted, and therefore it is not necessary to have a buffer to temporally store the input data that the system is unable to process in particularly high-demanding temporal intervals, as may happen in geometric intersection SfS systems.

- Another important characteristic of the voxel-based SfS approach is that it allows an easy mechanism to analyze a particular volumetric region. To do so, one only needs to analyze the voxels corresponding to the volume of interest.

- The voxel-based SfS algorithm operates at real-time when using simple projection tests. In our implementation, it is possible to process a volume of $3966 \times 5245 \times 4000\ mm^3$, using voxels of 3 cm edge size at 15 fps.

- The voxel-based Visual Hull of a general 3D object can be visually represented using simple hardware primitives. In addition, the projection of the reconstructed shape to a virtual view takes an inappreciable period of time when using a hardware solution such as any of the current commercial 3D video-cards using OpenGL or equivalent graphics library.

- And finally, one of the most important reasons to use voxel-based representations of the Visual Hull is that there exists a vast amount of work developed in the last decades on pixels that can be extended to the voxel level. Some of these tools are, among others: morphological operators, connectivity analysis, multi-resolution analysis (octrees are the quadtrees equivalent in 3D), mathematical model fitting (snakes, ellipsoid fitting), level sets, codification techniques, and so on.

Having mismatches between the original silhouettes and the projected Visual Hull is the main drawback of the voxel-based approach. Voxels have a quantization effect that is specially apparent when using voxels of large size. *Geometric* approaches theoretically do not suffer from this problem. However, in practice the reconstructed Visual Hull of *geometric* systems neither match the original silhouettes, since original images need to be downscaled and approximated in real-time implementations.

There is yet another more fundamental aspect to be considered before deciding between the voxel-based and geometric approaches. Usually, the most important part of a 3D application is to be able to make decisions based on whether a volumetric point belongs to the foreground or not. This is exactly what voxel-based reconstructions provides. On the contrary, in geometric solutions the positions of the 3D points of interest have to be previously analyzed in order to determine whether they are inside the series of geometric patches defining the Visual Hull or not. The parallelism with the image level is also of relevance; an image is a discrete set of points modeling an observed reality and a voxel-based representation is a discrete set of points modeling a 3D reality, indirectly measured using any type of reconstruction method.

Notwithstanding the quantization effect, the accuracy of the voxel-based Visual Hull can be improved by increasing the resolution of the voxel-space. To do so, the voxel edge size has to be reduced. Figure 2.12 shows the difference between the projection of the Visual Hull and the real silhouettes, using the synthetic Kung Fu Girl dataset images.

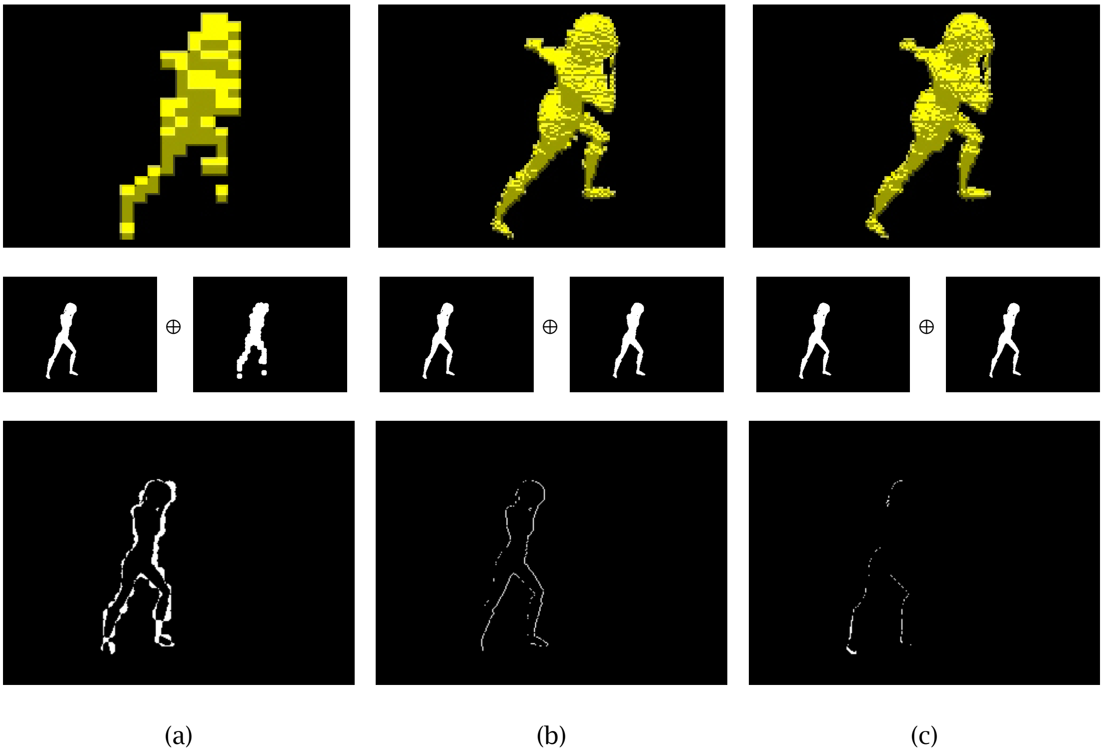


Figure 2.12: Effects of voxel resolution using 30 voxels per dimension, 166, and 300 from left to right, respectively.
The first row of images shows the reconstructed Visual Hull. In the second line, the original silhouette images and the projection of the Visual Hull are depicted. Note that the Visual Hull produces a coarser approximation of the silhouettes, specially in the lower resolution volume estimate in (a). The row at the bottom depicts the error after *xor*-ing the original silhouette images and the projection of the Visual Hull. Note that as the resolution increases, the error decreases.

2.2.4.3 Limitations of Shape from Silhouette

In 3D scenes, the Visual Hull is bounded by the Photo Hull and the Convex Hull. The Convex Hull does not contain hyperbolic regions, such as the arms and legs of a human body. In the Visual Hull, hyperbolic regions are effectively detected, but concavities cannot be detected since binary silhouettes do not provide enough information to detect concave patches. The Photo Hull is able to detect both concavities and hyperbolic regions. Think for instance in an opaque bowl. The projected silhouettes of the bowl do not suggest whether the bowl is covered with a lid or not, unless there is a camera inside the bowl. Space Carving deals better with this type of situation since the volume below the lid is not photo-consistent and the algorithm is able to carve below it.

SfS may also suffer from the *ghost* effect, consisting in the reconstruction of a non-existent entity. The effect usually appears when using a low number of cameras. See Figure 2.13, for an example using two 1D views of a 2D scene.

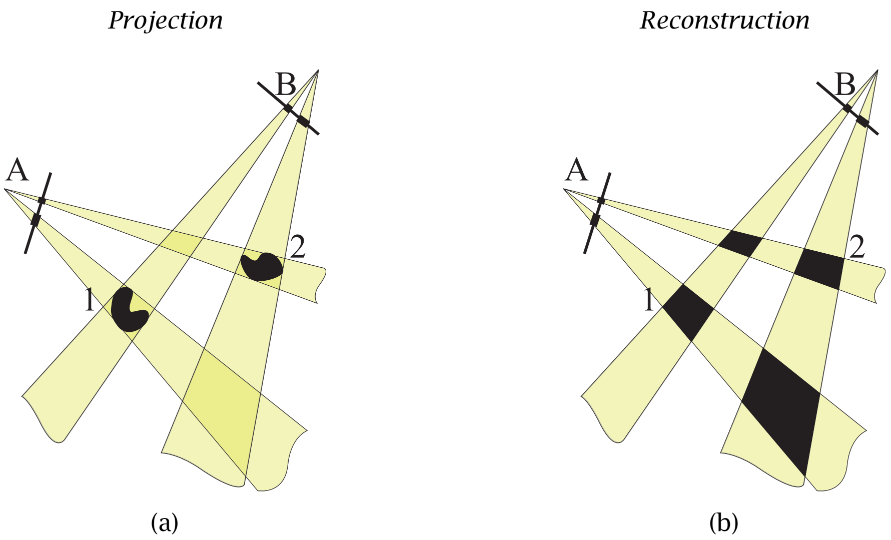
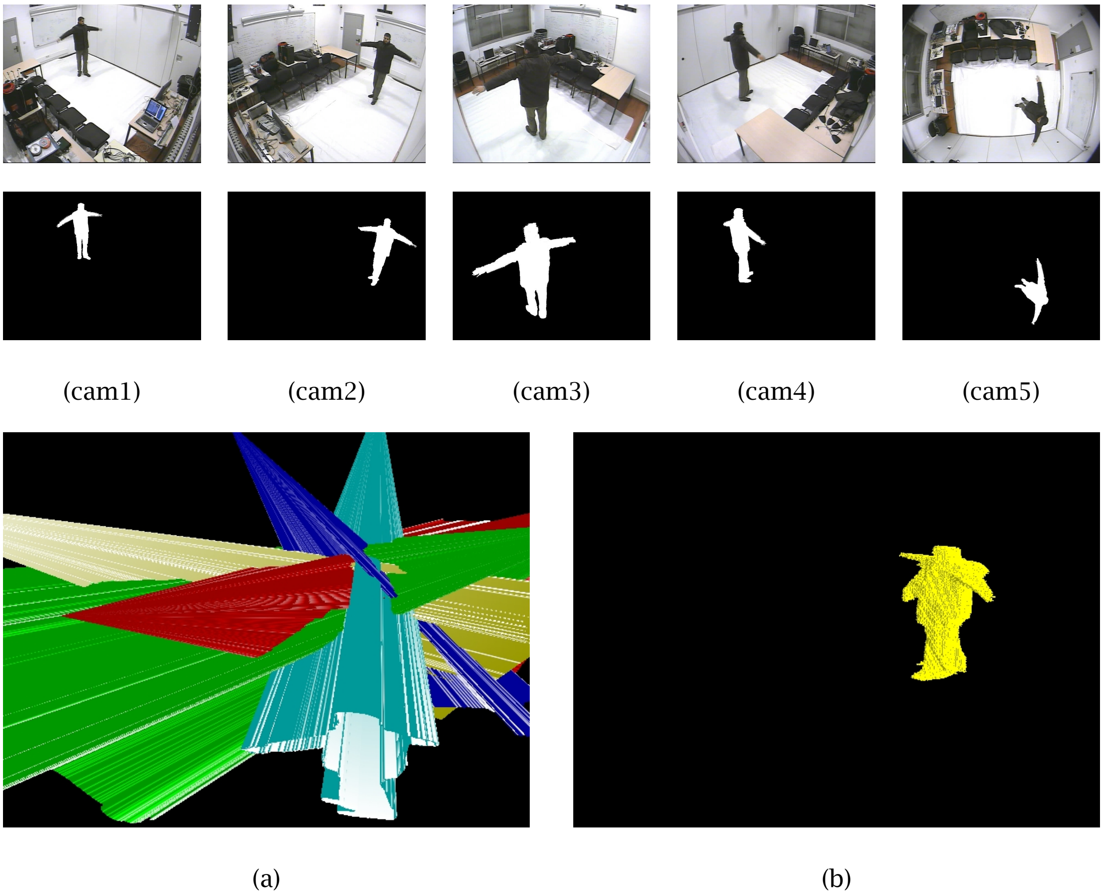


Figure 2.13: Projection and reconstruction process of a 2D scene using two 1D views.
In (a), it is shown how entities 1 and 2 are detected by cameras A and B. On the right, (b) shows the reconstruction of the Visual Hull. Notice that objects 1 and 2 are correctly reconstructed, though the reconstructed area is larger than the real shape on the left. Also note that two more fictitious *ghost* entities are also reconstructed since they are in the intersection of the visual cones. The problem can be alleviated by incorporating more cameras.

In 3D it is fairly uncommon to have the *ghost* effect, since the number of possible combinations of object locations is higher than in 2D. However, it is common to wrongly enlarge the entity's volume, if few cameras are used during the reconstruction process. See an example in Figure 2.14, where the reconstructed Visual Hull of a person appears as if the person had four arms. This effect can be easily explained by observing Figure 2.15. Note that the arrangement of the cameras makes it impossible to determine a more accurate estimate of the body. None of the cameras can resolve the ambiguity shown with a discontinuous line in Figure 2.15. In the figure, we indicate a possible location of a new camera (the one filled in with a rhomboidal pattern) that could have resolved this particular uncertainty. The general rule is that the more cameras used, the better the reconstructed shape. In an ideal situation, the reconstructed shape will be exactly the same as the shape of the real object, except for the concavities, which are undetectable using Shape from Silhouette.



(cam1) (cam2) (cam3) (cam4) (cam5)

(a) (b)

Figure 2.14: Visual cones intersection of a person in the smart-room of our lab. The images show a particular situation where the arms of the person are aligned so that the cameras cannot disambiguate the exact location of the extremities, and therefore the Visual Hull appears as if the person had 4 arms.
Using five views, the visual cones intersect constraining the shape estimates of the person. From (cam1) to (cam5), the original images and corresponding silhouettes are shown. At the bottom, (a) depicts the visual cones, and (b) shows the intersection of the visual cones, i.e., the Visual Hull.

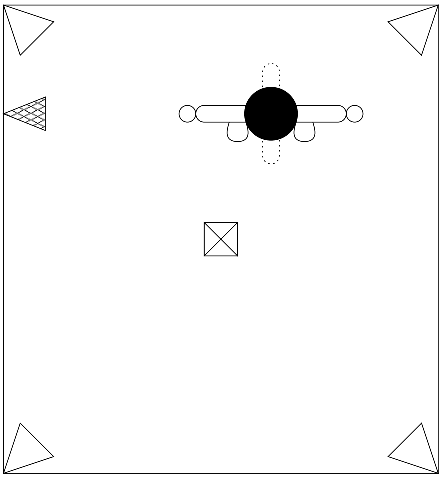


Figure 2.15: Representation of the zenithal perspective of the smart-room of the UPC showing the location of the person of Figure 2.14.
There are 4 cameras positioned in the room corners and a fifth one on the ceiling. The zones of uncertainty are shown in discontinuous line. Notice that the camera on the top cannot help in clarifying the scene for that the person is not aligned below it. Finally, in a rhomboidal pattern we show a possible location where a new camera could have resolved this particular uncertainty.

2.2.5 Single-Camera Volumetric Foreground Detection

Analogously as the Planar Detection Task can be performed using multiple cameras (see section 2.1.2), the volume estimate of an entity can be obtained using a single camera.

For static objects, the problem can be reduced to putting the object on a precisely calibrated turntable in front of a camera that registers the movement across time [Sze93]. However, if the turntable is not calibrated, then the unknown motion has to be estimated before the silhouettes can be combined across time [CKBH00]. More complex set-ups where the 3D structure of a rigid scene is reconstructed from a video sequence captured with a hand-held camera also exist [PGV$^+$04]. The technique, known as Structure from Motion, is far from real-time operation. However, it certainly gives impressive results.

Because these methods are based on tracking image points more than in scene space analysis, they will not be discussed further here.

2.3 Conclusions

In this chapter we have presented a literature review of the planar and volumetric foreground classification techniques used in the past, showing some of the different applications that 2D and 3D foreground segmentations cover.

The two-dimensional and three-dimensional foreground detection types have been presented as two independent processes. We have not made any assumption whether the silhouettes used in the SfS approach have been extracted with any particular method. However, it seems clear that in most of the cases the silhouettes are previously extracted using the planar foreground extraction approaches described in the first part of the chapter.

In the following chapter we present the 2D-3D detection integration as the main problem to resolve in this dissertation.

No new material has been presented in this chapter, although the exposition of the subject has followed a personal perspective. In particular, we have rewritten some of the more popular two-dimensional Gaussian-based background models in a unified notation, making it possible an easier comparison of the models.

Chapter 3

Problem Statement

SHAPE FROM SILHOUETTE was designed assuming that silhouette images are error-free, and therefore ready to be used as inputs of the algorithm. However, silhouette extraction is a problem in itself, and silhouettes are quite commonly defective.

From a three dimensional perspective, it seems reasonable to hypothesize that the reconstructed volume is a more robust detection, in view to the fact that the Visual Hull is reconstructed making use of multiple silhouettes. Under this assumption, the projection of the Visual Hull in each camera view could be used to help maintaining 2D models. However, in reality, the reconstructed volume is not necessarily more robust than two-dimensional detections. In fact, the Visual Hull inherits some of the two-dimensional weaknesses in a cumulative way.

In this chapter, we comment on the issues regarding two-dimensional error propagation to the three dimensional space. We also state the problem of how to shift the 2D-models update step from the pixel level to the more-informed volume level, assuming that we have the tools to correct the cumulative error propagation to the third dimension.

Finally, we propose the integrated planar-volumetric system which overcomes some of the limitations of current planar and volumetric systems. The problems and ideas here exposed are developed in the following chapters.

3.1 MAP *vs.* Exception-to-Background

In the most popular two-dimensional foreground segmentation systems, pixels are classified as foreground using an exception-to-background approach, that is, pixels are classified as foreground only when they do not belong to the background class (see Figure 3.1).

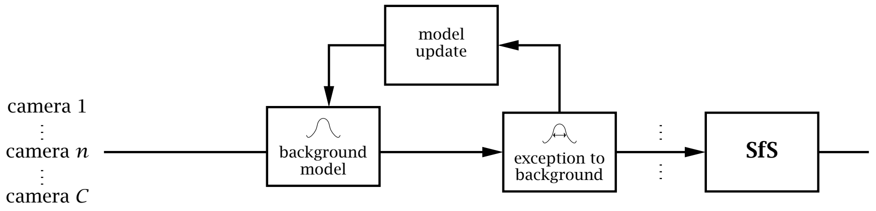


Figure 3.1: The schematic system block diagram of the classic exception-to-background approach for the planar foreground segmentation task. The output of the classifiers can be used to obtain a volume estimate of the three dimensional object shape.

An exception-to-background classification scheme is only focused towards pixel classification. In order to refine the classification process with more information about the underlying process, we rather compute the pixel's foreground and background likelihoods. This way, it is not only possible to classify the pixels using maximum a posteriori (MAP), but also to obtain a measure of how reliable the classification is.

The system outlined in Figure 3.2 shows how an MAP-based classification can be performed at the two-dimensional level. Following this line of thoughts, in the following chapter we propose an MAP extension to the most popular exception-to-background systems [EDHD99, SG00b, WADP97], so that it is possible to obtain more accurate segmentations. The proposed scheme exploits the probability of the classification at the model update step. In addition, the scheme can incorporate other probabilistic cues that are external to the pixel process. Finally, this system paves the way for a Bayesian SfS scheme, which makes use of 2D probabilities instead of the binary values indicating foreground or background.

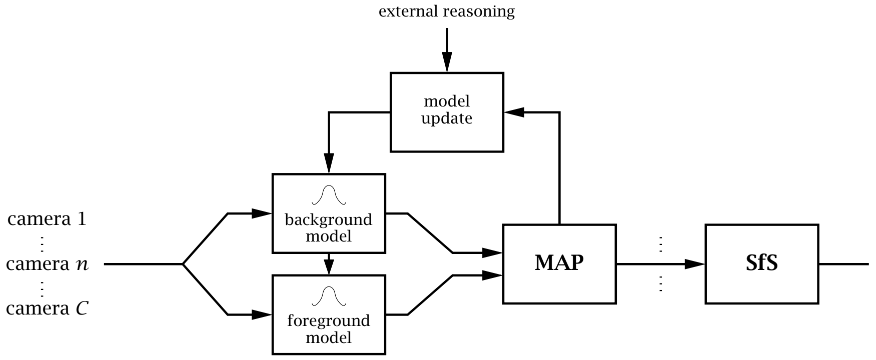


Figure 3.2: The schematic diagram of an MAP based foreground classifier. Note that both foreground and background likelihoods are required for proper operation. Also note that MAP decisions may be accompanied by some external reasoning, relieving some of the problems of more naive approaches that assume that all pixels are always updated only by inspection of the classification result.

3.2 MAP in the 3D Level

In SfS, the Visual Hull is obtained by summing the number of visual cone intersections in all the 3D points. If the number of intersections is equal to the total number of cameras, then the volumetric points are considered to be part of the Visual Hull. This approach assumes that 2D detections are unequivocal even though 2D decisions are often wrong. Instead, we propose to accompany the visual cones with their respective probabilities of foreground and background.

In our proposal, the probabilistic values that the pixels belong to the foreground or background classes are used to decide at the volume level. This scheme has a number of advantages over more traditional ones. First, we can set foreground and background priors at the volume level instead of at the 2D level. That is, we can say before hand how probable is that a volumetric point belongs to the foreground, based on last observations or on the arrangement of the scenario. Then, we can force the posteriors not to deviate from the prior knowledge introduced to the system.

Another advantage of a Bayesian approach at the 3D level is that we can use the projection of the probabilistic Visual Hull as the base to set the adaptation speed of the 2D models using the update step outlined in the previous section.

In order to obtain reconstructions with a low number of reconstruction errors that also project with low two-dimensional error rates, it is possible to exploit the geometrical constraints that the reconstruction of the Visual Hull must accomplish.

3.3 Geometric Constraints of the Visual Hull

Note that even a small undetected foreground region in a silhouette image can inhibit a proper reconstruction of the foreground volume estimate. On the contrary, it is unusual that two-dimensional false alarms propagate to the third dimension since the false visual cones should still intersect with other visual cones from the rest of cameras. This idea is fully developed in chapter 7.

So, the Visual Hull is more robust than silhouette images in terms of false alarms, but it accumulates misses from all the images to the third dimension. In other words, SfS is an ideal system when silhouette images only have false alarms, but it is considerably faulty with silhouette misses.

To solve this problem we have developed a technique called Shape from Inconsistent Silhouette. The underlying idea is that planar foreground detections and volume projections should match. If they do not match, then the most probable volume estimate -that we call Unbiased Hull- is chosen.

3.4 System Proposal

Finally, we propose to take decisions at the more informed volumetric level and to refine them by exploiting the geometrical constraints of the reconstructions. The final system design is shown in Figure 3.3.

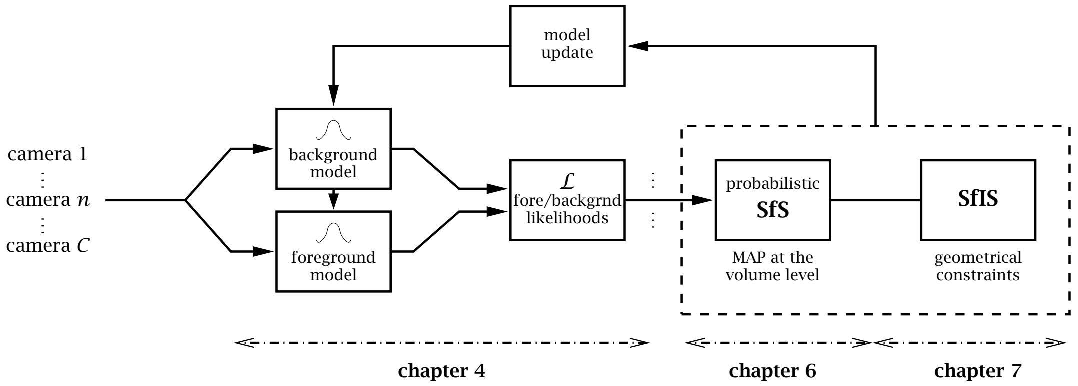


Figure 3.3: The schematic diagram of the novel reconstruction approach made up of three main processing steps. In the first step, the foreground and background likelihoods are processed. Later, a Bayesian SfS is employed to reconstruct the Visual Hull, that is geometrically inspected to force accomplishing the geometric constraints. Finally, the corrected reconstructed volume -the Unbiased Hull- is used to update two-dimensional models making use of simple external rules.

3.5 Conclusions

The main challenge that this dissertation faces is how to build a unified system that extracts planar and volumetric foreground detections in a unified framework.

The problem has been tackled in the past assuming separate two-dimensional and three dimensional processes. In a first stage, planar detections were obtained and later combined to obtain a volumetric estimate using the Shape from Silhouette technique. This approach, however, is unidirectional, and proceeds from the planar to the volumetric level degrading at each stage in an unrecoverable manner.

Instead, we propose to take decisions at the more informed volumetric level and to refine them by exploiting the geometrical constraints of the reconstructions, that is, the projection of the reconstructed volume has to match with the original silhouette images. Thus, the volume estimates are more robust than planar detections, and therefore can be used to assist during the planar models update stage.

Chapter 4

A Bayesian Interpretation of the Planar Foreground Segmentation Task

OVER THE YEARS, many works have been published on the two-dimensional foreground segmentation task, describing different methods that treat to extract that part of the scene containing active entities.

In most of the cases, the stochastic background process is modeled first, and then the foreground pixels are classified as an exception to the model. In other setups, the foreground process is also modeled, and the scene is classified using maximum a posteriori (MAP) or maximum likelihood (ML). Then, in order to guarantee accurate results along the time, the background models are continuously updated making use of all the pixel values that are classified as background. In this particular type of framework, all background observations contribute equally to update the background models. However, it seems reasonable to assume that observations with higher background probabilities should weight more than those with lower probabilities.

The main contribution of this chapter is the development of a theoretical framework in which pixel classification and update stages can be fully explained as a Bayesian procedure, where one stage is probabilistically related to the other. In addition, the update stage of the presented system permits to include higher-level probabilistic information that is external to the pixel process. This ability will be exploited in subsequent chapters.

In the chapter, most of the different classification and model update mechanisms in the literature are explored. The limitations of these systems, as well as their common assumptions, are explained. Then, our Bayesian framework is introduced. Finally, it is discussed how the proposed framework can accommodate standard background models, obtaining better results than other frameworks using the same models. These claims are supported by both theoretical developments and real-world examples.

4.1 Introduction

The most successful methods for detecting foreground observations at each pixel correspond to those that classify observations into foreground as an exception to a background model [EDHD99, HHD99, HHD00, MJD$^+$00, SG00b, WADP97] (see §2.1, on page 5 for more details). Of course, foreground models can also help. However, it is often difficult to model the color appearances of moving objects. Besides, the foreground models of each object have to be mapped to each pixel at each instant before performing a classification, which is also prone to errors. On the contrary, the background appearance of a pixel can be robustly learned using fixed cameras and, therefore, background models usually provide the most reliable source of information in the segmentation task.

In this chapter, we propose a maximum a posteriori classification framework. This framework allows easy incorporation of well-known background models and permits to obtain an estimate of the system's error rate and the probability of each classification. We also provide a simple solution for not having to model the foreground appearances and still be able to use the probabilistic setting.

Classification is only half of the problem. A good foreground segmentation system must count with both proper classification and model update. The update step has normally been linked to the classification step using reasonable assumptions, but without proper probabilistic justification. Usually, in order to maintain the background models along the time, the parameters of the models are continuously re-estimated. The maintenance process is based on the maximization of the likelihood of the background observations in the recent past. Thus, only the samples that have been previously categorized as background are used to update the background model parameters. Note that the possibility of having errors in the classification is not considered in this scheme. However, it seems reasonable to assume that background observations with higher background probabilities should weight more than those with lower probabilities. In the chapter, we show that this assumption is correct. To do so, we propose to maximize the complete pixel likelihood including both the foreground and background processes. The equations obtained show that the *classification* and *model update* steps are in fact probabilistically connected one to the other.

To sum up, the classification step of our scheme provides the probabilities that are then used to update the models. This is a key difference with other methods that simply use observations that have been classified as background, without considering any background probability whatsoever. Moreover, this new scheme opens the doors to the possibility of incorporating other sources of information that provide solid information about pixel probabilities. One of these external sources of information, consisting in projecting more-informed 3D probabilistic maps, will be detailed described in chapter 6.

The remainder of the chapter is structured as follows. In the next section, a short overview of the foreground and background models that have been widely used in the literature is given.

Section 4.3 is devoted to discussion of MAP and exception-to-background classification systems. Section 4.4 presents our model update technique, based on the maximization of the expected likelihood of foreground and background observations of a pixel. In the section, we also derive the equations for online model maintenance. In section 4.5, we present an experimental study of the system and, finally, the chapter concludes in section 4.6 with an overview of the main contributions presented in this chapter.

4.2 Literature Review on Foreground and Background Models

In order to perform a Bayesian classification of the imaging scene, it is important to characterize first both the foreground and background processes. In the following, we give an overview of some of the models that have been used in the literature that can be employed in the maximum a posteriori framework that we propose.

4.2.1 Successful Background Models Overview

In the literature review of this thesis (see §2.1.1) we already devised some of the background models that have been successfully implemented in many systems. Before a detailed description of the method that we present follows, it appears to be useful to first make a summary of these models.

A very fast method to learn and update a representation of the background of the imaging scene is to model the background color at each pixel location fitting a Gaussian function. This model was formalized in equation (2.1) in the literature review given in §2.1.1 on page 7.

A more elaborated method consists in using a Mixture of Gaussians (MoG) to model the background process at each pixel. This is very similar to the previous method. But, in addition, an MoG is also able to model a background scene that is constantly changing along the time such as in raining situations, waving flags, water, etc. See equation (2.2), also in §2.1.1, for its mathematical formulation.

Finally, it is possible to obtain better approximations of the background process at each pixel by learning a smooth continuous version of the histogram obtained from the last number of observed values in the same pixel location (see (2.3) in §2.1.1). This can be achieved by summing one Gaussian centered at a pixel value for each sample that is observed in the same location along the time. The choice of the deviation of the Gaussians -also called bandwidth- is theoretically insignificant assuming that the number of samples being used tends to infinite. However, in practical situations, the choice of the bandwidth is critical not to over-smooth neither to obtain a ragged density estimate. The practical considerations of the method are discussed in [EDHD99].

All the mentioned models above share that they use a pdf function to represent the background, i.e., the function is non-negative everywhere and its integral from $-\infty$ to ∞ is equal to 1.

The MAP-based foreground segmentation scheme that is presented in this chapter may be used with any models which are expressed as a pdf, including the models described above. These models have been introduced because they have been extensively used in the past with good results. But, of course, any other background models which can be expressed as a probability density function can be also employed.

4.2.2 Foreground Models

In order to make use of a maximum a posteriori setting, a foreground model must also exist. In addition, the foreground model has to be in the form of a pdf. This is not a trivial task and is the main reason that has conducted many practitioners to abandon the MAP setting and to adopt exception-to-background.

The main problem is how to obtain a reliable characterization of the foreground process of a pixel. The foreground entities of an image are those which are in a prominent place in a scene, due to the fact that they are constantly moving. Therefore, it is difficult to obtain a foreground characterization by inspection of a single pixel location, without using global information of what is happening at the whole image level.

In spite of that, in the method of Stauffer and Grimson [ISBG99, SG00b], a foreground model is obtained using only per-pixel information. This is achieved by creating a new Gaussian for any pixel color which does not fit the background class. The Gaussian is updated along the time with the pixel values that are clustered into it. A foreground model is therefore assumed during a certain period. Finally, if the period of observation is long enough, the foreground model is adopted as part of the background, losing the stochastic characterization of the foreground.

The method of S&G has proved to be very reliable, yet it seems reasonable to assume that the foreground process may be better characterized making use of the global image context. Several approaches -which were presented in §2.1.1- have been proposed in the literature that try to characterize complete foreground entities (the so-called blobs in the literature) [EDHD99, HHD00, KS00a, LHGT04, MD02, MJD+00, MRG99, MRG99]. Basically, in these approaches each foreground entity is characterized by means of a complex model that takes into account the geometrical properties of the entity. The fundamental premise of these methods is that a tracker is employed so that the foreground models of each entity can be correctly updated along the time. Finally, in a per-pixel MAP setting, the blob-based foreground models must be mapped to each pixel before performing the foreground segmentation. An MAP setting has been used in [KS00a, LHGT04, MD02], among others. However, in the model update stage of these works, the update process is taken as a separate task, not making use of the classification probabilities that MAP provides.

Before we present the foundations of our background learning technique, we have considered important to provide also a method which allows using MAP without requiring a complex setup to obtain foreground models. To do so, we propose using a uniform pdf to model the foreground process at each pixel. Indeed, more elaborated foreground models may allow to obtain better results. However, an MAP setting excels exception-to-background methods even with this naive foreground characterization, as will be shown in the following section.

Assuming that we do not have any clue about the foreground process in the scene we can, however, consider that in images with D channels, each pixel in the image has values in $\mathbf{D} = \{0, \cdots, 255\}^D$. Then, without more information about the foreground entities, we can assume that the likelihood of observing one of the values in **D**, given that it belongs to a foreground process is

$$p(\mathbf{I_x}|\text{foreground}) = \frac{1}{256^D}, \tag{4.1}$$

where $\mathbf{I_x}$ denotes the value of a certain pixel **x** in the image.

4.3 MAP *vs.* Exception-to-Background Classification

Several pixel foreground/background classification settings have been proposed in the past. In chapter 2, §2.1 on page 5 , we already devised some of the groups in which the foreground extraction techniques could be categorized. In the following lines, we rapidly go through these techniques again, providing their error probabilities and principal characteristics. First, the maximum a posteriori and maximum likelihood settings are formulated, and then the exception-to-background method is outlined. The model update part will be covered in the next section. As will be shown, in our proposal, model maintenance is partially built on the classification procedures reviewed in this section, and particularly on MAP. Thus, it is important to provide first a solid foundation of the classification methods in order to introduce the update scheme later.

A per-pixel probabilistic foreground and background classification setting associates with each class a random variable. In this particular classification problem there are only two classes under consideration: foreground, denoted with symbol ϕ, and background β; and each class is associated with the random variables: V_ϕ and V_β, with probability functions $p(\mathbf{I_x}|\phi)$ and $p(\mathbf{I_x}|\beta)$, respectively.

The classification task can be solved by choosing for each pixel the most probable class, i.e., that one with the highest probability.

Therefore, a pixel is classified into foreground if

$$\phi = \operatorname*{argmax}_{c=\{\phi,\beta\}} P(c|\mathbf{I_x}), \tag{4.2}$$

and analogously, a pixel is considered to belong to the background stochastic process if

$$\beta = \operatorname*{argmax}_{c=\{\phi,\beta\}} P(c|\mathbf{I_x}). \tag{4.3}$$

If we assume that the classes of each pixel are independent, then a pixel can be classified as foreground if

$$\phi = \operatorname*{argmax}_{c=\{\phi,\beta\}} \frac{P(c)p(\mathbf{I_x}|c)}{p(\mathbf{I_x})} \tag{4.4}$$

$$= \operatorname*{argmax}_{c=\{\phi,\beta\}} \frac{P(c)p(\mathbf{I_x}|c)}{P(\phi)p(\mathbf{I_x}|\phi) + P(\beta)p(\mathbf{I_x}|\beta)}, \tag{4.5}$$

which completes the maximum a posteriori setting.

Also, note that the MAP setting admits the alternative formulation,

$$\phi = \operatorname*{argmax}_{c=\{\phi,\beta\}} p(\mathbf{I_x}|c)P(c), \tag{4.6}$$

since $p(\mathbf{I_x})$ is independent of the ϕ class.

Another similar classification setting is the maximum likelihood (ML) approach, which does not take into account the prior probabilities of each class.

In an ML setting the classification is achieved by inspection of the pixel likelihoods of each class. For instance, a pixel is classified as foreground if

$$\phi = \operatorname*{argmax}_{c=\{\phi,\beta\}} p(\mathbf{I_x}|c), \tag{4.7}$$

which is different from (4.2) in that ML chooses the class that maximizes the likelihood of an observation, instead of choosing the most probable class, given an observation.

In fact, an exception-to-background setting is very similar to ML in that one only uses the likelihood functions. However, an exception-to-background setting differs from ML in that only the likelihood of the background process is used, and a pixel is classified as foreground when the likelihood of the observation is sufficiently low:

$$p(\mathbf{I_x}|\beta) < T, \tag{4.8}$$

where T is a threshold value that may be obtained using a training data set, for instance.

In general, MAP is always preferable to ML, since MAP-based classification can be seen as a more complete version of ML that incorporates prior information. In other words, MAP can be understood as a regularization ML when there exists prior information of the classes to be chosen. If there does not exist prior information, then MAP and ML are equivalent.

Let us finally point out some considerations concerning the per-pixel approach described. In this approach, each pixel is classified depending only on its value, supposing that the classifications of the pixels are independent of each other. This is obviously false in most images where there are groups of pixels (regions) with homogeneous characteristics.

This problem, which is finally the cause of the *salt and pepper* noise, is usually found in many foreground segmentation results, and may be solved after the classification process itself. In the following chapter, devoted to the practical aspects of the background learning and object detection/tracking approaches, we present a simple method to solve the salt and pepper issue (see §5.4). In short, the approach proposed consists in defining a neighborhood of pixels and to use them to classify not only the values of individual pixels but the rest of the pixels in the neighborhood. In addition, the spatial coherence of the result can be improved using a number of methods such as morphological operators designed to operate over spurious pixel detections.

4.3.1 Classification Probabilities

Assuming that we are using the uniform foreground model in (4.1), we can derive the foreground and background probabilities for each pixel,

$$P(\phi|\mathbf{I_x}) = \frac{P(\phi)p(\mathbf{I_x}|\phi)}{P(\phi)p(\mathbf{I_x}|\phi)+P(\beta)p(\mathbf{I_x}|\beta)} = \frac{P(\phi)\frac{1}{256^D}}{P(\phi)\frac{1}{256^D}+P(\beta)p(\mathbf{I_x}|\beta)} \tag{4.9}$$

$$P(\beta|\mathbf{I_x}) = \frac{P(\beta)p(\mathbf{I_x}|\beta)}{P(\phi)p(\mathbf{I_x}|\phi)+P(\beta)p(\mathbf{I_x}|\beta)} = \frac{P(\beta)p(\mathbf{I_x}|\beta)}{P(\phi)\frac{1}{256^D}+P(\beta)p(\mathbf{I_x}|\beta)}. \tag{4.10}$$

For the sake of simplicity, lets us consider a simple background model, using only one Gaussian to represent a single background mode. Inherently, this model assumes that the background is static, without moving leaves in a tree or waving flags, water and so on. The model, which was introduced in §2.1.1, is:

$$\mathrm{G}_\mathbf{x}(\mathbf{I_x}) = \frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_\mathbf{x}|}}e^{-\frac{1}{2}(\mathbf{I_x}-\mu_\mathbf{x})^T\Sigma_\mathbf{x}^{-1}(\mathbf{I_x}-\mu_\mathbf{x})}. \tag{4.11}$$

which leads to

$$P(\phi|\mathbf{I_x}) = \frac{P(\phi)\frac{1}{256^D}}{P(\phi)\frac{1}{256^D}+P(\beta)\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_\mathbf{x}|}}e^{-\frac{1}{2}(\mathbf{I_x}-\mu_\mathbf{x})^T\Sigma_\mathbf{x}^{-1}(\mathbf{I_x}-\mu_\mathbf{x})}} \tag{4.12}$$

$$P(\beta|\mathbf{I_x}) = \frac{P(\beta)\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_\mathbf{x}|}}e^{-\frac{1}{2}(\mathbf{I_x}-\mu_\mathbf{x})^T\Sigma_\mathbf{x}^{-1}(\mathbf{I_x}-\mu_\mathbf{x})}}{P(\phi)\frac{1}{256^D}+P(\beta)\frac{1}{(2\pi)^{D/2}\sqrt{|\Sigma_\mathbf{x}|}}e^{-\frac{1}{2}(\mathbf{I_x}-\mu_\mathbf{x})^T\Sigma_\mathbf{x}^{-1}(\mathbf{I_x}-\mu_\mathbf{x})}}. \tag{4.13}$$

If we further assume a grayscale color space, i.e., $D = 1$, then the posterior probabilities are

$$P(\phi|\mathbf{I_x}) = \frac{P(\phi)\frac{1}{256}}{P(\phi)\frac{1}{256}+P(\beta)\frac{1}{\sqrt{2\pi}\sigma_\mathbf{x}}e^{-\frac{1}{2}(\frac{\mathbf{I_x}-\mu_\mathbf{x}}{\sigma_\mathbf{x}})^2}} \tag{4.14}$$

$$P(\beta|\mathbf{I_x}) = \frac{P(\beta)\frac{1}{\sqrt{2\pi}\sigma_\mathbf{x}}e^{-\frac{1}{2}(\frac{\mathbf{I_x}-\mu_\mathbf{x}}{\sigma_\mathbf{x}})^2}}{P(\phi)\frac{1}{256}+P(\beta)\frac{1}{\sqrt{2\pi}\sigma_\mathbf{x}}e^{-\frac{1}{2}(\frac{\mathbf{I_x}-\mu_\mathbf{x}}{\sigma_\mathbf{x}})^2}}, \tag{4.15}$$

which are depicted for a couple of cases in Figure 4.1. It is interesting to observe how the camera's thermal noise and the model parameter $\sigma_\mathbf{x}$ used to represent that noise is related to each other. Note that in high quality cameras, i.e., cameras with low thermal noise, all background observations in a pixel tend to be very stable along the time, leading to small values of $\sigma_{\mathbf{x}}$. Under small values of the deviation, MAP does a good job at discriminating between foreground and background observations. However, when the noise in the system is strong (large values of $\sigma_{\mathbf{x}}$), then the MAP setting does not clearly favor one class above the other. This can be clearly observed in the figure mentioned before, where the foreground and background probabilistic curves asymptotically tend to their prior values as the $\sigma_{\mathbf{x}}$ grows.

With regard to the exception-to-background setting, classification probabilities cannot be derived since the method is only intended for taking decisions. If nothing else, the exception-to-background setting provides the likelihood that an observation belongs to the background model. However, since the background likelihood cannot be compared to the foreground likelihood, this piece of information alone does not seem to be very useful in most of the cases.

4.3.2 Classification Error

As mentioned before, in an exception-to-background segmentation setting it is impossible to obtain a measure of the probability of a given classification. On the contrary, we have shown that an MAP setting provides the posterior probabilities of each class. Moreover, in the following lines, we show that in the MAP setting it is also possible to obtain a measure of classification reliability in terms of the error probabilities.

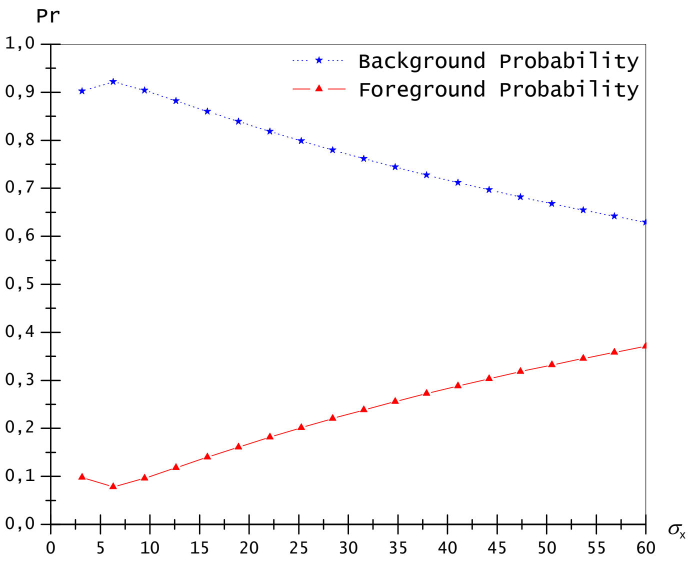
Note that it is important to know the segmentation error rate to inform to the subsequent parts of the system that make use of the classifications. For instance, the error rate of a foreground segmentation scheme is critical in the 3D reconstruction module that we present in chapter 7.

In order to present the formulation of the error rate of a foreground segmentation scheme, let us assume that we are using the foreground and background models in (4.1) and (4.11), respectively. We can graphically express both functions in Figure 4.2.

There are many sources of stochastic fluctuation. Suppose an observation $\mathbf{I}_{\mathbf{x}}$ is made leading to a decision $\hat{c}$. We can summarize average performance in terms of a confusion matrix, $P(\hat{c}|c)$, which for the detection task is a 2×2 array representing the hit (correct detections): $P(\hat{\phi}|\phi)$, false positive, also known as false alarm: $P(\hat{\phi}|\beta)$, false negative (miss): $P(\hat{\beta}|\phi)$, and correct rejection rates: $P(\hat{\beta}|\beta)$,

$$\begin{pmatrix} P(\hat{\phi}|\phi) & P(\hat{\phi}|\beta) \\ P(\hat{\beta}|\phi) & P(\hat{\beta}|\beta) \end{pmatrix}. \tag{4.16}$$

In other words, there is a hit when a pixel is correctly classified as foreground. A false alarm happens when a foreground detection is reported where none exists. A target is missed when a pixel belonging to the foreground is not recognized and reported. And finally, a correct rejection corresponds to a correct background classification. Note that the hit and miss rates sum one, and similarly for the false positive and correct rejection rates.



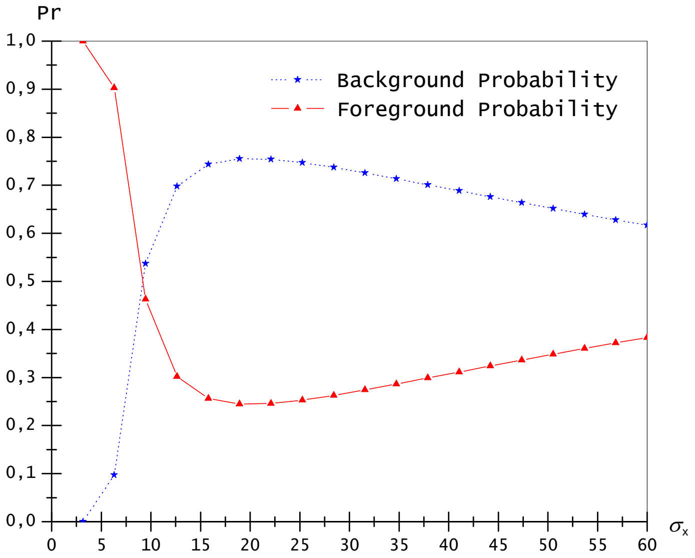


Figure 4.1: The two images above show the foreground and background probabilities of a pixel for different values of $\sigma_{\mathbf{x}}$, assuming a uniform foreground distribution in the grayscale color space. The image on the top shows the probabilities in the case where $\mu_{\mathbf{x}} - \mathbf{I}_{\mathbf{x}} = 5$. On the bottom, $\mu_{\mathbf{x}} - \mathbf{I}_{\mathbf{x}} = 20$. In both cases, when $\sigma_{\mathbf{x}}$ tends to 0, the Gaussian function is so narrow that the system tends to classify as foreground all the observations which are not very close to the mean of the Gaussian function. As the deviation increases, the probabilities tend to their prior values. In this case, we have set $P(\phi) = P(\beta)$, and therefore both functions tend to $\frac{1}{2}$.

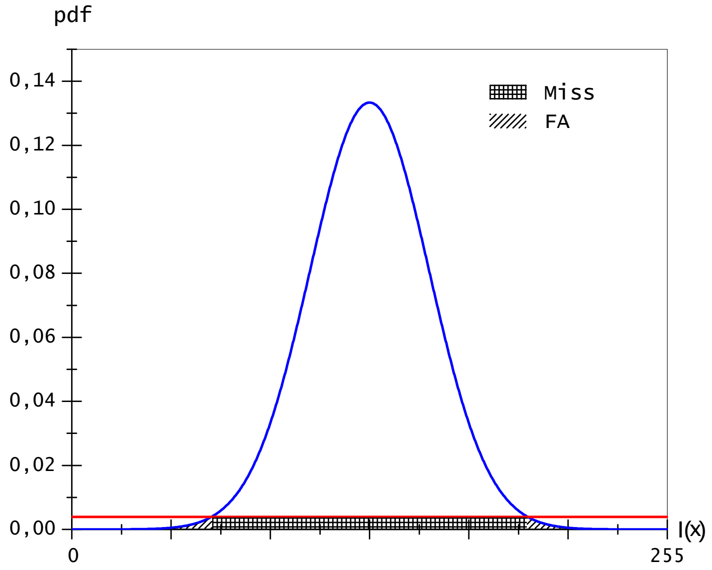


Figure 4.2: Probability density functions of a Gaussian and a uniform ($\frac{1}{256}$) distributions.

Obviously, an ideal system would maximize the frequency of hits and rejections while minimizing the frequency of misses and false alarms.

If the probabilistic distributions are known, the best decision one can make is choosing one class or another according to (4.4). Graphically, in an ML setting, the decision threshold between foreground and background can be set at the crossing of the two distributions. In the MAP setting, the distributions are previously weighted according to their prior probabilities. In the rest of the section it is considered that the priors of foreground and background are unknown and therefore fixed to the same value, deriving the same equations for the ML and MAP setting.

For example, consider the foreground detection task using the models depicted in the figure and assume that we do not have information about the priors. In that case, the cross of both distributions occurs at

$$\frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} e^{-\left(\frac{\mathbf{I}^{\diamond}_{\mathbf{x}}}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} = \frac{1}{256}, \tag{4.17}$$

where $\mathbf{I}^{\diamond}_{\mathbf{x}}$ will be used from now on to represent $\mathbf{I}_{\mathbf{x}} - \mu_{\mathbf{x}}$, reducing the following equations into simpler forms.

After isolating $\mathbf{I}^{\diamond}_{\mathbf{x}}$

$$\mathbf{I}_{\mathbf{x}}^{\diamond} = \pm\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}. \tag{4.18}$$

According to the equation above, the best decision that can be made is choosing background (β) for all the observations where the likelihood of background is higher than foreground. That is, we choose background for all the observed samples that lie between $\mathbf{I}_{\mathbf{x}}^{\diamond} > -\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$ and $\mathbf{I}_{\mathbf{x}}^{\diamond} < \sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$, and choose foreground (ϕ), otherwise.

Now, let us comment on the performance costs related with the decision scheme outlined above. From the figure, a false alarm (false positive) corresponds to any background observation $\mathbf{I}_{\mathbf{x}}^{\diamond}$ so that $\mathbf{I}_{\mathbf{x}}^{\diamond} > \sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$ or $\mathbf{I}_{\mathbf{x}}^{\diamond} < -\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$. Then, the probability of false alarm can be calculated by taking the integral of the background likelihood function on this interval i.e., in the tails of the Gaussian function,

$$P(\hat{\phi}|\beta) = \int_{-\infty}^{-\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}} \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}}e^{-\left(\frac{\mathbf{I}_{\mathbf{x}}^{\diamond}}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} d\mathbf{I}_{\mathbf{x}}^{\diamond} + \int_{\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}}e^{-\left(\frac{\mathbf{I}_{\mathbf{x}}^{\diamond}}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} d\mathbf{I}_{\mathbf{x}}^{\diamond}, \tag{4.19}$$

which can also be expressed in terms of the erf function as

$$P(\hat{\phi}|\beta) = 1 - \operatorname{erf}\left(\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}\right). \tag{4.20}$$

And since the probability of correct rejection is $1 - P(\hat{\phi}|\beta)$, then

$$P(\hat{\beta}|\beta) = \operatorname{erf}\left(\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}\right). \tag{4.21}$$

The probability of miss (false negative) can be calculated in a similar fashion as in the probability of false positive. From the figure, a false negative corresponds to any observation $\mathbf{I}_{\mathbf{x}}^{\diamond}$ so that $\mathbf{I}_{\mathbf{x}}^{\diamond} < \sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$ and $\mathbf{I}_{\mathbf{x}}^{\diamond} > -\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$. Then, the probability can be calculated by taking the integral of the foreground probability density function on the integration interval:

$$P(\hat{\beta}|\phi) = \int_{-\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}}^{\sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}} \frac{1}{256} = \frac{2\sqrt{2}\sigma_{\mathbf{x}}}{256}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}. \tag{4.22}$$

Then, the probability of hit is

$$P(\hat{\phi}|\phi) = 1 - \frac{2\sqrt{2}\sigma_{\mathbf{x}}}{256}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}. \tag{4.23}$$

And so we finally have the following confusion matrix for the ML/MAP setting

$$\begin{pmatrix} P(\hat{\phi}|\phi) & P(\hat{\phi}|\beta) \\ P(\hat{\beta}|\phi) & P(\hat{\beta}|\beta) \end{pmatrix}_{\text{ML/MAP}} = \begin{pmatrix} 1 - \frac{2\sqrt{2}\sigma_{\mathbf{x}}}{256}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)} & 1 - \operatorname{erf}\left(\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}\right) \\ \frac{2\sqrt{2}\sigma_{\mathbf{x}}}{256}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)} & \operatorname{erf}\left(\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}\right) \end{pmatrix}. \tag{4.24}$$

Apparently it is impossible to have such a complete confusion matrix for the exception-to-background setting. However, it is possible to obtain a measure of the performance of that setting after making some assumptions.

Since in an exception-to-background setting it is common to decide background when the observation is within 2.5 standard deviations of a distribution, then the equivalent to the point of cross in the ML/MAP setting (4.18) is

$$\mathbf{I}_{\mathbf{x}}^{\diamond} = \pm 2.5\sigma_{\mathbf{x}}. \tag{4.25}$$

Therefore, the probabilities of false alarm and correct rejection can be calculated similarly as in the ML/MAP setting.

The probability of false alarm is

$$P(\hat{\phi}|\beta) = \int_{-\infty}^{-2.5\sigma_{\mathbf{x}}} \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} e^{-\left(\frac{\mathbf{I}_{\mathbf{x}}^{\diamond}}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} d\mathbf{I}_{\mathbf{x}}^{\diamond} + \int_{2.5\sigma_{\mathbf{x}}}^{\infty} \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} e^{-\left(\frac{\mathbf{I}_{\mathbf{x}}^{\diamond}}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} d\mathbf{I}_{\mathbf{x}}^{\diamond}, \tag{4.26}$$

and in terms of the erf function

$$P(\hat{\phi}|\beta) = 1 - \operatorname{erf}\left(\frac{2.5}{\sqrt{2}}\right). \tag{4.27}$$

And the probability of correct rejection is then

$$(\hat{\beta}|\beta) = \operatorname{erf}\left(\frac{2.5}{\sqrt{2}}\right). \tag{4.28}$$

Probability functions of false alarm and correct rejection of the exception-to-background setting have been derived thus far. However, note that the probabilities of miss and hit cannot be calculated. The problem of one-class classification is a hard one [JD03]. In two-class classification, when models of both classes are available, a decision boundary is supported from both sides. Then we can take the integrals in both foreground and background models for the different performance measures. Because in case of one-class classification only the background class is available, just one side of the boundary is supported. If nothing else, we can assume that we have some knowledge of the stochastic foreground process.

In order to be as fair as possible between the ML/MAP and exception-to-background settings, we can calculate miss and hit rates of the later on the basis that the underlying foreground model is the same in both cases. In this case, $p(\mathbf{I}_{\mathbf{x}}|\phi)$ is as uninformative as possible, i.e., $p(\mathbf{I}_{\mathbf{x}}|\phi) = \frac{1}{256}$. Note that although the pdf function is the same one that we used in the

ML/MAP setting, in this case it is only used at the evaluation of the method, instead of also using it at the decision making. Intuitively, this will lead to better results in ML/MAP compared to exception-to-background as we will show at the end of this section.

The probability of miss ($P(\hat{\beta}|\phi)$) can be calculated similarly as in (4.22), i.e., we can take the integral of the foreground probability density function by the integration interval, which in this case corresponds to $\mathbf{I}_{\mathbf{x}}^{\diamond} \in [-2.5\sigma_{\mathbf{x}}, 2.5\sigma_{\mathbf{x}}]$.

Then, $P(\hat{\beta}|\phi)$ is

$$P(\hat{\beta}|\phi) = \frac{2 \cdot 2.5\sigma_{\mathbf{x}}}{256}. \tag{4.29}$$

And therefore the probability of correct detection ($P(\hat{\phi}|\phi) = 1 - P(\hat{\beta}|\phi)$) is

$$P(\hat{\phi}|\phi) = 1 - \frac{2 \cdot 2.5\sigma_{\mathbf{x}}}{256}. \tag{4.30}$$

Finally, this is the confusion matrix for the exception-to-background setting:

$$\begin{pmatrix} P(\hat{\phi}|\phi) & P(\hat{\phi}|\beta) \\ P(\hat{\beta}|\phi) & P(\hat{\beta}|\beta) \end{pmatrix}_{\text{exc.-to-back.}} = \begin{pmatrix} 1 - \frac{2 \cdot 2.5\sigma_{\mathbf{x}}}{256} & 1 - \operatorname{erf}\left(\frac{2.5}{\sqrt{2}}\right) \\ \frac{2 \cdot 2.5\sigma_{\mathbf{x}}}{256} & \operatorname{erf}\left(\frac{2.5}{\sqrt{2}}\right) \end{pmatrix}. \tag{4.31}$$

In Figure 4.3 we show the probabilities of false alarm and miss for both the exception-to-background and ML/MAP cases for different values of $\sigma_{\mathbf{x}}$. From the figure it seems to be clear that the ML/MAP setting performs better than exception-to-background for all cases.

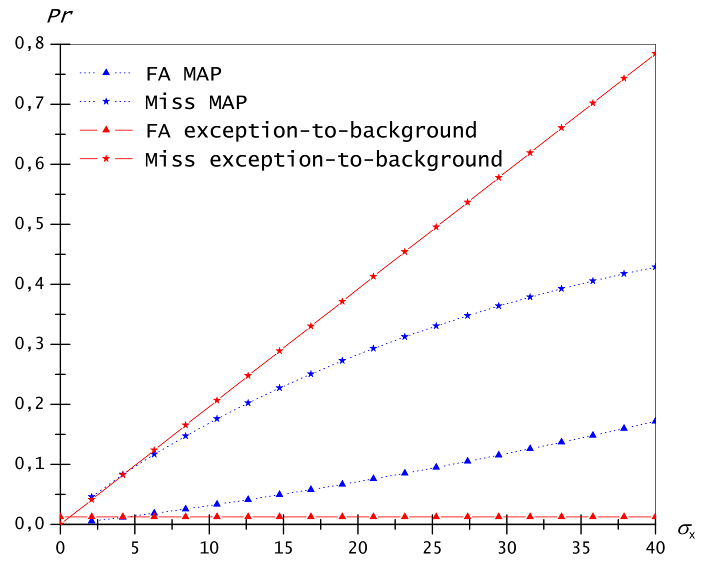
In the following, we prove that exception-to-background is always equal or worse than ML/MAP in terms of error probability. In fact, we provide evidence that the commonly used threshold $2.5\sigma_{\mathbf{x}}$ is not optimal in an exception-to-background setting. Indeed, without knowledge of the foreground process, it should only be assumed that the likelihood is uniform for all values. Under this assumption, the threshold that provides lower error probability should be equal to that one used in a Bayesian classification with a uniform foreground model.

If priors of background and foreground are the same, we have the following error probability in the exception-to-background setting:

$$\frac{1}{2}P(\hat{\phi}|\beta) + \frac{1}{2}P(\hat{\beta}|\phi) = \frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{T}{\sqrt{2}\sigma_{\mathbf{x}}}\right)\right) + \frac{1}{2}\left(\frac{2T}{256}\right), \tag{4.32}$$

where $\pm T$ are the thresholds used in the classification stage (typically set to $\pm 2.5\sigma_{\mathbf{x}}$). Note that the integration intervals used to derive (4.32) have been assigned assuming that $T > -T$ and, thus, the function is valid only in the interval $T > 0$. Also note that the function is concave in the interval, since its second derivative is always negative there (see Figure 4.4):

$$\frac{d^2\left(\frac{1}{2}\left(1 - \operatorname{erf}\left(\frac{T}{\sqrt{2}\sigma_{\mathbf{x}}}\right)\right) + \frac{1}{2}\left(\frac{2T}{256}\right)\right)}{dT^2} = \frac{-T}{\sqrt{2\pi}\sigma_{\mathbf{x}}^3}e^{-\left(\frac{T}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} < 0, \forall T \in]0, \infty]. \tag{4.33}$$



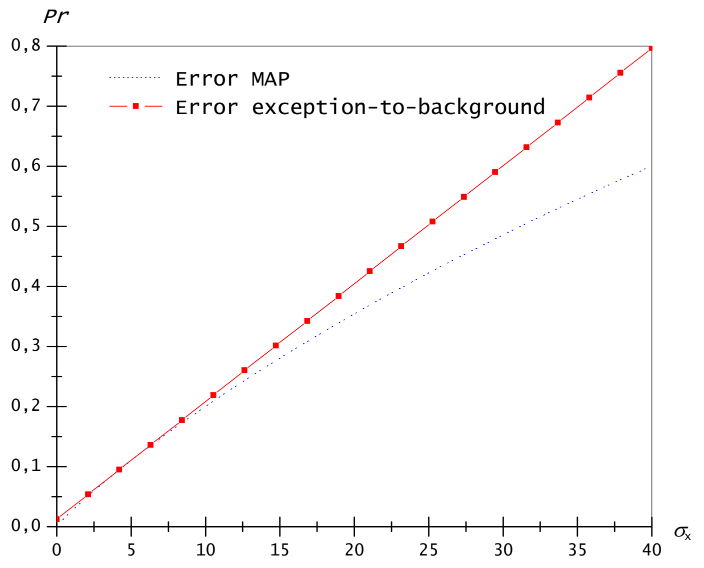


Figure 4.3: The figures compare the error probabilities of the ML/MAP setting and exception-to-background, with decision threshold $T = 2.5\sigma_{\mathbf{x}}$.
Note that the probability of false alarm in the exception-to-background setting is constant with respect to $\sigma_{\mathbf{x}}$, and therefore it is lower than in the ML/MAP setting for large values of $\sigma_{\mathbf{x}}$. On the contrary, the probability of miss in the exception-to-background setting increases linearly with $\sigma_{\mathbf{x}}$ while in the ML/MAP setting, the probability of miss tends to decline as the $\sigma_{\mathbf{x}}$ increases.
Altogether, after summing both probabilities (figure, bottom), the error rate in the exception-to-background setting is always larger than in the ML/MAP setting.

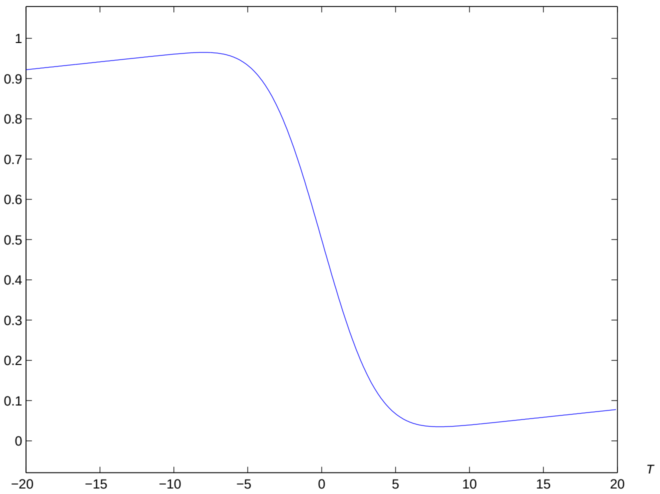


Figure 4.4: Error probability of the exception-to-background setting w.r.t decision threshold T, assuming equiprobable $P(\phi)$ and $P(\beta)$, and $\sigma_{\mathbf{x}} = 3$. See that there exists only one minimum between 0 and ∞.

Then, we can take the first derivative of the error probability to find the T that gives the minimum:

$$\frac{d\left(\frac{1}{2}\left(1-\operatorname{erf}\left(\frac{T}{\sqrt{2}\sigma_{\mathbf{x}}}\right)\right)+\frac{1}{2}\left(\frac{2T}{256}\right)\right)}{dT} = \frac{1}{256} + \frac{1}{\sqrt{2\pi}\sigma_{\mathbf{x}}} e^{-\left(\frac{T}{\sqrt{2}\sigma_{\mathbf{x}}}\right)^2} = 0. \tag{4.34}$$

This happens with $T = \sqrt{2}\sigma_{\mathbf{x}}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma_{\mathbf{x}}}{256}\right)}$, corresponding to the threshold that it is automatically used in the ML/MAP setting in (4.18).

In conclusion, an exception-to-background setting can perform as well as ML/MAP when the foreground models used in ML/MAP are uninformative. If this is the case, the decision threshold of the exception-to-background setting must be set to the same one used in the ML/MAP setting. Finally, note that although we have derived the equations for a background model assuming only one Gaussian per pixel, the generalization to an MoG is fairly simple since one only needs to incorporate the sum of other Gaussians. On the contrary, the model update step is a more difficult problem. Therefore, in the next section, we first consider the more general case of learning an MoG and the single Gaussian case is particularized at the end of the section.

4.4 Model Update

Once the classification determination has been made, it is important to update the foreground and background models by including the most recent input. There exist many different methods to update the models. The most important bit is how to seize the entry information, that is, how to measure the relevance of the observations when updating the models. For instance, in the approach of S&G, they propose to adapt the learning rate of background models to the likelihood that a pixel value belongs to the model, without further Bayesian justification.

Instead, we propose using a Bayesian scheme. The method we advocate is centered on the expectation maximization approach [DLR77]. Expectation maximization (EM) is an algorithm for finding maximum likelihood estimates of parameters in probabilistic models, where the model depends on unobserved latent variables. EM alternates between performing an expectation (E) step, which computes an expectation of the likelihood by including the latent variables as if they were observed, and a maximization (M) step, which computes the maximum likelihood estimates of the parameters by maximizing the expected likelihood found on the E step. The parameters found on the M step are then used to begin another E step, and the process is repeated.

It can be shown that an EM iteration does not decrease the observed data likelihood function. However, there is no guarantee that the sequence converges to a maximum likelihood estimator. For multimodal distributions, this means that an EM algorithm will converge to a local maximum (or saddle point) of the observed data likelihood function, depending on starting values. There is a variety of heuristic approaches for escaping a local maximum such as using several different random initial estimates, or applying simulated annealing [KCM83].

Although its final result gives a probability distribution over the latent variables (in the Bayesian style) together with a point estimate for the parameters (either a maximum likelihood estimate or a posterior mode), some theorists claim that EM is a partially non-Bayesian, maximum likelihood method. In fact, the fully Bayesian approach consists in treating the model parameters as another latent variable, in what is called *variational Bayes*. However, in this work we have chosen to use the most common EM approach. The complete details of the *variational Bayes* approach are covered in [Att99].

In the following, we provide the basics of the EM algorithm. We derive the equations for learning the parameters of an MoG background model, considering a uniform foreground model. This is the main difference w.r.t. the other approaches, that simply maximize the background model without considering the foreground process. Using an MoG in the background makes it possible to compare EM to the update scheme of the S&G method. In addition, the derivation of the single Gaussian background model can be simply obtained as a particularization of the MoG case. This section will also help in the understanding of some of the assumptions that are implicit in the scheme of S&G. Moreover, we will show that the update step of the EM method is directly tied to the ML/MAP classification described in §4.3, closing the Bayesian classification-update loop.

4.4.1 EM of a Gaussian Mixture and Uniform Functions

The mixture density parameter estimation problem is probably one of the most widely used applications of the EM algorithm.

In this case, we are assuming the following probabilistic background model:

$$\begin{aligned} p(\mathbf{I_x}|\beta) = \mathrm{MoG}_{\mathbf{x}}(\mathbf{I_x}) = \sum_{k=1}^{K} w_{\mathbf{x},k}\mathrm{G}_{\mathbf{x},k}(\mathbf{I_x}) = \\ \sum_{k=1}^{K} \frac{w_{\mathbf{x},k}}{(2\pi)^{D/2}\sqrt{|\Sigma_{\mathbf{x},k}|}} e^{-\frac{1}{2}(\mathbf{I_x}-\mu_{\mathbf{x},k})^T\Sigma_{\mathbf{x},k}^{-1}(\mathbf{I_x}-\mu_{\mathbf{x},k})}, \end{aligned} \tag{4.35}$$

where K is the total number of Gaussians used in each pixel, and where $w_{\mathbf{x},k}$ is the prior probability that a background pixel is represented by a certain mode k of the mixture ($\sum_{k=1}^{K} w_{\mathbf{x},k} = 1$). These priors are often referred as the weights of the Gaussians. Also note that the means and covariances are indexed w.r.t. a Gaussian k of the MoG in $\mathbf{x}$: $\Sigma_{\mathbf{x},k}$ and $\mu_{\mathbf{x},k}$.

The foreground model we are considering in the derivation of the EM method is a uniform function. Therefore, the likelihood function for a certain pixel $\mathbf{x}$ is:

$$p(\mathbf{I_x}|\theta) = P(\beta)\mathrm{MoG}_{\mathbf{x}}(\mathbf{I_x}) + P(\phi)\frac{1}{256^D}, \tag{4.36}$$

where the model parameters (θ) to estimate for the pixel are:

$$\theta = \left\{\mu_{\mathbf{x},1}, \cdots, \mu_{\mathbf{x},K}, \Sigma_{\mathbf{x},1}, \cdots, \Sigma_{\mathbf{x},K}, w_{\mathbf{x},1}, \cdots, w_{\mathbf{x},K}\right\}. \tag{4.37}$$

Now, let $\mathbf{I_x}[1], \cdots, \mathbf{I_x}[M]$ be a number of statistically independent observations at pixel location $\mathbf{x}$ drawn from either Gaussians $\mathrm{G}_{\mathbf{x},1}, \cdots, \mathrm{G}_{\mathbf{x},K}$ or from the uniform function. Using these observations, the problem to solve can be written as

$$\begin{aligned} \theta^\star &= \underset{\theta}{\operatorname{argmax}} \mathcal{L}(\theta|\mathbf{I_x}[1], \cdots, \mathbf{I_x}[M]) = \\ &= \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{I_x}[1], \mathbf{I_x}[2] \cdots, \mathbf{I_x}[M]|\theta) = \\ &= \underset{\theta}{\operatorname{argmax}}\, p(\mathbf{I_x}[1]|\theta)p(\mathbf{I_x}[2]|\theta) \cdots p(\mathbf{I_x}[M]|\theta), \end{aligned} \tag{4.38}$$

which is a difficult problem to optimize because it contains the products of sum of Gaussians and uniform distributions. However, if we consider that the observed data $\{\mathbf{I_x}[1], \cdots, \mathbf{I_x}[M]\}$ is incomplete, and postulate the existence of unobserved data items $\mathcal{Y} = \{y_1, y_2, \cdots, y_M\}$ whose value informs us which distribution is the cause of a certain observation, the expression to optimize is significantly simplified. That is, we assume that $y_i \in \{1, \cdots, Y\}$ for each i.

Concretely, let $y_i = k \in \{1, \cdots, Y-1\}$ if the ith sample was generated by the kth Gaussian and $y_i = Y$ if the sample was generated by the foreground process:

$$y_i = \begin{cases} 1, & \text{if the } i\text{-th sample was generated by the 1st Gaussian} \\ 2, & \text{if the } i\text{-th sample was generated by the 2nd Gaussian} \\ \cdots & \\ k, & \text{if the } i\text{-th sample was generated by the } k\text{-th Gaussian} \\ \cdots & \\ Y-1=K, & \text{if the } (Y-1)\text{-th sample was generated by the } (Y-1)\text{-th Gaussian} \\ Y, & \text{if the } i\text{-th sample was generated by the foreground process.} \end{cases} \tag{4.39}$$

If $\mathcal{Y}$ is introduced, then the expression of the likelihood function to maximize in the M step can be expressed as

$$\begin{aligned} \mathcal{L}(\theta|\mathbf{I}_{\mathbf{x}}[1],\cdots,\mathbf{I}_{\mathbf{x}}[M],\mathcal{Y}) &= \prod_{i=1}^{M} p(\mathbf{I}_{\mathbf{x}}[i],\mathcal{Y}|\theta) = \\ &= \prod_{i=1}^{M} p(\mathbf{I}_{\mathbf{x}}[i],y_i|\theta). \end{aligned} \tag{4.40}$$

The problem, of course, is that we do not know $\mathcal{Y}$. But if we assume that $\mathcal{Y}$ is a random vector, we can proceed. Remind that EM is about maximizing the expectation of the likelihood. After all, this random vector is introduced simply because it reduces the complexity of (4.38). Thus, instead of maximizing (4.38), EM maximizes the expectation of the likelihood in (4.40) with respect to the unobserved data items. And, as will be shown, the final expression to maximize will be much more simpler if $\mathcal{Y}$ is introduced.

First, we derive the equations of the E step. The E step, at the n-th iteration, corresponds to the estimation of which distribution was used (either one of the Gaussians or the uniform function), conditioned on the observation, using the values from the last maximization step.

We separate the E step into two, for each type of distribution,

$$P(y_i = j|\mathbf{I}_{\mathbf{x}}[i],\theta_n) = \begin{cases} P(y_i = k|\mathbf{I}_{\mathbf{x}}[i],\theta_n), & \text{for observations with Gaussian } k = 1\cdots K. \\ P(y_i = Y|\mathbf{I}_{\mathbf{x}}[i],\theta_n), & \text{for foreground observations.} \end{cases}$$

Lets first introduce the equation for the E step for all $k \in \{1 \cdots K\}$:

$$\begin{aligned}
P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n) &= \frac{p(y_i = k, \mathbf{I}_\mathbf{x}[i]|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} = \\
&= \frac{p(y_i = k, \mathbf{I}_\mathbf{x}[i]|\phi, \theta_n)P(\phi|\theta_n) + p(y_i = k, \mathbf{I}_\mathbf{x}[i]|\beta, \theta_n)P(\beta|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} = \\
&= \frac{p(y_i = k, \mathbf{I}_\mathbf{x}[i], \phi|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} + \frac{p(y_i = k, \mathbf{I}_\mathbf{x}[i], \beta|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} = \\
&= \frac{P(y_i = k|\mathbf{I}_\mathbf{x}[i], \phi, \theta_n)p(\mathbf{I}_\mathbf{x}[i], \phi|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} + \\
&\qquad + \frac{P(y_i = k|\mathbf{I}_\mathbf{x}[i], \beta, \theta_n)p(\mathbf{I}_\mathbf{x}[i], \beta|\theta_n)}{p(\mathbf{I}_\mathbf{x}[i]|\theta_n)} = \\
&= P(y_i = k|\mathbf{I}_\mathbf{x}[i], \phi, \theta_n)P(\phi|\mathbf{I}_\mathbf{x}[i], \theta_n) + \\
&\qquad + P(y_i = k|\mathbf{I}_\mathbf{x}[i], \beta, \theta_n)P(\beta|\mathbf{I}_\mathbf{x}[i], \theta_n), \qquad (4.41)
\end{aligned}$$

and since $P(y_i = k|\mathbf{I}_\mathbf{x}[i], \phi, \theta_n)$ is an impossible event because it is not possible to observe one of the Gaussian modes, given that the observation corresponds to the foreground process, then

$$P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n) = P(\beta|\mathbf{I}_\mathbf{x}[i], \theta_n)P(y_i = k|\mathbf{I}_\mathbf{x}[i], \beta, \theta_n), \qquad \forall k \in \{1 \cdots K\}.$$

The posterior probability $P(y_i = k|\mathbf{I}_\mathbf{x}[i], \beta, \theta_n)$ of a certain class k, given that the class belongs to one of the background modes, can be expressed using the Bayes theorem as the fraction between the likelihood of the k-th background mode over the sum of likelihoods of all the background modes:

$$\begin{aligned}
P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n) &= P(\beta|\mathbf{I}_\mathbf{x}[i], \theta_n)P(y_i = k|\mathbf{I}_\mathbf{x}[i], \beta, \theta_n) = \\
&= P(\beta|\mathbf{I}_\mathbf{x}[i], \theta_n)\frac{w^{(n)}_{\mathbf{x},k}G^{(n)}_{\mathbf{x},k}(\mathbf{I}_\mathbf{x}[i])}{\sum_{k=1}^{K} w^{(n)}_{\mathbf{x},k}\mathrm{G}^{(n)}_{\mathbf{x},k}(\mathbf{I}_\mathbf{x}[i])} = \\
&= P(\beta|\mathbf{I}_\mathbf{x}[i], \theta_n)\frac{w^{(n)}_{\mathbf{x},k}G^{(n)}_{\mathbf{x},k}(\mathbf{I}_\mathbf{x}[i])}{\mathrm{MoG}^{(n)}_\mathbf{x}(\mathbf{I}_\mathbf{x}[i])}, \qquad \forall k \in \{1 \cdots K\}, \qquad (4.42)
\end{aligned}$$

where $w^{(n)}_{\mathbf{x},k}$, $G^{(n)}_{\mathbf{x},k}$ and $\mathrm{MoG}^{(n)}_\mathbf{x}$ denote the weight, Gaussian likelihood and MoG likelihood in the n-th iteration, respectively.

And the posterior for foreground observations is:

$$P(y_i = Y|\mathbf{I}_\mathbf{x}[i], \theta_n) = P(\phi|\mathbf{I}_\mathbf{x}[i], \theta_n), \qquad (4.43)$$

which completes the E step, at the n-th iteration.

The EM procedure continues as follows. In the n-th iteration, $P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n)$ is computed for each i and each k using (4.42) and ($P(y_i = Y|\mathbf{I}_\mathbf{x}[i], \theta_n)$) is also computed for each i using (4.43). This is followed by the M-step described next.

In the M step, we want to maximize the expected likelihood of the joint event (4.40) w.r.t. the unknown data $\mathcal{Y}$. Log-likelihood is also often used instead of true likelihood because it leads to easier formulas, but it still attains its maximum at the same point as the likelihood. Then:

$$Q(\theta,\theta_n) = E_{\mathcal{Y}}\left[\log\prod_{i=1}^{M} p(\mathbf{I_x}[i],\mathcal{Y}|\theta)\,\middle|\,\mathbf{I_x}[1],\cdots,\mathbf{I_x}[M],\theta_n\right], \tag{4.44}$$

which we want to maximize with respect to θ. The right side of the equation can be re-written as

$$\begin{aligned}
Q(\theta,\theta_n) =& E_{\mathcal{Y}}\left[\log\prod_{i=1}^{M} p(\mathbf{I_x}[i],\mathcal{Y}|\theta)\,\middle|\,\mathbf{I_x}[1],\cdots,\mathbf{I_x}[M],\theta_n\right]\\
=& E_{\mathcal{Y}}\left[\sum_{i=1}^{M}\log p(\mathbf{I_x}[i],\mathcal{Y}|\theta)\,\middle|\,\mathbf{I_x}[1],\cdots,\mathbf{I_x}[M],\theta_n\right]\\
=& \sum_{i=1}^{M} E_{\mathcal{Y}}\left[\log p(\mathbf{I_x}[i],\mathcal{Y}|\theta)\,\middle|\,\mathbf{I_x}[i],\theta_n\right]\\
=& \sum_{i=1}^{M} E_{y_i}\left[\log p(\mathbf{I_x}[i],y_i|\theta)\,\middle|\,\mathbf{I_x}[i],\theta_n\right] = \text{(by definition of expectation)}\\
=& \sum_{i=1}^{M}\Big(\sum_{j=1}^{Y} P(y_i=j|\mathbf{I_x}[i],\theta_n)\log p(\mathbf{I_x}[i],y_i=j|\theta)\Big).
\end{aligned} \tag{4.45}$$

If we expand the probability of the joint event, we get

$$\begin{aligned}
Q(\theta,\theta_n) =& \sum_{i=1}^{M}\Big(\sum_{j=1}^{Y} P(y_i=j|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i=j,\theta)P(y_i=j|\theta)\right)\Big) =\\
=& \sum_{i=1}^{M}\Big(\sum_{k=1}^{K} P(y_i=k|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i=k,\theta)P(y_i=k|\theta)\right) +\\
& \underbrace{P(y_i=Y|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i=Y,\theta)P(y_i=Y|\theta)\right)}_{R}\Big).
\end{aligned} \tag{4.46}$$

The key thing to understand is that $\mathbf{I_x}$ and θ_n are constants and θ is a variable that we wish to adjust. That said, note that maximization of (4.46) can be further simplified since the right part of the equation (R) does not vary with different values of θ. The main reason for this is that we have decided to choose a uniform function for the foreground process without adjustable parameters. Therefore, $p(\mathbf{I_x}[i]|y_i=Y,\theta) = p(\mathbf{I_x}[i]|\phi)$ and $P(y_i=Y|\theta) = P(\phi)$[1].

Then,

$$\begin{aligned}
&\theta_{n+1} = \underset{\theta}{\operatorname{argmax}}\, Q(\theta,\theta_n) =\\
&\qquad \underset{\theta}{\operatorname{argmax}} \sum_{i=1}^{M}\sum_{k=1}^{K} P(y_i=k|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i=k,\theta)P(y_i=k|\theta)\right).
\end{aligned} \tag{4.47}$$

[1] $P(y_i=Y|\theta) = \sum_{\mathbf{I_x}=0}^{256^D-1} P(y_i=Y|\theta,\mathbf{I_x})p(\mathbf{I_x}) = \sum_{\mathbf{I_x}=0}^{256^D-1}\frac{P(y_i=Y)\frac{1}{256^D}}{p(\mathbf{I_x})}p(\mathbf{I_x}) = P(y_i=Y) = P(\phi)$.

The problem to solve is therefore:

$$\frac{dQ(\theta,\theta_n)}{d\theta} = \frac{d}{d\theta}\sum_{i=1}^{M}\sum_{k=1}^{K} P(y_i = k|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i = k,\theta)P(y_i = k|\theta)\right) = 0. \tag{4.48}$$

We have the following constraint:

$$\sum_{k=1}^{K} P(y_i = k|\theta) = P(\beta). \tag{4.49}$$

If we add a Lagrange multiplier with the constraint, we get

$$\begin{aligned}\frac{dQ(\theta,\theta_n)}{d\theta} &= \frac{d}{d\theta}\Bigg(\frac{d}{d\theta}\sum_{i=1}^{M}\sum_{k=1}^{K} P(y_i = k|\mathbf{I_x}[i],\theta_n)\log\left(p(\mathbf{I_x}[i]|y_i = k,\theta)P(y_i = k|\theta)\right)\\ &\quad -\lambda(\sum_{k=1}^{K} P(y_i = k|\theta) - P(\beta))\Bigg),\end{aligned} \tag{4.50}$$

which gives the following system of equations:

- The new estimate for the mean, using some differentiation rules (see [KML04, chapter3] and [DLR77] for similar expression derivations) is:

$$\begin{aligned}\frac{dQ(\theta,\theta_n)}{d\mu_{\mathbf{x},k}} &= 0 \Rightarrow\\ \mu^{\star}_{\mathbf{x},k} &= \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n)\mathbf{I_x}[i]}{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n)}.\end{aligned} \tag{4.51}$$

- New estimate for covariance:

$$\begin{aligned}\frac{dQ(\theta,\theta_n)}{d\Sigma^{(}_{\mathbf{x},k}n)} &= 0 \Rightarrow\\ \Sigma^{\star}_{\mathbf{x},k} &= \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n)(\mathbf{I_x}[i]-\mu^{\star}_{\mathbf{x},k})(\mathbf{I_x}[i]-\mu^{\star}_{\mathbf{x},k})^T}{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n)}.\end{aligned} \tag{4.52}$$

- And finally, new estimate for $P(y_i = k|\theta)$:

$$\begin{aligned}\frac{dQ(\theta,\theta_n)}{dP(y_i = k|\theta)} &= 0 \Rightarrow\\ P(y_i = k|\theta) &= \frac{1}{\lambda}\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n).\end{aligned} \tag{4.53}$$

If we insert into the equation above the constraint, then

$$\begin{aligned}\sum_{k=1}^{K} P(y_i = k|\theta) &= \sum_{k=1}^{K}\frac{1}{\lambda}\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n) = P(\beta) \Rightarrow\\ \lambda &= \frac{1}{P(\beta)}\sum_{k=1}^{K}\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i],\theta_n).\end{aligned} \tag{4.54}$$

Inserting λ into our estimate:

$$P(y_i = k|\theta)^\star = \frac{P(\beta)\sum_{i=1}^{M} P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n)}{\sum_{k=1}^{K}\sum_{i=1}^{M} P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n)} = \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I}_\mathbf{x}[i], \theta_n)}{M}, \tag{4.55}$$

where it has to be noted that priors $P(y_i = k|\theta)$ and $w_{\mathbf{x},k}$ are related to each other as follows:

$$w_{\mathbf{x},k} = \frac{P(y_i = k|\theta)}{P(\beta)}. \tag{4.56}$$

The estimates $\mu^\star_{\mathbf{x},k}$ (4.51), $\Sigma^\star_{\mathbf{x},k}$ (4.52) and $P(y_i = k|\theta)^\star$ (4.55) now become our θ_{n+1}, to be used in the next estimation step (4.42).

4.4.1.1 Discussion

The derivation of the equations above is different from the classical derivation that considers only a Gaussian mixture without a foreground model. In the problem presented we needed to include the uniform pdf which makes the mathematical formulation slightly different from the classical form. We have decided to include the derivation of the equations as a reference to the reader without stopping at the details. There are several works which describe the particularities of the mathematics of the EM derivation of a Gaussian mixture. Even though the problem is different from what we have described, it is still worth it to consult the classical problem for an in-depth description of the EM problem. Some good classical references are [Bil98, DLR77, MK97, MP00] among others.

4.4.2 Online Expectation Maximization

Note that one of the best characteristics of the approach of Stauffer and Grimson is that the background model is updated instead of fully recomputed at every time, which makes their algorithm very fast. However, equations (4.51), (4.52) and (4.55) assume a fixed number of observations M. In other words, EM, as presented so far, is inherently offline. It requires multiple passes through the data set. As the data set grows, so does the computation per iteration of EM. This limitation is a common limitation of the ordinary EM algorithm. A practical online implementation that is capable of foreground segmentation of each frame as it is acquired has to re-estimate all the parameters incrementally from each new sample.

In the following, we adapt the derivations of offline EM to online EM. As we have shown, the

M-step updates the model parameters in the n-th iteration as follows:

$$\begin{aligned} P(y_i = k|\theta)^\star &= \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i], \theta_n)}{M} \\ \mu_{\mathbf{x},k}^\star &= \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i], \theta_n)\mathbf{I_x}[i]}{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i], \theta_n)} \\ \Sigma_{\mathbf{x},k}^\star &= \frac{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i], \theta_n)(\mathbf{I_x}[i] - \mu_{\mathbf{x},k}^\star)(\mathbf{I_x}[i] - \mu_{\mathbf{x},k}^\star)^T}{\sum_{i=1}^{M} P(y_i = k|\mathbf{I_x}[i], \theta_n)}. \end{aligned} \tag{4.57}$$

The main idea behind an online update scheme is that last observation M is used to feed the last iteration n. Then, given a certain instant t, $t = M = n$:

$$\begin{aligned} P(y_i = k|\theta)^\star = P(y_i = k|\theta)[t] &= \frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)}{t} \\ \mu_{\mathbf{x},k}^\star = \mu_{\mathbf{x},k}[t] &= \frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)\mathbf{I_x}[i]}{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)} \\ \Sigma_{\mathbf{x},k}^\star = \Sigma_{\mathbf{x},k}[t] &= \frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)(\mathbf{I_x}[i] - \mu_{\mathbf{x},k}^\star)(\mathbf{I_x}[i] - \mu_{\mathbf{x},k}^\star)^T}{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)}. \end{aligned} \tag{4.58}$$

We start deriving the online version of the priors:

$$\begin{aligned} P(y_i = k|\theta)[t] = &\frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)}{t} = \\ = &\frac{\left(\sum_{i=1}^{t-1} P(y_i = k|\mathbf{I_x}[i], \theta_t)\right) + P(y_i = k|\mathbf{I_x}[t], \theta_t)}{t}. \end{aligned} \tag{4.59}$$

Assuming the posterior probabilities of the old data are forgotten, they do not need to be re-estimated, and the sum can be replaced by the old prior estimate

$$P(y_i = k|\theta)[t] \simeq \frac{(t-1)P(y_i = k|\theta)[t-1] + P(y_i = k|\mathbf{I_x}[t], \theta_t)}{t}, \tag{4.60}$$

and after simplification of the expression:

$$P(y_i = k|\theta)[t] \simeq P(y_i = k|\theta)[t-1] + \frac{1}{t}\left(P(y_i = k|\mathbf{I_x}, \theta_t) - P(y_i = k|\theta)[t-1]\right), \tag{4.61}$$

where $\frac{1}{t}$ is known as the learning speed. The learning speed is often left constant and represented with $\alpha = \frac{1}{t}$. The idea behind using a constant value is that the system prioritizes those last values within a sliding window of $t = T$ samples.

We operate similarly with the mean vector and covariance matrix. The mean vector can be expressed as:

$$\mu_{\mathbf{x},k}[t] = \frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)\mathbf{I_x}[i]}{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)} = \frac{\sum_{i=1}^{t} P(y_i = k|\mathbf{I_x}[i], \theta_t)\mathbf{I_x}[i]}{tP(y_i = k|\theta)[t]}, \tag{4.62}$$

which we can break into two terms as we did with the priors in (4.59):

$$\mu_{\mathbf{x},k}[t] = \frac{\left(\sum_{i=1}^{t-1} P(y_i = k|\mathbf{I_x}[i], \theta_t)\mathbf{I_x}[i]\right) + P(y_i = k|\mathbf{I_x}[t], \theta_t)\mathbf{I_x}[t]}{tP(y_i = k|\theta)[t]}. \tag{4.63}$$

Here again, we assume the posterior probabilities of the old data frozen:

$$\mu_{\mathbf{x},k}[t] \simeq \frac{(t-1)P(y_i = k|\theta)[t-1]\mu_{\mathbf{x},k}[t-1] + P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t)\mathbf{I}_\mathbf{x}[t]}{tP(y_i = k|\theta)[t]}, \tag{4.64}$$

and, using (4.61) to re-express $P(y_i = k|\theta)[t-1]$ in the equation above, we finally obtain:

$$\mu_{\mathbf{x},k}[t] = \mu_{\mathbf{x},k}[t-1] + \frac{P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t)}{t}\left(\frac{\mathbf{I}_\mathbf{x}[t] - \mu_{\mathbf{x},k}[t-1]}{P(y_i = k|\theta)[t]}\right). \tag{4.65}$$

The derivation of the online update equation of the covariance matrix is exactly the same as with the mean.

Finally, the derivation of the online version of EM considering both a foreground and background model is:

$$\begin{aligned}
&P(y_i = k|\theta)[t] = P(y_i = k|\theta)[t-1] + \alpha\left(P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t) - P(y_i = k|\theta)[t-1]\right)\\
&\mu_{\mathbf{x},k}[t] = \mu_{\mathbf{x},k}[t-1] + \alpha P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t)\left(\frac{\mathbf{I}_\mathbf{x}[t] - \mu_{\mathbf{x},k}[t-1]}{P(y_i = k|\theta)[t]}\right)\\
&\Sigma_{\mathbf{x},k}[t] = \Sigma_{\mathbf{x},k}[t-1] +\\
&\qquad + \alpha P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t)\left(\frac{(\mathbf{I}_\mathbf{x}[t] - \mu_{\mathbf{x},k}[t-1])(\mathbf{I}_\mathbf{x}[t] - \mu_{\mathbf{x},k}[t-1])^T - \Sigma_{\mathbf{x},k}[t-1]}{P(y_i = k|\theta)[t]}\right).
\end{aligned} \tag{4.66}$$

And if only one Gaussian per pixel is used:

$$\begin{aligned}
&\mu_{\mathbf{x},k}[t] = \mu_\mathbf{x}[t-1] + \frac{\alpha P(\beta|\mathbf{I}_\mathbf{x}[t], \theta_t)}{P(\beta)}\left(\mathbf{I}_\mathbf{x}[t] - \mu_{\mathbf{x},k}[t-1]\right)\\
&\Sigma_{\mathbf{x},k}[t] = \Sigma_{\mathbf{x},k}[t-1] +\\
&\qquad + \frac{\alpha P(\beta|\mathbf{I}_\mathbf{x}[t], \theta_t)}{P(\beta)}\left((\mathbf{I}_\mathbf{x}[t] - \mu_\mathbf{x}[t-1])(\mathbf{I}_\mathbf{x}[t] - \mu_\mathbf{x}[t-1])^T - \Sigma_\mathbf{x}[t-1]\right),
\end{aligned} \tag{4.67}$$

where $P(\beta)$ is a constant prior value.

4.4.2.1 Discussion

In the scheme proposed, the background models are updated using all the color values that are observed along the time and the system is in charge of automatically weighting the contribution of each sample. Note that from (4.42), it follows that $P(y_i = k|\mathbf{I}_\mathbf{x}[t], \theta_t) = P(\beta|\mathbf{I}_\mathbf{x}[t], \theta_t)P(y_i = k|\mathbf{I}_\mathbf{x}[t], \beta, \theta_t)$, meaning that the system first determines the background probability of an observation (using MAP), and then the probability of a certain mode, assuming that the system has observed a color value corresponding to the background. Thus, implicitly, the process of pixel model update is making use of the MAP classification setting. This permits obtaining better background models and, therefore, also better classifications.

In summary, the EM-MAP setting that has been proposed gives an integrated Bayesian explanation of the whole classification/update loop of the foreground segmentation task. This can be more clearly observed by inspecting (4.67), where only one Gaussian is used. Since the term $P(y_i = k|\mathbf{I_x}[t], \beta, \theta_t)$ does not appear in the equations, the expressions are simpler. Note that in this case, the mean vector and covariance matrix of a pixel $\mathbf{x}$ are simply updated proportionally to the background probability of the observed value $\mathbf{I_x}[t]$ at the instant t. Contrarily, in other systems [KB01, MRG99, NH98, Now91, Tra91], the background models are updated using only the color observations that have been classified as background. In these other approaches, the probability of the background process never intervenes during the model update step.

Concluding, the advantages of the proposed scheme can be outlined as follows:

1. Background observations are not pre-classified before being used to update the model. In the system, all pixel observations are employed to update the models. In this respect, the system is simpler to use.
2. Only those observations with high background probability contribute more. That is, the speed of adaptation of the background models is proportional to the lack of uncertainty of the background observation. Contrarily, in the systems that only maximize the likelihood of the background, all uncertain observations that are erroneously classified as background contribute to wrongly update a model *at full speed*.

The details of the robustness of S&G algorithm have been explained throughout this thesis. In the following lines, we compare both approaches, showing the assumptions that S&G makes and how it compares to our approach.

Our approach differs from the approach of S&G in that we use probabilities instead of likelihoods. Let us briefly remind the equations used during the update stage in S&G approach:

$$w_{\mathbf{x},k}[t] = \begin{cases} w_{\mathbf{x},k}[t-1] + \alpha(1 - w_{\mathbf{x},k}[t-1]), & \text{if matched} \\ (1-\alpha)w_{\mathbf{x},k}[t-1], & \text{if not matched} \end{cases}$$

$$\mu_{\mathbf{x},k}[t] = (1-\rho_{\mathbf{x},k})\mu_{\mathbf{x},k}[t-1] + \rho_{\mathbf{x},k}\mathbf{I_x}[t] \tag{4.68}$$

$$\sigma^2_{\mathbf{x},k}[t] = (1-\rho_{\mathbf{x},k})\sigma^2_{\mathbf{x},k}[t-1] + \rho_{\mathbf{x},k}\left(\mathbf{I_x}[t] - \mu_{\mathbf{x},k}[t-1]\right)^T\left(\mathbf{I_x}[t] - \mu_{\mathbf{x},k}[t-1]\right), \tag{4.69}$$

where $\rho_{\mathbf{x},k}$ is the adaptation learning rate used in Gaussian k and pixel $\mathbf{x}$: $\rho_{\mathbf{x},k} \propto G_{\mathbf{x},k}(\mathbf{I_x})$.

In the approach of Stauffer and Grimson, it is not possible to obtain the probability that a sample belongs to a certain mode. Therefore, the update speed can only be linked to the likelihood, while, in fact, it should have been linked to its probability, as we have shown. This leads to some problems: $\rho_{\mathbf{x},k}$ can be too small due to the likelihood factor, leading to too slow adaptations in the means and covariance matrices. In fact, it is common to use constant values of $\rho_{\mathbf{x},k}$ to solve this issue.

In addition, the parameters used in our system seem to be more intuitive. In fact, in our case, we only need to choose the priors of foreground and background and the number of Gaussians that will be used. On the approach of S&G one has to decide the number of Gaussians, the threshold T, corresponding to the minimum prior probability that the background is in the scene. Also, the user has to select the adaptation speed α. And in practical scenarios, the user also needs to find a suitable value for $\rho_{\mathbf{x},k}$. Since the system proposed by Stauffer and Grimson is partially heuristic, it is difficult to rationally obtain good values without spending some time experimenting.

Another of the key advantages of our proposal is that the system lets the user set the prior probabilities of foreground and background. Indeed, the speed of adaptation of the models depends on the priors. Of course, one can simply set a fixed value. However, one can also select the priors based on some external reasoning or higher level sources of information as we discuss below.

4.4.3 Introducing High Level Information

From (4.66), it follows that background models are updated based on *how probable is that the observed sample belongs to the background.* To do so, we use the probabilities that the Bayesian MAP setting estimates.

Our scheme offers some more flexibility, though. For instance it is possible to use averaged probabilities $P(\beta|\mathbf{I}_\mathbf{x})$ from neighboring pixels so that the update step is less prone to the camera's thermal noise. One could also apply kernel windows over the map of pixel probabilities giving more weight to central pixels and less to their neighbors. A 2D Gaussian kernel would perfectly fit for this case.

Also, this scheme permits using higher dimensional information. Indeed, 3D foreground information can be more precise than that one which is only 2D-based. In §6.2, on page 101, we describe how to create a 3D probabilistic map from 2D probabilities in a multi-camera environment. Since the learning rate in our scheme is directly tied to the background probability, it is very simple to switch the learning rate to other more informed probabilistic values such as the projection of the 3D maps in this case. Some of these aspects will be described in the following chapters of this thesis.

4.4.4 Replacing Gaussian Surfaces

We finish the section with some further remarks on the different methods that can be used for avoiding local maxima in the EM algorithm and to adapt to sudden scene changes.

In the approach of S&G, those Gaussian surfaces that repeatedly do not represent observations in the recent past are replaced with new ones. To do so, every time that a new observation is not clustered with any Gaussian, the surface with lowest weight is discarded and a new Gaussian is created. The new Gaussian takes the observed color as its mean. A large default value is given to its initial variance and a small value is used for its initial weight. This approach allows the system to adapt to sudden scene changes. However, it has inconveniences too, since static foreground observations are eventually merged into the background (see §2.1.1.2, on page 13).

Of course, this can be the intended behavior in certain applications, such as in parking lots, where the cars that have been parked can be considered as background after a few frames. On the contrary, in other situations such as in meeting rooms, this may not be the intended behavior. Note that most likely, it will not be desirable that the people quietly attending a meeting eventually get merged into the background.

In our testing scenarios we have often decided not to let the system to renew its Gaussian surfaces, since we usually have not been interested in letting the static foreground merge into the background. Consequently the Gaussian modes of the background models have to be initialized using a few seconds of recordings without foreground objects in the scene. After this initial period, the parameters of the MoGs are simply updated using the online EM equations described before. The details of the process are as follows:

Initially, a set of Gaussian surfaces with random parameters are initialized. Then, during the training interval without foreground objects, the background probability of each observation is examined. If the probability is sufficiently low, a new surface is created and the surface with lowest prior is discarded. This process is repeated for each observation in the training period (typically 50 frames), letting the system to escape from local maxima. After the training period, the system is locked in the sense that we do not let replacing any surface. At this stage, the online EM equations are employed without further modifications, therefore assuming a controlled environment.

To allow the proposed system to behave similarly as the system of S&G, the online update mechanism should be configured to operate always as in the training period. In this configuration, the minimum background probability that triggers the creation of new Gaussians will have to be set for each scenario.

4.5 Results

In §4.3.2, we presented the theoretical improvements of our classification setting over the well-known algorithm by Stauffer and Grimson. Figure 4.3 was also presented for an easier visual verification of these improvements. Actually, these results were presented assuming the same background models with identical parameters for both approaches while, in fact, the EM-MAP approach uses better learned models due to the improved update scheme presented in the previous section.

In addition to the mentioned theoretical results, in this section we present some images obtained in a real world scenario. Of course, results on real world scenarios depend on the set up, recording conditions, and so on. The images that we present only have to be understood as the way to perform a visual inspection of how one algorithm compares to the other in a specific environment.

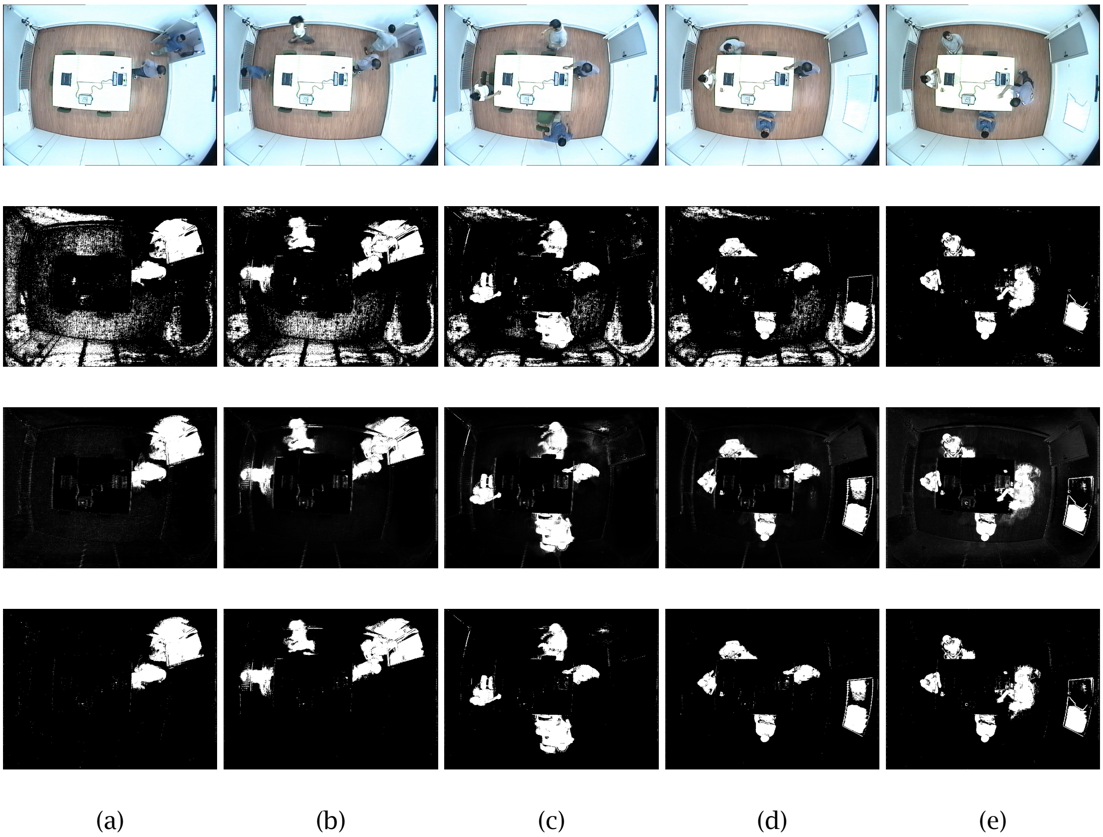


Figure 4.5: Images corresponding to frames 0 to 400. Second row shows the foreground segmentation using the S&G approach. Third and fourth row shows the probability and final segmentation of the MAP approach, respectively. When the door is opened (column (a)), there is a sudden change of illumination all aver the room. Note how the illumination change affected the S&G approach. Columns (d) and (e) show a new foreground appearance on the right part of the images corresponding to a background change due to a new slide being shown with the projector beamer placed over the table.

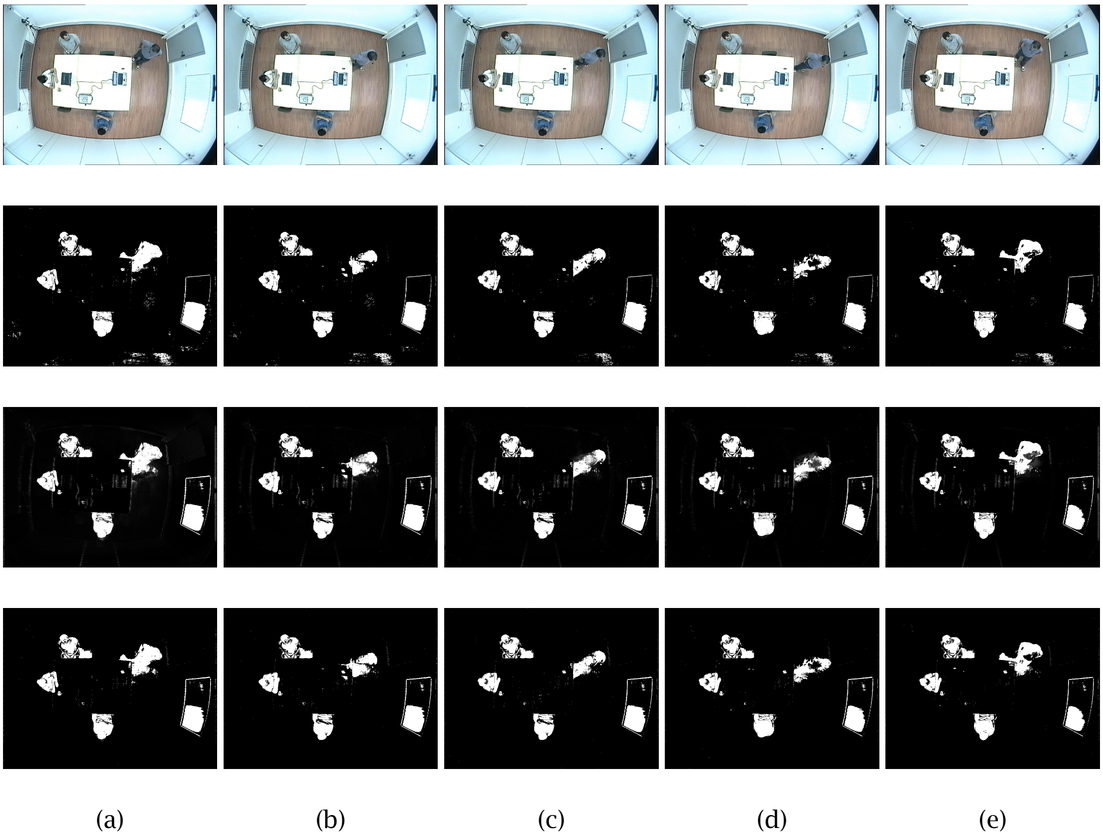


Figure 4.6: Images corresponding to frames 500 to 1000. Second row shows the foreground segmentation using the S&G approach. Third and fourth row correspond to the probability and final segmentation of the MAP approach, respectively. Once the illumination has been stabilized, then both the MAP and S&G approaches behave similarly. However, note that MAP scheme performs better than the algorithm of S&G in the walls where some of the errors produced by the sudden illumination change that took place in frame 0 has not been recovered yet in the approach of S&G.

In this case, the experiment has been performed in the smart-room of our lab. In the scene that we present (see Figure 4.5 and Figure 4.6) some people open the door and enter the room. The video has been recorded at 25 fps from a camera that is located at the ceiling of the room. The figures show one of every 100 recorded frames. In particular, Figure 4.5 shows frames from 0 to 400 and Figure 4.6 shows images from 500 to 1000. The second row of images correspond to the segmented foreground employing the approach of S&G. The third row shows the foreground probability of the MAP approach after scaling the probability from $[0, 1]$ to $[0, 255]$. Finally, the fourth row shows the segmentation after thresholding the probability at $\frac{1}{2}$, corresponding to the pixels where its color observations have larger foreground probabilities than background probabilities. In both cases, only one Gaussian per pixel has been employed since this is enough for indoors situations. In both approaches, we have used a training period of 50 frames and we have not let the system to renew the Gaussians. In the experiments shown, we have used (4.67) as the online model update equations of our approach. And, since only one Gaussian is used, the parameters of the background model that have been updated in the approach of S&G have only been the mean and variance of each pixel, using (4.68) and (4.69), respectively.

In the images, note that there is a sudden change of illumination just in the moment when the door is opened (see Figure 4.5). The change of illumination makes the camera's auto gain to adapt to the new lighting conditions. Therefore, the brightness of the captured image is globally affected. The foreground segmentation performed using the S&G approach is clearly affected in this case even though we chose the value of ρ that worked best in this sequence ($\rho = 0.01$). On the other hand, the EM-MAP approach performs better in this case without the need to tweak parameters. The behavior of both approaches can be explained as follows. When the illumination changed, the value of each color observation was shifted from the mean of its corresponding Gaussian. In the EM-MAP approach the decision threshold between foreground and background was set at the crossing between the uniform function ($\frac{1}{256}$) representing the foreground and the Gaussian. This threshold proved to be more robust than using 2.5 standard deviations of the distribution with the shifted color values. In addition, the EM-MAP approach was fastest to adapt the models by immediately using the observations to update the models proportionally to their background probability.

Once the background models are updated to the new lighting conditions (see Figure 4.6) then both the MAP and the exception-to-background schemes do not differ as much as in period of sudden illumination change. However, note that the presented framework performs better than the algorithm of S&G in the walls where some of the errors produced by the sudden illumination change that took place in frame 0 (Figure 4.5) has not been recovered yet in the approach of S&G.

4.6 Conclusion

In this chapter, we have presented a novel scheme for effective Bayesian foreground segmentation that exploits the probability of the classification at the model update step.

The principal advantages and characteristics of the system can be summarized in the following points:

- The system supports MoGs for the background models, among other pdfs. Note that MoGs have been successfully used many times in the past in other frameworks. Thus, the system has been built over robust building blocks which have been adapted and improved in a Bayesian way.
- The system is not only able to classify pixel observations but to provide the probability of a given classification. These probabilistic values will be used in chapter 6 to build three-dimensional probabilistic maps. Moreover, the system also provides its probabilities of false alarm and miss, which will come in handy in chapter 7 to evaluate the error probabilities of a volumetric reconstruction method that employs foreground segmentations from multiple views.
- Finally, background models are updated proportionally to the probability that an observation belongs to the background. This correlation between model update and pixel probability can be exploited by using other more-informed probabilistic sources of information that are external to the pixel process. This finding will be used to bridge the gap between the planar and volumetric foreground detection tasks, unifying the algorithms presented in this chapter and the mentioned chapters 7 and 6.

Chapter 5

Planar Foreground Segmentation; Application to Object Detection and Tracking

THIS CHAPTER is devoted to two of the key research issues in computer vision - the detection and tracking of multiple objects in the cluttered dynamic scene - that underpin the intelligence aspects of advanced visual surveillance systems aiming at automated events detection and behavior analysis. We discuss two major contributions in resolving these problems within a systematic framework. First, for accurate object detection, an efficient and effective scheme is proposed to remove cast shadows / highlights based on a conditional morphological reconstruction. The detection is based on a hierarchical foreground segmentation method that characterizes and groups pixels into regions according the time they have remained in the same position of the scene. Second, for effective tracking, a temporal-template-based tracking scheme is introduced, using multiple descriptive cues (velocity, shape, color, etc.) of the 2D object appearance. A scaled Euclidean distance is used as the matching metric, and the template is updated using Kalman filters when a matching is found or by linear mean prediction in the case of occlusion. Extensive experiments are carried out on video sequences from various real-world scenarios.

5.1 Introduction

In recent years, there has been considerable interest in visual surveillance of a wide range of indoor and outdoor sites by various parties. This is manifested by the widespread and unabated deployment of CCTV cameras in public and private areas. In particular, the increasing connectivity of broadband wired and wireless IP networks, and the emergence of IP-CCTV systems with smart sensors, enabling centralized or distributed remote monitoring, have further fueled this trend. It is not uncommon nowadays to see a bank of displays in an organization showing the activities of dozens of surveillance sites simultaneously. However, the limitations and deficiencies, together with the costs associated with human operators in monitoring the overwhelming video sources, have created urgent demands for automated video analysis solutions. Indeed, the ability of a system to automatically analyze and interpret visual scenes is of increasing importance to decision making, offering enormous business opportunities in the sector of information and communications technologies.

In monitoring a visual scene that is cluttered and busy, the importance of detection and tracking of any number of moving objects of interest can never be overestimated. This is the central element of an object- based intelligent video surveillance system, of which the two types of application are:

- to allow real-time detection of unforeseen events that warrant the attention of security guards or law enforcement officers to take preventive actions [LFP98],
- to enable tagging and indexing of interesting (customer-defined) scene activities/statistics into a metadata database for rapid forensic analysis [PLP02].

In addition, object detection and tracking are the building blocks of higher-level vision-based or assisted event monitoring and management systems with a view to understanding the complex actions, interactions, and abnormal behaviors of objects in the scene. The range of applications include detection of criminal behaviors in banks [GMBT04], marketing data analysis in shopping malls [HF01, Sen02], and well-being monitoring at home [CGP$^+$04].

5.2 Surveillance Systems - Challenges

Vision-based surveillance systems can be classified in several different ways, depending on the conditions in which they are designed to operate:

- indoor, outdoor or airborne,
- the type and number of sensors,
- the objects and level of details to be tracked.

In this chapter we mainly focus on processing videos captured by a single fixed outdoor CCTV camera overlooking areas where there are a variety of vehicle and/or people activities. However, we also conduct further indoor experiments for the sake of showing the completeness of the detection and tracking method proposed.

There are typical number of challenges associated with the chosen scenario in a realistic surveillance application environment.

Natural cluttered background: A natural outdoor environment is usually noisy and difficult to characterize. The video sequences captured are also often subjected to a compression process such as MPEG or JPEG before being transmitted via a network or stored for analysis. This introduces coding-induced noise into the already noisy imaging sources. Indoor situations are easier to handle in this respect.

Dynamic background: The scene background is not normally a fixed structure, but often changes with time. In the case of a swaying tree or flag, each pixel in the background cannot be fully characterized by a single color since two or more different appearances could be alternating. Cross color interferences and the use of fluorescent fixtures can also introduce repetitive patterns in indoor situations.

Illumination changes: Outdoor surveillance systems suffer heavily from the change of weather conditions. Rain, sunset, sunrise, as well as floating clouds can have a dramatic impact on the scene illumination. Hence, they will degrade the performance of object detectors and trackers if these factors are not accommodated properly. Indoor situations suffer from similar problems. For instance, in office environments with windows, weather changes can also easily alter the global illumination of the room. The use of projector beamers during presentations is another illumination problem in indoor situations.

Occlusions: In typical indoor or outdoor scenes with many moving objects, occlusion is a crucial issue that needs special treatment. The occlusion can happen in the following cases:

- inter-object, where objects occlude each other: this problem becomes acute when two or more objects enter into the scene while occluding each other,
- thin scene structures: thin objects in the scene such as trees or streetlights can break a moving object into several (typically two) separate parts,
- large scene structures: because of large scene structures such as buildings, moving objects may disappear completely for a period of time, and then re-appear, e.g., a pedestrian walking behind a parked or moving van.

Object entries and exits: Before a newly detected object in the scene is confirmed, it is important to know first if this is a new entry, and if so, how it is going to be modeled. It is equally important to take a correct decision about how and when to delete an existing object after its track is lost from the scene for some time.

Shadows and highlights: These are more problematic when the tracking process is carried out in outdoor environments, as very strong shadows or long shadows, sometimes larger than the actual object, are not uncommon. In addition, there are two types of shadows that need different treatment:

- cast-shadows: these are areas in the background projected by an object in the direction of light rays, which can, without careful consideration, be easily taken as part of an object, causing difficulties to the ensuing object tracking and classification tasks,
- self-shadows: these are parts of the object that are not illuminated by direct light, which a simple shadow-removal procedure is likely to get rid of, resulting in an inaccurate object detection.

5.3 Related Work

These technical challenges, together with the ever-increasing demand of intelligent video surveillance applications, have led over recent years to extensive research activities that propose various new ideas, solutions and frameworks for robust object detection and tracking [HHD00, JS02]. Most adopt a type of *foreground segmentation* technique to firstly detect foreground pixels. A connected component analysis (CCA) is then usually followed to cluster and label the foreground pixels into separate meaningful blobs, from which some inherent appearance and motion features can be extracted. Finally, there usually is a blob-based tracking process aiming to find persistent blob correspondences between consecutive frames. In addition, most application systems also deal with the issues of object categorization or identification (and possibly detailed parts analysis) either before [HHD00] or after [ZA01] the tracking is established.

With regard to foreground segmentation, the reader is referred to §2.1, on page 5 and §4 on page 41 for a more complete description of the state-of-the-art in the field, as well as for our proposal, respectively.

One major issue after proper foreground segmentation concerns shadow detection and removal [CGPP03]. An effective shadow removal scheme should completely remove cast shadows, but should not distort a foreground object's shape by removing extremities or deleting possible self-shadows. The use of a color constancy model for shadow detection has been well studied by Horprasert et al [HHD99], assuming that the chromaticity of a shadowed region is preserved but its intensity decreases. However, in the case where shadow removal based on color properties alone may not be effective or color information is not available, variants of gradient information can be further exploited to fulfill the task [Bev03]. Combinations of multiple cues (e.g., color, normalized color, gradient) were also considered by Javed et al [JS02] and McKenna et al [MJD$^+$00]. Often, appropriate heuristic rules have to be adopted [Bev03, MJD$^+$00] in order to accurately recover the true shape of an object.

Regarding the matching method and the choice of suitable metrics, the inherent heterogeneous nature of the features extracted from the 2D blobs has motivated some researchers to use only a few features, e.g., the size and velocity [JS02] for motion correspondence, and the size and position with Kalman predictors [SG00b]. Others using more features conducted the matching in a hierarchical manner, e.g., in the order of centroid, shape, and then color as discussed by Zhou and Aggarwal [ZA01]. Note that if certain a priori factors are known, e.g., the type of an object to be tracked is a single person, then a more complex dynamic appearance model of the silhouette can be employed [HHD00]. Also, in Elgamal et al [EDHD99], the kernel density function was used to model the color distribution of an object to help detect and track individual persons who start to form a group and occlude each other; McKenna et al [MJD^{+}00] provides another relevant example where probabilistic object models were exploited. Furthermore, domain knowledge of a physical site can be built beforehand for more effective management of object entry and exit [Sta03] and for better handling the object occlusion issues in some applications [XE02].

In this chapter we describe an effective multi-object detection and tracking system in which a few novel ideas are introduced to deal with these challenging issues. This leads to the enhancement of several aspects of state-of-the-art object detection and tracking techniques. In particular, we employ a technique to suppress the falsely detected foreground pixels, mainly caused by video compression artifacts. A novel framework is introduced for effective cast shadows/ highlights removal while preserving the original object shape. A novel hierarchical segmentation module is also proposed to identify meaningful blobs, based on the similarities that neighboring pixels show in terms of the temporal persistence of the observed colors. An integrated matching strategy is discussed, using the scaled Euclidean distance metric, in which a feature set characterizing a foreground object is used simultaneously, taking care of both the scale and variance of each of the features. This matching method is not only robust (in the sense of tolerating sudden speed changes or direction changes), but also allows an easy inclusion of more extracted features, if necessary, leaving room for future enhancement. Figure 5.1 schematically depicts the block diagram of our proposed object detection and tracking system, which comprises two named major functional modules, each in turn containing a number of processing steps. The object classification module is included for completeness, though it will not be discussed here; interested readers are referred to Javed and Shah [JS02] or Zhou and Aggarwal [ZA01] for more information.

The chapter is structured as follows. In the next section, techniques for pixel-domain analysis, leading to segmented foreground object blobs, are discussed, with emphasis on the introduction of a novel shadow removal scheme. In section 5.5 the different aspects regarding how to group pixels into blobs are discussed. Section 5.6 is devoted to discussion of multi-object tracking, including the use of a temporal object template, the adoption of a parallel matching procedure and the partial occlusion handling. Experimental studies of the tracking system with various real-world test sequences are also discussed in this section. This chapter concludes in section 5.7 with a discussion of future research directions and possible system enhancements.

5.4 Foreground Pixels Extraction with Shadow Removal

As depicted in Figure 5.1, the first issue to be addressed is to extract scene pixels forming part of the foreground moving objects via foreground segmentation. We propose using any background model that allows a proper representation of the background scene undergoing

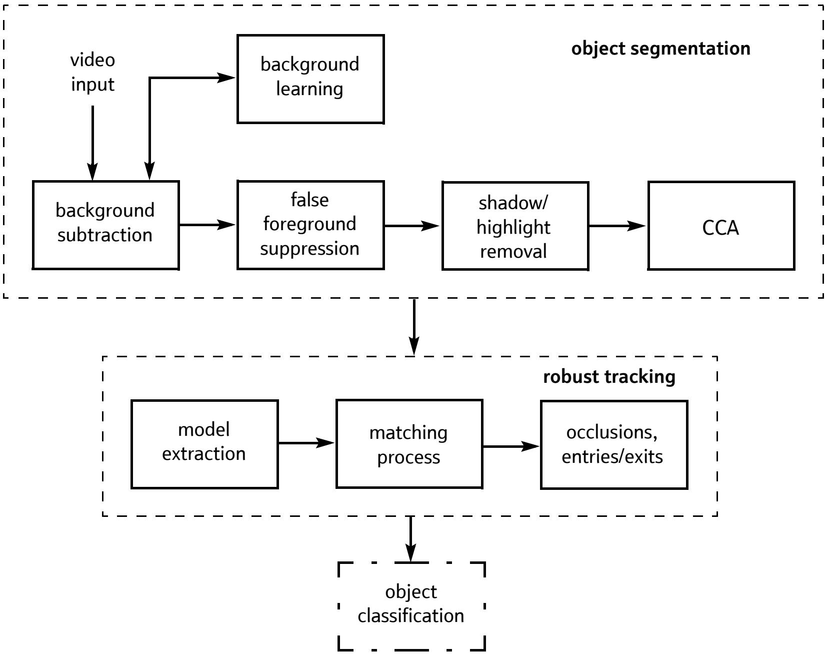


Figure 5.1: The schematic system block diagram showing the two main functional modules.

slow and smooth lighting changes and momentary and random variations such as trees or flags swaying in the wind. Gaussian mixtures(see §2.1 and §4) have proved to be fast and reliable in our experiments, but any other background models can be employed in the proposed scheme instead of them. Considering that the foreground pixels thus obtained are likely to suffer from false detections due to imaging and compression noise as well as camera jitter, a false-foreground-pixel-suppression procedure is introduced to alleviate this problem. The idea is that, for each pixel initially classified as a foreground pixel, the background models of its 8-connected neighboring pixels are examined. If the majority of them (> 5) agree that the pixel is a background pixel, then that pixel is considered as a false detection and removed from foreground. This technique has proved to alleviate some problems consisting in slight camera movements due to the combination of wind and not having the camera firmly attached.

5.4.1 A Novel Shadow / Highlight Removal Scheme

Once the foreground pixels are identified, a further detection scheme is applied to locate areas that are likely to be cast shadows or highlights. In the following, we discuss a novel scheme for effective shadow (and highlights) detection using both color and texture cues. Since in any shadow-removal algorithm misclassification errors often occur, resulting in distorted object shapes, the core of this scheme is the use of a technique capable of correcting these errors. The technique is based on a greedy thresholding followed by a conditional morphological dilation. The greedy thresholding removes all shadows together with some true foreground pixels. The conditional morphological dilation then aims to recover only those deleted true foreground pixels constrained within the original foreground mask. The working mechanism of this novel scheme is shown in Figure 5.2 and comprises the following four steps.

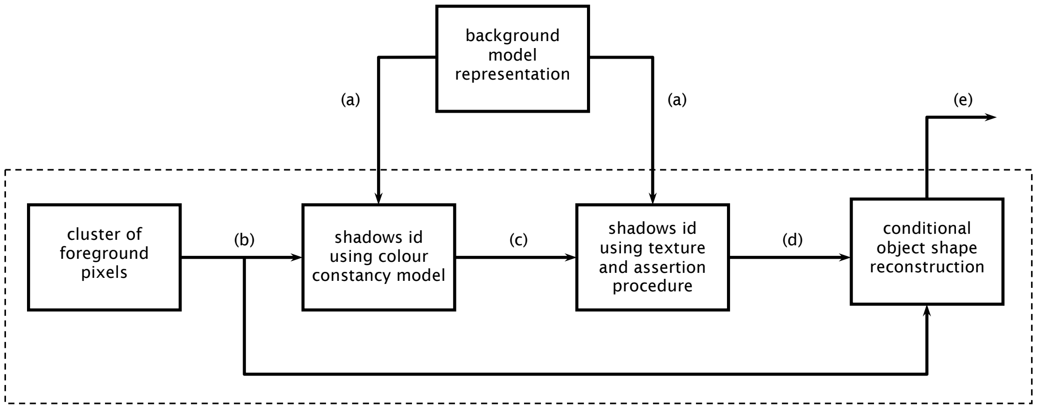

background model representation
(a)
(a)
(e)
cluster of foreground pixels
(b)
shadows id using colour constancy model
(c)
shadows id using texture and assertion procedure
(d)
conditional object shape reconstruction

Figure 5.2: The schematic diagram of the novel shadows / highlights removal approach made up of four main processing steps. The input and output of each block are as follows (a) the adaptive background image; (b) initial foreground segmentation result; (c) shadows / highlights removal using color constancy model; (d) the result after shadows assertion using gradient / texture information, generating a marker image; and (e) final reconstructed foreground regions.

Color-based detection: As the first step, a simplified version of the color constancy model introduced by Horprasert et al [HHD99] is employed. This model evaluates the variability in both brightness and color distortions in the RGB color space between the foreground pixels and the adaptive background. When using a Gaussian mixture, a background reference image can be obtained by taking the mean value of the Gaussian with highest weight (i.e., highest prior). If another background model is used, then the reference image can also be obtained by choosing the most likely value of the model. Possible shadows and highlights are then detected by certain thresholding decisions [LPX05b]. It was observed though that this procedure is less effective in cases where the objects of interest have similar colors to those of presumed shadows.

Texture-based detection: The same regions with or without cast shadows tend to retain similar texture (edge) properties despite the difference in illumination. To exploit this fact, a Sobel edge detector is used to compute the horizontal and vertical gradient for both the foreground pixels and their corresponding ones in the background reference image. For each pixel, the Euclidean distance with respect to the gradients is evaluated over a small neighborhood region, which is then employed to examine the similarity between the foreground and reference pixel. If the distance is less than a certain threshold, then a possible shadow pixel is suggested.

Assertion procedure: Based on the detection results from the above two steps, an assertion procedure is introduced, which confirms a pixel as belonging to foreground only if both the above two outputs agree. The output from this procedure is a seed *marker* image (as shown in Figure 5.4(c)), free of shadows and highlights.

Conditional object shape reconstruction: The above processing steps are designed to effectively remove cast shadows and highlights, though they also invariably delete some foreground object pixels (self-shadows), causing the distortion of a real object's shape. Therefore, a morphology-based conditional region reconstruction step is employed to restore each object's original shape from the *marker* image.

The mathematical morphology reconstruction filter uses the *marker* as the seed to rebuild an object inside the original *mask* image. In our case, the *marker* image (see Figure 5.4(c)) is a binary image in which a pixel is set to "1" when it corresponds to a foreground, not a cast shadow / highlight, pixel. On the other hand, the *mask* image (see Figure 5.4(b)) is also a binary image where a "1" pixel can correspond to a foreground pixel, or a cast shadow / highlight pixel, or speckle noise.

It is highly desirable that the *marker* image $\tilde{M}$ contains only real foreground object pixels, i.e., not any shadow / highlight pixels so that those regions will not be reconstructed. Therefore, the use of very aggressive thresholds is necessary in the color-based removal process to ensure that all the shadow / highlight pixels are removed. A speckle noise removal filter is also applied to suppress the remaining isolated noisy foreground pixels and to obtain a good quality *marker* image, $\tilde{M}$.

The speckle removal filter is also implemented using morphological operators as shown in (5.1):

$$\tilde{M} = M \cap (M \oplus N), \tag{5.1}$$

where M is the binary image generated after shadow removal and assertion process and N denotes the structuring element, shown in Figure 5.3, with its origin at the center.

The dilation operation in (5.1) identifies all the pixels that are four-connected to (i.e., next to) a pixel of M. Hence, $\tilde{M}$ identifies all the pixels that are in M and also have a four-connected neighbor, eliminating the isolated pixels in M.

As a result, only the regions not affected by noise which are clearly free of shadows / highlights (Figure 5.4(c)) are subject to the shape partial reconstruction process shown in (5.2):

$$R = M_s \cap (\tilde{M} \oplus SE), \tag{5.2}$$

where M_s is the *mask*, $\tilde{M}$ is the *marker* and SE is the structuring element whose size usually depends on the size of the objects of interest, although a 9×9 square element proved to work well in our tests.

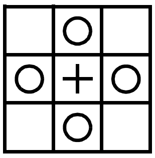


Figure 5.3: The 3 × 3 morphological structuring element used for speckle filtering. Note that the origin is not included.

To sum up, this process consists of a dilation of the *marker* image, followed by the intersection with the *mask* image. The underlying idea is that there should be a fairly large number of valid object pixels remaining after the shadow removal processing. These pixels are appropriate for leading the reconstruction of neighboring points as long as they form part of the silhouette in the original blob (prior to the shadow removal as in Figure 5.4(b)). The finally reconstructed blobs are shown in Figure 5.4(d). Figure 5.5 shows additional results in an indoor situation where many shadows appeared due to the illumination in the scene.

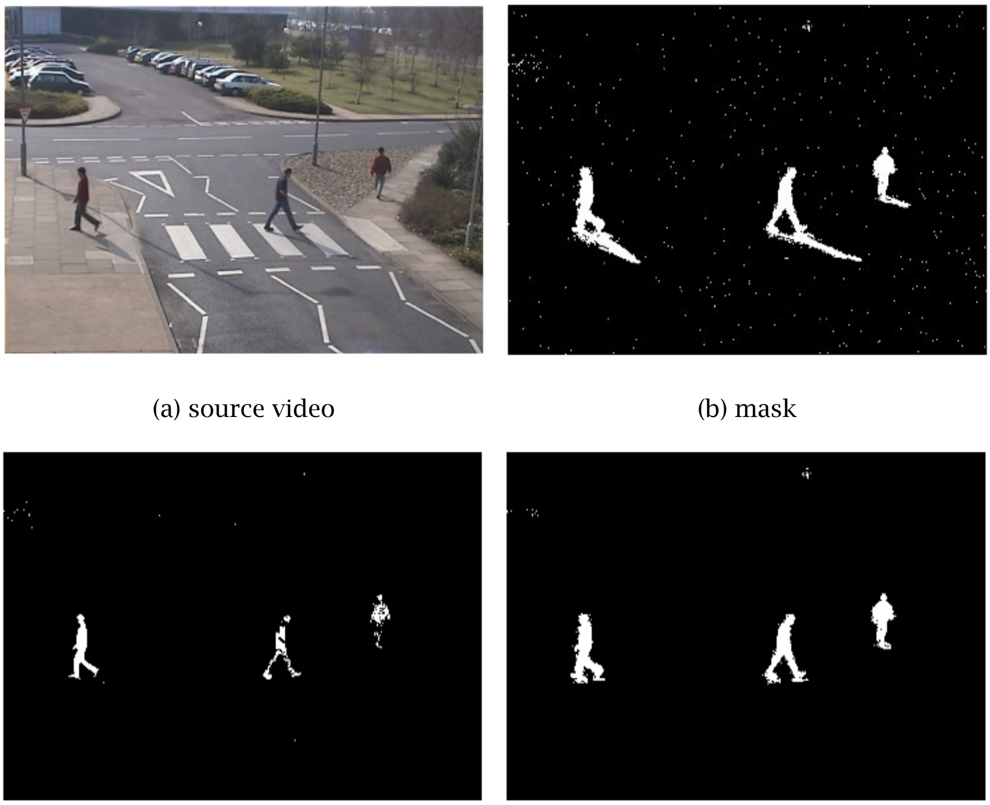
This novel combined scheme gives favorable results compared to the current state-of-the-art ones to suppress shadows / highlights. Figure 5.4 illustrates an example of this scheme at various processing stages.

5.5 Grouping Foreground Pixels into Meaningful Blobs

After the cast shadows / highlights removal procedure, a classic 8-connectivity connected component analysis (CCA) can be performed to group into blobs all those pixels that presumably belong to individual objects. These blobs are then ready to be temporally tracked throughout their movements within the scene along the time.

In addition to the classical CCA-based pixel grouping technique, we propose yet another more robust pixel grouping scheme which leads to more accurate blob segmentations. This is a CPU demanding technique. However, even though the scheme is not suitable for real-time operation with current hardware, it is still a very relevant approach for suspicious object detection applications where the method only has to be applied when an operator requires it.

The proposed method combines any of the adaptive background learning techniques previously discussed (§2.1, on page 5 and §4, on page 41) with a hierarchical segmentation method based on Binary Partition Trees (BPT) [SG00a]. The result is a region-based dynamic scene description, where each active region is characterized by a temporal feature, reflecting on the time it remains in the same position of the scene. This description is then used to classify the background and foreground objects of the scene and can also be used as an additional feature for region tracking and scene understanding.



(a) source video

(b) mask

(c) marker

(d) final foreground mask

Figure 5.4: (a) A snapshot of a surveillance video sequence, where the cast shadows from pedestrians are strong and large; (b) the result of initial foreground pixels segmentation, with the moving shadows being included; (c) the marker image obtained after the shadow removal processing; and (d) the final reconstructed objects with erroneous pixels corrected.

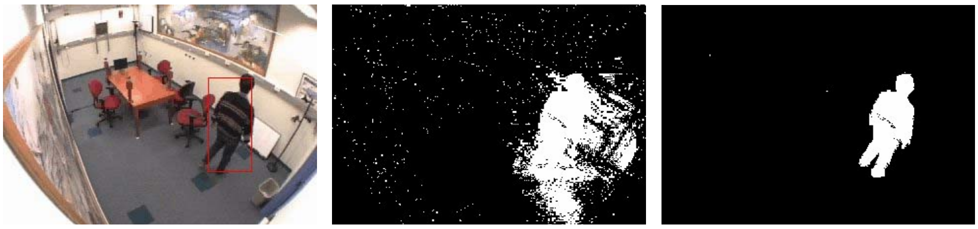


Figure 5.5: Illustration of an indoors smart room scenario. On the left, the incoming image; following, the mask image obtained after the foreground segmentation; and on the right, the final reconstructed objects shapes.

Figure 5.6 depicts schematically the block diagram of the proposed scheme. Based on an adaptive background model, we propose to create a *pixel persistence map* (PPM) indicating, for every pixel, the time elapsed since its color features have changed. In other words, the PPM will contain information about the time that any object in the scene has remained in the same position.

After the map has been created, the hierarchical BPT segmentation method is applied. We employ Binary Partition Trees since they provide a structured representation of the regions that can be obtained from an image. In fact, both spatial and temporal information are used to create this structure in our scheme. Color information is used to create an initial partition from which the BPT is initialized, and then, the pixel persistence information is employed to create the merges that lead to the final BPT.

One of the advantages of the proposed scheme is the possibility that it offers of distinguish between *old* and *recent* background. For instance, a car that has been parked (see Figure 5.8), a bag that has been abandoned (Figure 5.9), or a person who has entered into the scene and then stands still, will not be either integrated in the background or foreground as is the case with most common background learning techniques. Instead, they will be identified as independent objects characterized by the time that they have remained in the same position.

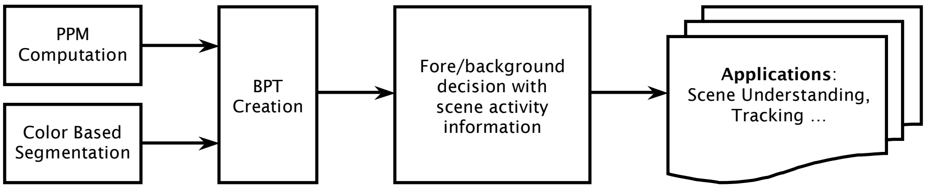

PPM Computation
Color Based Segmentation
BPT Creation
Fore/background decision with scene activity information
Applications: Scene Understanding, Tracking ...

Figure 5.6: The hierarchical segmentation system block diagram showing the chain of functional modules.

5.5.1 Pixel Persistence Map

In video sequences captured with a fixed camera, every background pixel usually shows similar values over the time. However, in *pixel activity* situations (foreground pixels), the new pixel values usually form a connected region with *temporal persistence* homogeneity. Imagine for instance an object placed over the floor at a certain moment (see Figure 5.9). Even if this object has different colors, all the pixels within the object will share the fact that they have been observed in the same place, during the same amount of time. The same reasoning can also be applied with regions corresponding to moving objects. In this case, all the pixels will share the fact that they have not been previously seen.

In order to segment the regions with the criterion of *temporal persistence* homogeneity, it is crucial to study first the pixel value occurrences along the time. In the proposed scheme, a probabilistic model is used for every pixel in the image to account for the recent history of photometric variations of the pixel. From now on, we particularize the discussion to the case of a mixture of Gaussians model since this particular model has been explained in detail before in this dissertation (see §2.1 and §4). However, the method can be generalized to other sort of models with minimal changes.

As explained in §4.4.4, on page 66, when using mixtures of Gaussians, it is possible to let the system replace a Gaussian of the model each time that a particular RGB value has not been seen before. This new Gaussian then characterizes this color appearance from that instant on. As we explained, this is the intended behavior in certain applications. Here, we use this ability to develop the following concepts.

The number of consecutive matches that the k-th Gaussian of a mixture of Gaussians explains an observation is a very important piece of information. We call this feature, the *background detections record* ($\mathrm{BDR}_{\mathbf{x},k}$), and we update it as follows[1]:

$$\mathrm{BDR}_{\mathbf{x},k}[t] = \begin{cases} \mathrm{BDR}_{\mathbf{x},k}[t-1] + \alpha(1 - \mathrm{BDR}_{\mathbf{x},k}[t-1]), & \text{if Gaussian } k \text{ explains the observation} \\ (1-\alpha)\mathrm{BDR}_{\mathbf{x},k}[t-1], & \text{otherwise,} \end{cases} \tag{5.3}$$

where $\frac{1}{\alpha}$ defines the *time constant* reflecting on the speed at which $\mathrm{BDR}_{\mathbf{x},k}[t]$ changes. Note that different criteria can be used to decide whether a certain Gaussian explains an observation or not. For instance, one could check whether the pixel's color value ($\mathbf{I_x}$) is within 2.5 standard deviations of the distribution mean, as discussed in §2.1.1.1, on page 11 and in §4.3.2, on page 48. On the other hand, one could compute the posterior probability of this particular mode, as discussed in §4.4.2 on page 62. Some considerations regarding shadowed or highlighted regions also have to be made here. When a shadow / specular reflection is wrongly detected as foreground and recovered later by the shadow removal scheme, then the PPM has to be built making use of this corrected classification, that is, choosing the Gaussian that best explains the corrected observation, even though it had not been chosen in the first iteration.

The $\mathrm{BDR}_{\mathbf{x},k}[t]$ update process can be considered as a low-pass filtered average of the number of occasions that the model has characterized a color appearance thus far. Therefore, the map will assume higher values in areas, e.g., background, where similar colors have appeared frequently and consecutively over the recent history. On the contrary, the map will show lower values in areas where new colors, e.g., due to a moving object, have appeared that sustain for a shorter period of time.

In addition, $\mathrm{BDR}_{\mathbf{x},k}[t]$ can be used to learn the exact number of *consecutive background detections* of a Gaussian mode. Indeed, if we solve the recursion in the equation, we can obtain the

[1] Note that in an exception-to-background approach using mixtures of Gaussians, the PPM can be directly built using the weights of the Gaussians which characterize the color values of the current frame $\mathrm{BDR}_{\mathbf{x},k}[t] = w_{\mathbf{x},k}[t]$. However, in an EM-MAP approach (see chapter 4) this assumption does not hold, and therefore $\mathrm{BDR}_{\mathbf{x},k}[t]$ has to be explicitly calculated.

number of frames elapsed since the background model with initial $\mathrm{BDR}_{\mathbf{x},k}[0]$[1] has repeatedly characterized the same color until the current value of $\mathrm{BDR}_{\mathbf{x},k}[t]$. Therefore, we can determine for how long an object has had its presence in a certain position:

$$\begin{aligned}\mathrm{BDR}_{\mathbf{x},k}[t] &= (1-\alpha)\mathrm{BDR}_{\mathbf{x},k}[t-1]+ \\ &(1-\alpha)^t\mathrm{BDR}_{\mathbf{x},k}[0] + \alpha\sum_{i=0}^{t-1}(1-\alpha)^i \underbrace{=}_{0<(1-\alpha)<1} (\mathrm{BDR}_{\mathbf{x},k}[0]-1)(1-\alpha)^t + 1. \qquad (5.4)\end{aligned}$$

And, thus:

$$t = \log_{1-\alpha}\frac{\mathrm{BDR}_{\mathbf{x},k}[t]-1}{\mathrm{BDR}_{\mathbf{x},k}[0]-1}. \qquad (5.5)$$

Finally, the $\mathrm{BDR}_{\mathbf{x}}[t]$ of pixel $\mathbf{x}$ at a certain time t, is that one corresponding to the $\hat{k}$-th Gaussian that best explains it:

$$\mathrm{BDR}_{\mathbf{x}}[t] = \mathrm{BDR}_{\mathbf{x},\hat{k}}[t]. \qquad (5.6)$$

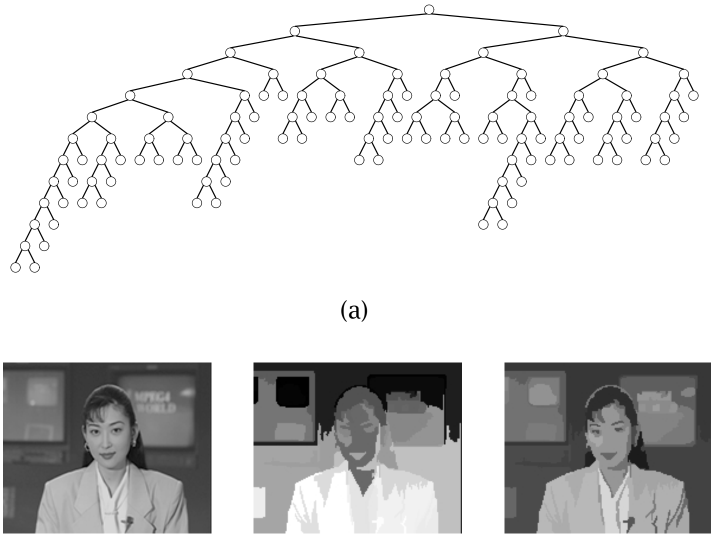
The $\mathrm{BDR}_{\mathbf{x}}[t]$ will be the feature used to create the pixel persistence map in the following section.

5.5.2 Binary Partition Tree

A Binary Partition Tree [SG00a] is a structured and compact representation of the most likely regions that can be obtained from an initial partition of an image given a certain homogeneity criterion. Several approaches can be used to create this tree. We have used a segmentation that follows a bottom-up approach. The selected algorithm first constructs the Region Adjacency Graph of an initial partition. Using an iterative region mergin algorithm, the BPT is then created by keeping track of the regions that are merged at each iteration until one region is obtained. That is, for each pair of neighboring regions a homogeneity measure is assessed, and the pair whose distance is the lowest is merged. The process is iterated until one final region is obtained. An example is shown in Figure 5.7. We have taken an initial partition of only 60 regions in Figure 5.7 (b). The leaves of the tree in Figure 5.7 (a) represent the regions of this initial partition. The remaining nodes of the tree represent regions that are obtained by merging the regions represented by its two child nodes. The root node represents the entire image support.

The BPT should be created in such a way that the most meaningful regions are represented in its nodes. In our case, we aim at detecting foreground objects. For this reason we generate an initial partition by merging of flat zones with a spatial color similarity criterion, and from this initial partition, we then create the BPT using the temporal persistence map.

[1] $\mathrm{BDR}_{\mathbf{x},k}[0]$ can be simply set to zero. However, if one is using the exception-to-background approach described by Stauffer and Grimson [SG00b], then $\mathrm{BDR}_{\mathbf{x},k}[0] = w_k[0]$, where $w_k[0]$ is a design parameter of the system.



(a)

(b)

Figure 5.7: Example of Binary Partition Tree. In (a) the Binary Partition Tree is shown. In (b), from left to right: Original image; Initial partition with 60 regions; Image where each region of the original image has been filled with the mean color value of the original image.

For the initial partition, a region growing approach [SG00a] is used. The homogeneity measure that is used in this first stage is based on a weighted Euclidean distance, shown in (5.7), in the YUV color space, where more emphasis is given to the luminance component:

$$d(r_1,r_2) = \sqrt{\gamma(\overline{y_1} - \overline{y_2})^2 + \frac{1-\gamma}{2}\,((\overline{u_1} - \overline{u_2})^2 + (\overline{v_1} - \overline{v_2})^2)}, \tag{5.7}$$

where $\overline{y_i}$, $\overline{u_i}$ and $\overline{v_i}$ correspond to the mean value of the Y, U and V channels, respectively, over all the pixels in region i.

Regions (r_1, r_2) are compared using the mean values of their YUV components. Regions are merged until a termination criterion is reached (usually the number of regions or the PSNR obtained when representing the original image by the partition with all regions filled with their mean values). This first stage leaves the initial partition.

The homogeneity feature that we use to construct the BPT from the initial partition is the PPM. In this second stage, the new merging criteria is the L_1 distance between the $\overline{\mathrm{BDR}_i}$ of two regions (r_1, r_2):

$$d(r_1,r_2) = \left|\overline{\mathrm{BDR}_{\mathbf{1}}[t]} - \overline{\mathrm{BDR}_2[t]}\right|, \tag{5.8}$$

where $\overline{\mathrm{BDR}_{\mathbf{i}}[t]}$ is the value of node i, corresponding to the mean value of the $\mathrm{BDR}_{\mathbf{x}}[t]$, as defined in equation (5.6), over all the pixels $\mathbf{x}$ in region i.

That is, the merging order among the regions is defined using the distance between the mean values of the persistence within the corresponding regions.

In this stage, the complete BPT is constructed defining the mergings until a single node is reached.

Using this homogeneity criterion, the nodes of the tree are characterized by the time their corresponding regions have remained in the same position of the scene.

5.5.3 Discussion and Experiments

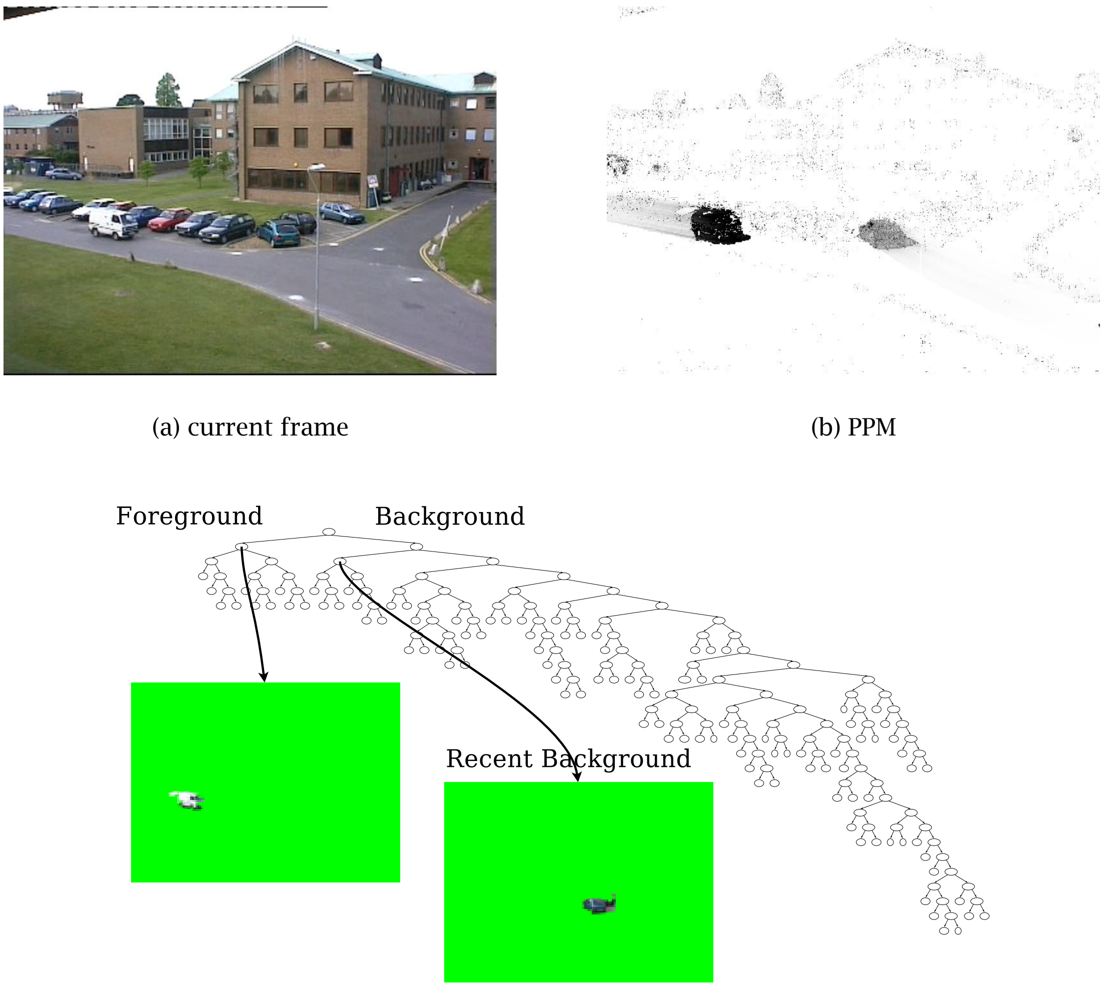
The tree can be used as an inspection tool by an operator. By choosing just a few of the upper-level nodes of the tree (see Figures 5.9 and 5.8), the operator can visually identify the regions corresponding to objects recently positioned in a specific place. The operator can then obtain the minutes and seconds that a suspicious object has been static. In addition, more sophisticated tools can be created so that the tree is automatically analyzed in search of objects with specific characteristics, such as a particular shape, color and persistence in the scene.

The system has been evaluated on standard test sequences such as the set of benchmarking images sequences provided by PETS'2001 and a range of our own captured image sequences under various compression formats.

Figure 5.8 (a) shows an example where the green car on the street bend has been parked a few moments (frames) ago, and the white van is currently moving. First, the current frame is segmented using color homogeneity until 500 regions are left. Then, the PPM (b) is used as the merging criteria to obtain the BPT, in (c). As there is only one foreground object, a second level node contains the entire background. When there are multiple foreground objects, the background scene is represented in lower-level nodes. A termination criterion can be set so that the complete BPT is not built and the node representing the background does not merge with the nodes containing foreground appearances. Moreover, we can determine using (5.5), the temporal persistence of the node that has been labeled as *recent background* (Figure 5.9 (c)). In this case, $t = 169.6$, with $\alpha = 0.002$, and $\mathrm{BDR}_{\mathbf{x}}[0] = 0$ which indicates very accurately during how many frames the car has been parked (the image shown corresponds to frame 750, camera 1 of PETS'01 sequences).

Figure 5.9 depicts another example: in this case, a bag has been left on a table and a person is walking. Similar considerations to the previous case apply here. This situation, though, offers an example of a typical surveillance situation where a suspicious package is detected, and it is urgent to know in which instant it was placed there, so that security agents can easily inspect the recorded images at the exact instant. In this case $t = 116.5$, with $\alpha = 0.005$ and $\mathrm{BDR}_{\mathbf{x}}[0] = 0$.

In this section, we have presented a novel approach to object segmentation in video sequences - a hierarchical segmentation procedure using BPTs - which builds upon the temporal information derived from a pixel-wise background learning technique based on mixtures of Gaussians.

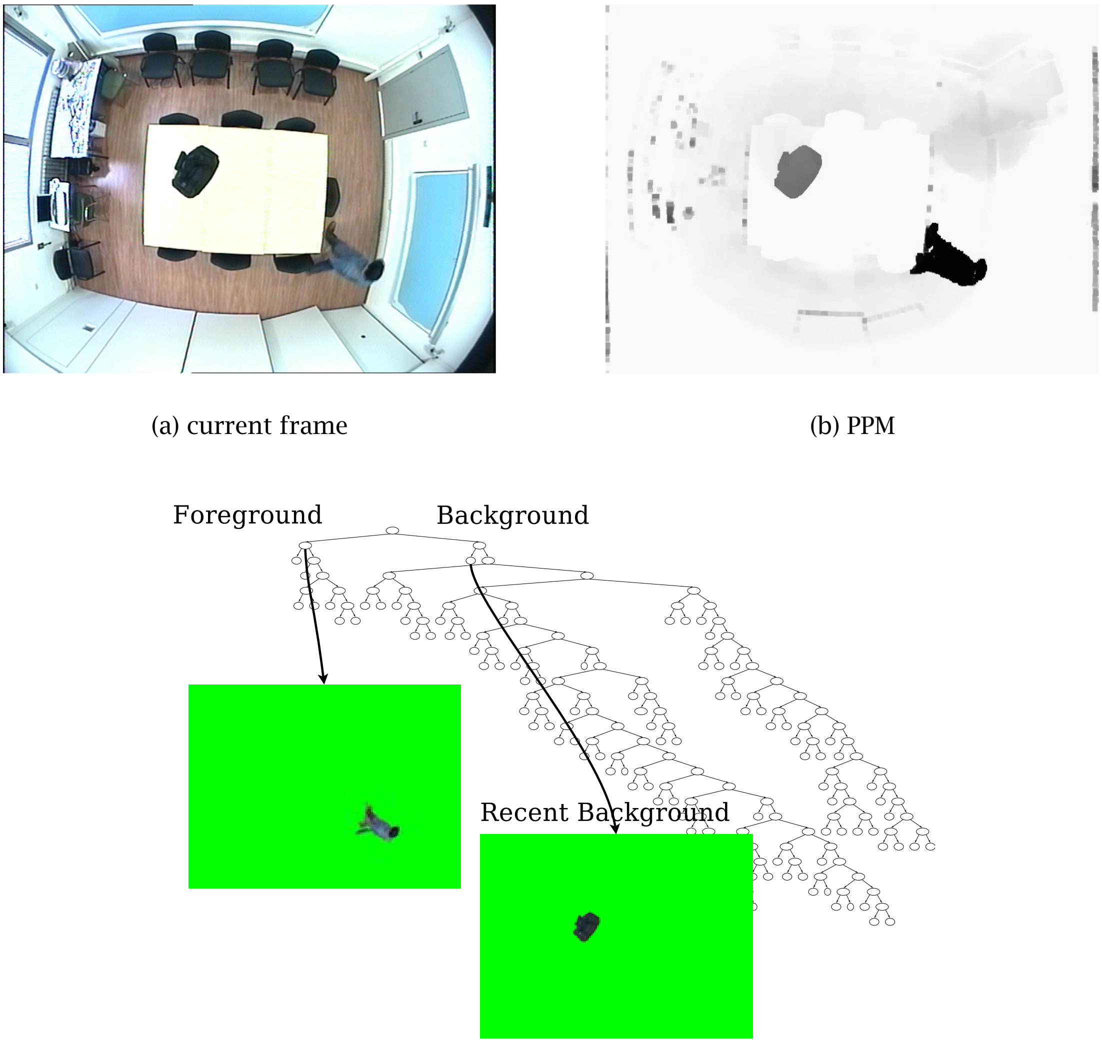


(a) current frame

(b) PPM

(c) BPT

Figure 5.8: An outdoor scene example of the BPT creation.



(a) current frame

(b) PPM

(c) BPT

Figure 5.9: An indoors scene example of the BPT creation.

The system has proved to be very robust to sporadic detection errors. Since the $\text{BDR}_{\mathbf{x},k}$ of each Gaussian are updated as a low pass filter process, an individual error in a certain frame does not distort the accumulated $\text{BDR}_{\mathbf{x},k}$. The method also has some drawbacks, though. In fact, the proposed scheme accounts for the observation time of certain colors in pixels. However, when these pixels are partially occluded during a period of time, the $\text{BDR}_{\mathbf{x},k}$ of each Gaussian are not updated. In other words, the method does not provide the time since an object was placed on a certain position, but the time for which the object has been seen. In the proposed scenarios, both measures tend to be very similar, but this consideration has to be kept in mind when employing the proposed scheme.

To finalize, the advantages of this scheme can be summarized into:

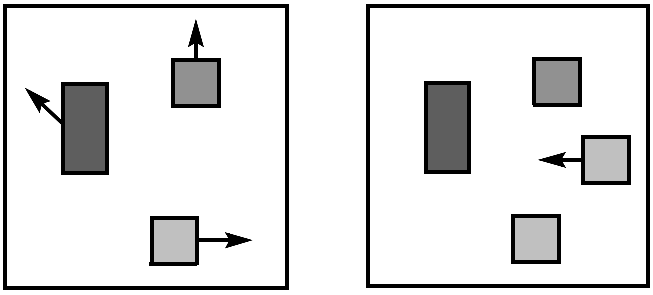
- This approach is a novel attempt to combine spatial information with background modeling for segmentation. The foreground / background decisions are taken on a region basis instead of a pixel-basis. The decision is thus more robust to noise effects and does not require a connected component analysis to classify the different foreground objects.
- The detected regions are characterized by the time elapsed since they reached the current position.
- Using the BPT, we can separate the foreground objects from the background, as well as distinguish between *old* and *recent* background. That is, an object which has reached a stable position in the scene (a recently parked car, a newly abandoned bag, a moving person becoming still), which tends to become part of the background in most state-of-the-art techniques, can be easily identified with the proposed approach.

5.6 Multi-Object Tracking Using Temporal Templates

After grouping detected pixels (free of cast shadows and highlights) into blobs, the active entities within the scene are ready to be temporally tracked throughout their movements along the time.

Efficient tracking of multiple objects is a challenging and important task in computer vision where the performance of the tracking algorithms depends on the scenario. In the following, we discuss the basics of an effective single-fixed-camera, far-field tracking system working in an outdoors scenario. Even though the main operation blocks are described in the following, we advice the reader to refer to [LPX04, SW05, XL07a, XL07b, XLL04] for a more detailed discussion about system implementation issues. A 3D extension to the system can be found here [LP05]. Finally, a version of the tracker that includes both visual and acoustical information can also be found in [ACFS^{+}06].

Figure 5.10 illustrates an example where three entities (indexed by l) have been tracked to frame t. The figure shows the uncertainty during the matching process between the tracked entities and the newly detected candidate blobs (indexed by k) in frame $t+1$. As the figure suggests, in addition to the matching, the system has to be able to detect new candidates. All these aspects will be discussed in the subsequent sections.



(a) Objects (templates) in frame t

(b) Candidate blobs in frame $t+1$

Figure 5.10: The illustration of object tracking between two consecutive frames. On the left are the three objects already tracked, for which feature template models exist; on the right are the four newly detected candidate blobs in frame $t+1$, for which matching to the corresponding tracks are sought, noting the far right one just enters the viewing scene.

5.6.1 Temporal Templates

Each object of interest in the scene is modeled by a temporal template of persistent characteristic features. In the current studies, a set of five significant features are used describing the *velocity* $\mathbf{v} = (v_x, v_y)$ at its centroid $\mathbf{c} = (c_x, c_y)$; the *size*, or number of pixels, contained (s); the *ratio* (r) of the major-axis vs. minor-axis of the best-fit ellipse of the blob [FF95]; the *orientation* of the major-axis of the ellipse (θ); and the *dominant color* representation (**dc**), using the principal eigenvector of the aggregated pixels' color covariance matrix of the blob.

Therefore at time t, we have, for each object l centered at $(c_{l,x}, c_{l,y})$, a template of features $\mathbf{T_l}[t] = (\mathbf{v_l}, s_l, r_l, \theta_l, \mathbf{dc_l})$ (refer to Table 5.1 for a summary of the features).

5.6.2 Matching Procedure

We choose to use a parallel matching strategy in preference to the serial matching one such as that used in [ZA01]. The main issue now is the use of a proper distance metric that best suits the problem under study. Obviously, some features are more persistent for an object while others may be more susceptible to noise. Also, different features normally assume values in different ranges with different variances. Euclidean distance does not account for these factors as it will allow dimensions with larger scales and variances to dominate the distance measure.

Table 5.1: The five significant features of each object

$\mathbf{v_l} = (v_{l,x}, v_{l,y})$	the velocity at its centroid $\mathbf{c_l} = (c_{l,x}, c_{l,y})$
s_l	the size, or number of pixels contained
r_l	the ratio of the major and minor axis of the best-fit ellipse of the blob [FF95]; it is a better descriptor of an objects posture than its bounding box
θ_l	the orientation of the major-axis of the ellipse
$\mathbf{dc_l}$	the dominant color, computed as the principal eigenvector of the color co-variance matrix for pixels within the blob [ZA01]

One way to tackle this problem is to use the Mahalanobis distance metric, which takes into account not only the scaling and variance of a feature, but also the variation of other features based on the covariance matrix. Thus, if there are correlated features, their contribution is weighted appropriately.

However, in the current work, the features are assumed to be statistically independent, and therefore the covariance matrix is assumed to be diagonal. In fact, the features proposed in Table 5.1 can be safely considered to be independent, and consequently the mentioned assumption does not have practical implications while it allows faster computations.

Taking the previous considerations into account, the final measure of distance between the template l and a candidate blob k is shown in (5.9):

$$d(l,k,t) = \sqrt{\sum_{i=1}^{N} \frac{\left(\hat{\mathbf{T}}_{l[i]}[t] - \mathbf{C}_{k[i]}[t+1]\right)^2}{\sigma^2_{l[i]}[t]}} \tag{5.9}$$

where the index i runs through all the $N = 5$ features of the template, $\hat{\mathbf{T}}_{l[i]}[t]$ is the prediction of template $\mathbf{T}_{l[i]}[t]$ and $\sigma^2_{l[i]}$ is the corresponding component of the variance vector $\sigma^2_l[t]$, that is dynamically updated frame by frame once a match with a candidate $\mathbf{C}_{k[i]}[t+1]$ is performed. $\hat{\mathbf{T}}_{l[i]}[t]$ can be predicted making use of particle filters, Kalman filters and similar techniques. In our experiments we have employed a Kalman filter [WB03] since it is sufficiently efficient to permit estimating the state of our dynamic system from the series of incomplete and noisy measurements.

Having defined a suitable distance metric, the matching process can be described in greater detail as follows (refer to Table 5.2 for a schematic overview of all the important aspects during the tracking process).

Step 1: For each new frame at time $t + 1$, all the valid candidate blobs detected k are matched against all the predicted object templates l via equation (5.9). A ranking list is then built for all the pairs object (l) - candidate (k). Then this list is sorted from low to high cost.

Table 5.2: Key aspects during the tracking process

$\mathbf{T}_{l[i]}[t]$	the template of features
$\sigma_l^2[t]$	its vector of variances
$\mathrm{KF}_l[t]$	the related set of Kalman filters
$\mathrm{TK}_{\mathrm{counts}}[t]$	the counter of tracked frames, i.e., current track length
$\mathrm{MS}_{\mathrm{counts}}[t]$	the counter of lost frames
$\hat{\mathbf{T}}_{l[i]}[t]$	the expected values in frame $t+1$ by Kalman prediction

The matching pairs with the lowest cost value $d(l,k,t)$, that is also less than a threshold (e.g., 10 proved to be reasonable), are identified as match pairs.

Step 2: If object l is matched by the candidate blob k in frame $t + 1$, i.e., by way of the template prediction $\hat{\mathbf{T}}_{l[i]}[t]$, then $\mathrm{TK}_{\mathrm{counts}}$ is increased by 1, and the normal updates for l are performed.

Step 3: If object l has found no match at all in frame $t + 1$, presumably missing or occluded, then $\mathrm{MS}_{\mathrm{counts}}$ is increased by 1. The object l is carried over to the next frame, though the following rule apply:

If object l has been missing for a number of frames, or $\mathrm{MS}_{\mathrm{counts}} \geq$ MAX_LOST (e.g., 5), then it is discarded, taken as either becoming still (merged into background) or having entered into a building or car. Otherwise, if $\mathrm{MS}_{\mathrm{counts}} <$ MAX_LOST, the vector of variances $\sigma_l^2[t]$ is adjusted according to (5.10) to assist the tracker to recover the lost object that may undergo unexpected or sudden movements.

$$\sigma_{l[i]}^2[t+1] = (1+\delta)\sigma_{l[i]}^2[t], \tag{5.10}$$

where $\delta = 0.05$ is a good choice.

Step 4: For each candidate blob k in frame $t + 1$ that is not matched, a new object template $\mathbf{T}_k[t + 1]$ is created from $\mathbf{C}_k[t + 1]$. The choice of initial variance needs some consideration, which can be copied from either very similar objects already in the scene or typical values obtained by prior statistical analysis of correctly tracked objects. Note that this new object will not be declared (marked) until after it has been tracked for a number of frames, or $\mathrm{TK}_{\mathrm{counts}} \geq$ MIN_SEEN (e.g., 20), so as to discount any short momentary object movements. Objects will be discarded if they do not satisfy this condition.

5.6.3 Occlusions Handling

In the current approach, no use is made of any special heuristics on areas where an object may enter (exit) into (from) the scene. The possible background structures that may occlude moving foreground objects are also unknown *a priori* [XE02]. Objects may just appear or disappear in the middle of the image, and, hence, positional rules are not enforced, as opposed to Stauffer [Sta03].

During the possible occlusion period, the object template of features is updated using the average of the last 50 correct predictions to obtain a long-term tendency prediction. Occluded objects are better tracked using the averaged template predictions. In doing so, small erratic movements in the last few frames are filtered out. Predictions of positions are constrained within the blob that occludes the current *occluded* object.

5.6.4 Experimental Results

The system has been evaluated extensively on standard test sequences such as the set of benchmarking image sequences provided by PETS'2001 and a range of our own captured image sequences under various weather conditions and video compression formats

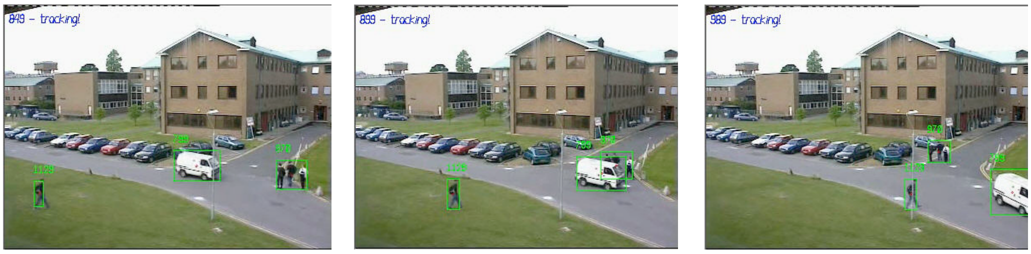


Figure 5.11: An example (from PETS2001) illustrating one of the difficult tracking situations that the system handles successfully, in which the moving white van, first occluded by the thin streetlight pole, then partially occludes a group of walking people (from left to right): before, during, and after occlusion. The tracking labels have been correctly kept.

For PETS sequences, original images are provided in JPEG format, and their frame size is 768 × 576 pixels. In our experiments though, the sub-sampled images of size 384 × 288 pixels were used. Also, an AVI video file was created for each image sequence using an XviD codec, introducing a second temporal compression. Apart from these compression artifacts, the imaging scenes also contain a range of difficult defects, including thin structures, window reflections, illumination changes due to slowly moving clouds, and swaying leaves in trees. Our system has dealt with all these problems successfully, and handles very well the complex occlusion situations. Figure 5.11 shows an example where the white van is occluded by a thin structure, or streetlight pole (left), and subsequently a group of people are largely blocked by the van for a few frames (middle).

For the other sequences, a CIF-size image frame (352 × 288 pixels) is used. The original video was captured at 25 fps using Mini DV format, and then converted to MPEG-1, followed

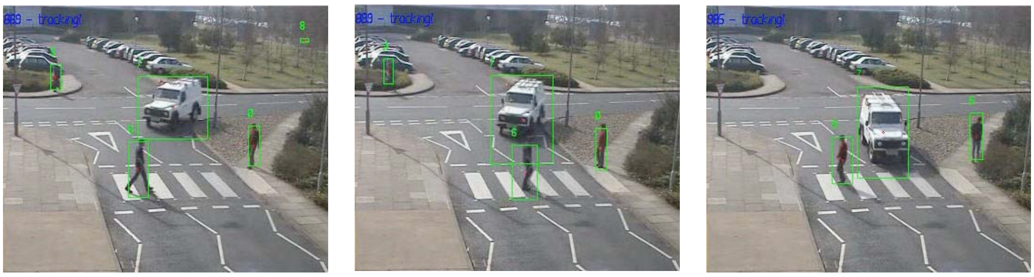


Figure 5.12: Another example illustrating the success of the system in dealing with severe shadows problem and complex dynamic occlusion situation. Two people were walking towards each other across the pedestrian crossing, whilst a van is approaching and slowing down (from left to right) before, during, and after their intersection.

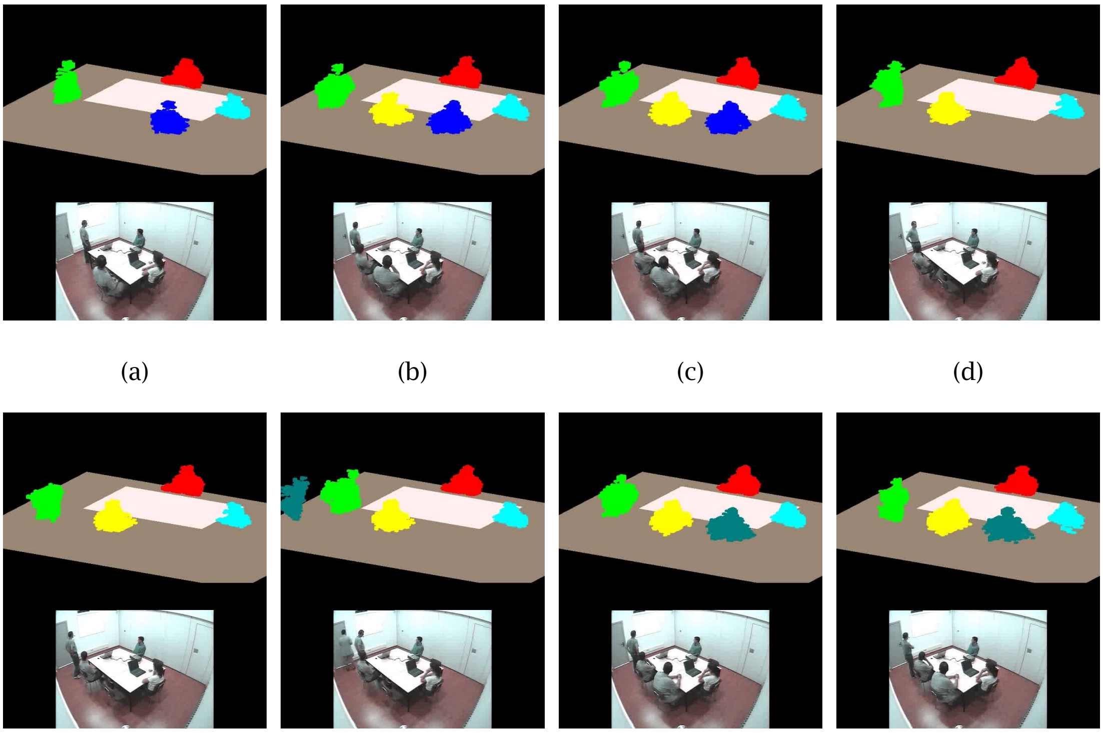
by an XviD compression. Figure 5.12 illustrates an example of a complex and difficult situation where large and strong shadows exist and three objects (two people and a van) pass by each other. The system runs at an average rate of 12 fps on a PC with a single 2 GHz Pentium-4 processor.

Some problems occurred when a few individually moving objects start to join each other and form a group. These objects are correctly tracked within the limit of predefined MAX_LOST frames as if they were occluding each other. Beyond this limit the system decides that they have disappeared, and creates a new template for the whole group. Other problems may occur when objects abruptly change their motion trajectories during occlusions; sometimes the system is able to recover the individual objects after the occlusion, but on other occasions new templates are created.

The system copes with shadows and highlights satisfactorily in most cases, though very long cast shadows may not always be completely removed. A small defect of the algorithm is that the reconstructed region contains a small patch of shadow in an object's exterior where the cast shadow starts (see the feet of the people in Figure 5.4(d)). This patch is about half the size of the structuring element used, and is produced during the conditional dilation. Intersection with the mask image cannot suppress this segment as all the shadowed regions form part of the mask.

Finally, Figures 5.13 and 5.14 show some results of the 3D extension of our tracker. The 3D tracker does not have problems with inter-object occlusions at different depths in the 3D domain. Note that the figures show the projection of 3D volumetric information where the view-point can be freely set (see Figure 5.14 for an example of image rendering at slightly different angles).

However, object grouping and de-grouping can still be a challenge. When different objects are very close, the system tends to identify them as forming part of the same object as Figure 5.14 shows. In spite of this, the system is able to recover the correct identities of the tracked objects after de-merging (see Figure 5.14(c)) by employing the features stored in the templates. The complete details of the 3D extension of the tracker are described in [LP05].



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

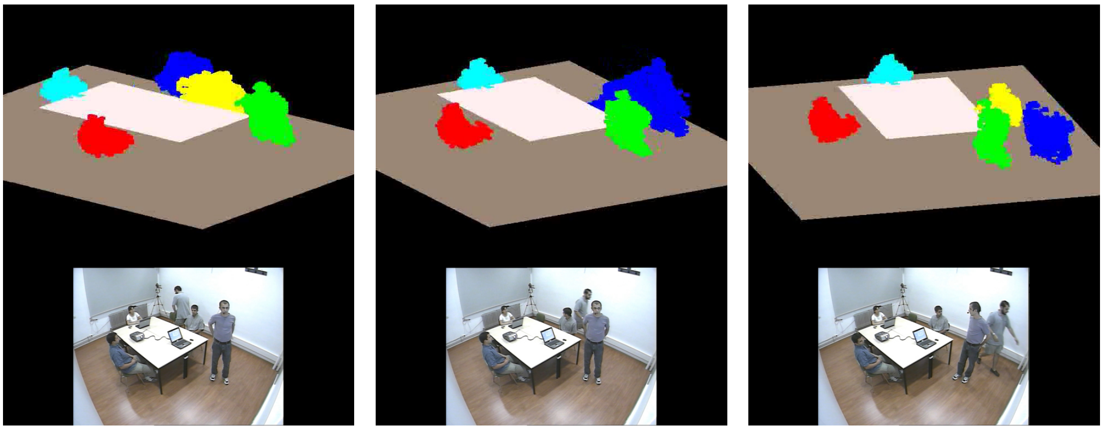
(e) (f) (g) (h)

Figure 5.13: An example of our 3D tracker [LP05]. The illustration shows the projection of the 3D volume. Different colors have been assigned to each one of the 3D objects to show their temporal correspondence. The sequence of images in the figure corresponds to one frame for every 30 seconds. The complete sequence can be obtained in `http://gps-tsc.upc.es/imatge/_jl/`.

5.7 Conclusion

In this chapter, we have presented a vision-based system for accurate segmentation and tracking of moving objects in cluttered and dynamic outdoor environments surveyed by a single fixed camera. Each foreground object of interest has been segmented and shadows/ highlights removed by an effective scheme.

We have also introduced a novel foreground segmentation method that takes into account the spatial information of the scene and characterizes and groups pixels into regions according the time they have remained in the same position of the scene. Thus, the method is able to



(a) (b) (c)

Figure 5.14: Another example of our 3D tracker. In this figure, a frame is shown for every two seconds. When objects are very close, the system tends to identify them as forming part of the same object. However, the tracker is able to recover correct object identities after de-merging by employing the features stored in the templates.

provide a hierarchical classification of the objects depending on the time elapsed since they reached the current position.

In the chapter, we have also presented a tracking system based on the previously mentioned segmentation and shadow removal steps. The 2D appearance of each detected object blob is described by multiple characteristic cues including velocity, size, elliptic-fit aspect ratio, orientation, and dominant color. This template of features is used, by way of a scaled Euclidean distance-matching metric, for tracking between object templates and the candidate blobs appearing in the new frame.

In completing the system, we have also introduced technical solutions dealing with false foreground pixel suppression, and temporal template adaptation. Experiments have been conducted on a variety of real-world wide-area scenarios under different weather conditions. Good and consistent performance has been confirmed. The method has successfully coped with illumination changes, partial occlusions, clutters, and scale and orientation variations of objects of interest and, especially, it is not sensitive to noise incurred by the camera imaging system and different video codecs.

Chapter 6

Cooperative Background Modeling using Multiple Cameras

In this chapter and the following one the 3D foreground detection is the main focus. This chapter is devoted to a theoretical and experimental study of a Bayesian volumetric foreground detection technique and the following one describes a technique for estimating that part of the volume which projects inconsistently and propose a criteria for classifying it.

The Visual Hull is formally defined as the intersection of the visual cones formed by the back-projection of several 2D binary silhouettes into the 3D space. Silhouettes are usually extracted using a foreground classification process, which is performed independently in each camera view.

In a temporal perspective, 2D foreground segmentation techniques initially established the basement which was later used for object / people detection in 2D. Advances in the field led to this other set of systems designed to build binary volumetric models (the Visual Hull) from a set of binarized foreground masks, in what is called Shape from Silhouette (SfS) [Bau74, Lau91, Lau94, Lau95]. In this chapter we reconsider the current 3D segmentation chain. We offer a new perspective in which planar foreground classification in a view is achieved in accordance with the rest of views in a Bayesian framework. In this novel approach, the 3D space is simultaneously reconstructed and classified, instead of previously classifying images and reconstructing the volume later, as it is the case of current approaches. This is achieved by obtaining 3D probabilistic information from 2D probabilistic maps and then projecting back 3D probabilities to each view. Projected probabilities are then used to update the two-dimensional background models. The scheme permits to obtain better 3D foreground detections that in turn also permit to obtain better planar foreground detections.

The advantages of the system proposed are threefold. First, 2D models are updated using 3D information, and therefore giving more precise 2D and 3D classifications, second, the proposed scheme permits setting fine-grained thresholds for 3D space classification, and third, it is possible to set priors or external probabilistic information in the 3D space. All these advantages are discussed in detail in the chapter.

6.1 Introduction

Background modeling of two-dimensional imaging scenes has been one of the fields where computer vision has had a major impact, driving successful deployment of several visual surveillance / behavior modeling systems.

Several multi-camera 3D foreground detection systems are based on these 2D foreground segmentation techniques that make use of 2D background modeling. Shape from Silhouette (SfS) for 3D model extraction is the approach taken in most of these systems (see §2.2.4 on page 27 for a complete review of the SfS approach).

In the state-of-the-art SfS approaches that employ 2D binary masks extracted after background modeling, the models in each view are always temporally maintained along the time using evidence only from that view. Contrarily, in our proposed approach, the 2D models are updated using evidence from all the cameras in a Bayesian framework. Thus, both 2D and 3D classifications are performed using existing information in all views along the time.

3D Bayesian classification also allows a more fine-grained classification than traditional SfS since the 3D space is classified making use of 2D probabilities instead of 2D *foreground* or *background* binary information. In [FB05], a fine-grained classification scheme was also employed. However, the 3D map was built from 2D likelihoods instead of 2D probabilities. In addition, the 3D map was not used to feed back the background models in the images. In [SVZ00] an algorithm based on graph cuts determines the 3D shape with lowest cost (smoothest shape consistent with the observations). Even though it is not a probabilistic approach, it allows taking decisions not only based on binary silhouettes.

Finally, the proposed technique permits setting external probabilistic information or foreground and background priors to 3D regions instead of a more tedious 2D prior setting. Say for instance a situation where an operator wants to set low foreground priors to those regions below a table in a room. In our more flexible method, one can define a geometric region in real world coordinates that is automatically transferred to the 2D models so that it is not necessary to manually inspect all the 2D views. Also, 3D prior setting automatically handles the problem of 2D region occlusion. Normally, it is not possible to set priors in image regions because foreground objects can occasionally occlude this area. Indeed, 2D regions represent some part of the viewing scene at different depths and this makes it difficult to apply 2D priors. However, in a 3D Bayesian classification framework, 3D priors do not suffer from this limitation since occlusions are only inherent to the 2D space. Apart from setting 3D priors, the scheme permits incorporating external 3D probabilistic information to the 3D map. So, taking the idea a bit further, in the following chapter we propose a reconstruction method that is able to obtain the volume that minimizes the probability of volumetric misclassification. In other words, the method is able to obtain the most probable global explanation of a multi-camera scene. In §7.5 on page7.5, we explain how this external piece of information can be incorporated back to the Bayesian 3D map built using the tools that we following present.

The rest of the chapter is structured as follows. In section 6.2, the basis on which our probabilistic 3D classification approach is built is given. Section 6.3 describes the 3D classification method and section 6.4 extends the 3D foreground classification framework so that 2D outliers are taken into consideration. In section 6.5, a 2D model update process making use of multi-camera information is described. Sections 6.6 and 6.7 presents the implementation issues and experimental studies of this system with real-world tests, respectively. The chapter concludes in section 6.8 with a discussion of future research directions and system enhancement.

6.2 Probabilistic Voxel Classification

In order to define our proposed 3D classification framework, first it is important to establish foundations on which the 3D classification process is based.

2D foreground classification can be achieved via many different approaches, as discussed in chapter 2. However, a 3D probabilistic representation is only possible when the two-dimensional foreground classifications of the scenes can be probabilistically justified.

In chapter 4, we provided the basis for Bayesian 2D foreground classification. The chapter also discussed how to adapt existing exception-to-background approaches to a maximum a posteriori (MAP) scheme, showing that MAP outperforms exception-to-background approaches. The proposed approach presented here takes the MAP scheme for 2D foreground classification as its building block.

Before a more detailed description is made, we first need to adapt the notation used in chapter 4 so that a pixel position $\mathbf{x}$ and an image view $\mathbf{I}$ can be referred to each one of the C views: $\mathbf{x_i}$ and $\mathbf{I}_i$. In addition, for the sake of simplicity, we particularize the pixels' background models to the simple case of a single Gaussian per pixel $G_{i,\mathbf{x_i}}(\mathbf{I}_i(\mathbf{x_i}))$ (corresponding to view i, pixel $\mathbf{x_i}$)) and we employ a uniform function as the foreground distribution. These particular distributions are chosen in order to concur with the models used to explain the MAP setting in §4.3.1 on page 47. However, any other foreground and background likelihood functions can be used without any problems whatsoever.

As a final remark, we take the voxel-based SfS approach to define our Bayesian 3D classification framework. For a more in-depth description of other SfS variants and differences among them, we advise the reader to refer to §2.2.4 on page 27.

In the following, we first consider error-free 2D models, and in section 6.4, we include an outlier model to the 2D models.

6.3 Probabilistic Voxel Classification Considering Error-Free 2D Models

Voxel-based SfS can be thought as a classification problem. Consider a pattern recognition problem where, in a certain view $\mathbf{I}_i$, a voxel in location $\mathbf{v}$ is assigned to one of the two classes ϕ (2D-foreground), or β (2D-background), given a measurement $\mathbf{I}_i(\mathbf{x_i})$, corresponding to the pixel value of the projected voxel: $\mathbf{v} \rightarrow \mathbf{x_i}$, in camera i [HZ04][1].

Now, let us represent with super classes $(\Gamma_0, \cdots, \Gamma_K)$ all possible combinations of 2D-fore/background detections in all views $(i = 1, \cdots, C)$:

$$\begin{array}{lllllll}
\Gamma_0 & = \{ & \phi, & \phi, & \phi, & \cdots, & \phi \quad \} \\
\Gamma_1 & = \{ & \beta, & \phi, & \phi, & \cdots, & \phi \quad \} \\
\Gamma_2 & = \{ & \phi, & \beta, & \phi, & \cdots, & \phi \quad \} \\
\vdots & & & & & & \\
\Gamma_C & = \{ & \beta, & \beta, & \phi, & \cdots, & \phi \quad \} \\
\vdots & & & & & & \\
\Gamma_k & = \{ & \Gamma_k[1], & \Gamma_k[2], & \Gamma_k[3], & \cdots, & \Gamma_k[C] \ \} \\
\vdots & & & & & & \\
\Gamma_K & = \{ & \beta, & \beta, & \beta, & \cdots, & \beta \quad \}
\end{array}$$

with the following prior probabilities

$$\begin{aligned}
P(\Gamma_0) &= P(\phi)P(\phi)\cdots P(\phi) = P(\phi)^C = P_S \\
P(\Gamma_1) &= P(\beta)P(\phi)\cdots P(\phi) = P(\beta)P(\phi)^{C-1} \\
&\vdots \\
P(\Gamma_K) &= P(\beta)P(\beta)\cdots P(\beta) = P(\beta)^C,
\end{aligned}$$

where a voxel classified as foreground, i.e., a voxel of the 3D-Shape, belongs to super class Γ_0, with P_S prior probability[2]. Contrarily, an undetected voxel, i.e., a voxel of the 3D background, belongs to any of the other super classes $(\Gamma_{k\neq 0})$, since voxels are not detected when *at least* one projected voxel $(\mathbf{x_i})$ is not classified as a foreground pixel. The total number of 3D background super classes is $K = \sum_{i=1}^{C} \binom{C}{i}$.

According to Bayesian theory, given observations $(\mathbf{I}_i(\mathbf{x_i}), i = 1, \cdots, C)$, a super class Γ_j is assigned, provided the a posteriori probability of that interpretation is maximum:

$$P(\Gamma_j|\mathbf{I}_1(\mathbf{x_1}), \cdots, \mathbf{I}_C(\mathbf{x_C})) = \max(P(\Gamma_k|\mathbf{I}_1(\mathbf{x_1}), \cdots, \mathbf{I}_C(\mathbf{x_C}))). \tag{6.1}$$

[1] Note that by taking only one pixel per view for each voxel, we are implicitly considering a very simple, though common, projection test. In the following chapter, several projection tests will be presented so that this and other schemes can be generalized.

[2] The prior probability of detecting a foreground voxel can be simply obtained by computing the detected voxel / total voxel occupancy ratio using conventional SfS, for instance. $P(\phi)$ and $P(\beta)$ are obtained from P_S: $P(\phi) = \sqrt[C]{P_S}$ and $P(\beta) = 1 - P(\phi)$. Priors can also be set after studying the particularities of each set-up, setting low foreground priors where activity is unlikely, for instance.

In stereo-vision, i.e., with cameras positioned over a short baseline, views have high correlation between them. However, it is reasonable to assume that camera views are statistically independent among them in environments with a scatter of cameras around the scene, which is the case that is going to be considered here.

Thus, assuming here and in the rest of the chapter that the super classes are conditionally independent to the views, and using the Bayes theorem:

$$P(\Gamma_k|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C})) = \frac{P(\Gamma_k)\prod_{i=1}^{C} p(\mathbf{I}_i(\mathbf{x_i})|\Gamma_k)}{p(\mathbf{I}_1(\mathbf{x_1}))\cdots p(\mathbf{I}_C(\mathbf{x_C}))}, \tag{6.2}$$

where $p(\mathbf{I}_i(\mathbf{x_i})|\Gamma_k)$ is the likelihood of the observation in camera i, given a certain super class in a given camera i. For instance, given $\Gamma_2 = \{\phi,\beta,\phi,\cdots,\phi\}$, likelihoods $p(\mathbf{I}_1(\mathbf{x_1}))$ and $p(\mathbf{I}_2(\mathbf{x_2}))$ are

$$\begin{aligned} p(\mathbf{I}_1(\mathbf{x_1})|\Gamma_2) &= p(\mathbf{I}_1(\mathbf{x_1})|\Gamma_2[1]) = p(\mathbf{I}_1(\mathbf{x_1})|\phi) = \frac{1}{256^3} \\ p(\mathbf{I}_2(\mathbf{x_2})|\Gamma_2) &= p(\mathbf{I}_2(\mathbf{x_2})|\Gamma_2[2]) = p(\mathbf{I}_2(\mathbf{x_2})|\beta) = G_{2,\mathbf{x_2}}(\mathbf{I}_2(\mathbf{x_2})). \end{aligned}$$

Substituting (6.2) into (6.1) we finally obtain the decision rule

$$\Gamma_j = \operatorname*{argmax}_{\Gamma_k} P(\Gamma_k)\prod_{i=1}^{C} p(\mathbf{I}_i(\mathbf{x_i})|\Gamma_k[i]). \tag{6.3}$$

Or in terms of a posteriori probabilities

$$\Gamma_j = \operatorname*{argmax}_{\Gamma_k} P(\Gamma_k)\prod_{i=1}^{C} \frac{P(\Gamma_k[i]|\mathbf{I}_i(\mathbf{x_i}))}{P(\Gamma_k[i])}, \tag{6.4}$$

which is equivalent to

$$\Gamma_j = \operatorname*{argmax}_{\Gamma_k} P(\Gamma_k)^{1-C}\prod_{i=1}^{C} P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})), \tag{6.5}$$

where $P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i}))$ is the probability of a super class, given a certain observation. For instance, given $\mathbf{I}_2(\mathbf{x_2})$, the probability of super class $P(\Gamma_{C+1})$ is

$$\begin{aligned} P(\Gamma_{C+1}|\mathbf{I}_2(\mathbf{x_2})) &= P(\beta)P(\beta|\mathbf{I}_2(\mathbf{x_2}))P(\phi)^{C-2} \\ &= P(\beta)\frac{P(\beta)G_{2,\mathbf{x_2}}(\mathbf{I}_2(\mathbf{x_2}))}{p(\mathbf{I}_2(\mathbf{x_2}))}P(\phi)^{C-2}, \end{aligned}$$

where $p(\mathbf{I}_2(\mathbf{x_2}))$ is the unconditional joint distribution of pixel $\mathbf{x_2}$ in view $\mathbf{I}_2$ (see §4.4, in chapter 4).

Both (6.3) and (6.5) decide the most probable super class. However (6.3) can be used to obtain faster classifications, even though the probabilities are not explicitly computed.

6.3.1 Discussion

Note that the decision rule is very strict in the sense that a single misclassification in a view inhibits a correct interpretation of the process occurred. Misclassifications are specially sensible in the case of super class Γ_0, since a single misdetection of a ϕ class will let a erroneous 3D background detection. On the contrary, misclassifications in a 3D background super class often will lead to another 3D background super class, which is not a severe problem. A more in-depth analysis of this biased behavior to error types is given in the following chapter.

In order to prevent such type of errors, we can force the classifiers not to deviate from the prior probabilities. This can be done with two different interpretations of the problem:

1. Considering an outlier model in the 2D models [Aug03].
2. Assuming that $P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) = P(\Gamma_k)(1+\delta_{ki})$ [KHDM98].

Both interpretations are discussed in the following sections.

6.4 Probabilistic Voxel Classification Considering Outliers in the 2D Model

If we consider that the 2D model has an associated probability of outlier e, then we can use the prior probability when the model fails

$$P'(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) = eP(\Gamma_k) + (1-e)P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})), \tag{6.6}$$

and then,

$$\begin{aligned} &P'(\Gamma_k|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C})) = \\ &\prod_{i=1}^{C}\left(eP(\Gamma_k) + (1-e)P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i}))\right). \end{aligned} \tag{6.7}$$

A Taylor expansion in f around 0, after replacing variables $f = (1-e)$, gives

$$\begin{aligned} P'(\Gamma_k|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C})) &= (eP(\Gamma_k))^C + \\ + (eP(\Gamma_k))^{C-1}(1-e)\sum_{i=1}^{C}P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) &+ O((1-e)^2). \end{aligned} \tag{6.8}$$

If e is close to 1, then only the first two terms matter. This is a rather strong assumption but it may be satisfied when observed data is highly ambiguous. In cases not so ambiguous, (6.7) has to be used instead.

Under this assumption, super class Γ_j is chosen using the following decision rule

$$\Gamma_j = \underset{\Gamma_k}{\operatorname{argmax}} \left((eP(\Gamma_k))^C + (eP(\Gamma_k))^{C-1}(1-e) \sum_{i=1}^{C} P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) \right). \tag{6.9}$$

6.4.1 Probabilistic Voxel Classification Considering Non-Deviated Posteriors

A similar result to (6.9) can be obtained expressing a posteriori probabilities as

$$P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) = P(\Gamma_k)(1+\delta_{ki}), \tag{6.10}$$

where $\delta_{ki} \ll 1$. This expression assumes that a posteriori probabilities computed by the respective classifiers will not deviate dramatically from the prior probabilities [KHDM98].

Substituting (6.10) into (6.5), and neglecting terms of second and higher order we obtain

$$\Gamma_j = \underset{\Gamma_k}{\operatorname{argmax}} \left((1-C)P(\Gamma_k) + \sum_{i=1}^{C} P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i})) \right). \tag{6.11}$$

Note that both interpretations described in (6.9) and (6.11) convert the product ($\prod_{i=1}^{C} P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i}))$) in (6.5) into a sum ($\sum_{i=1}^{C} P(\Gamma_k|\mathbf{I}_i(\mathbf{x_i}))$). Interestingly, this is the probabilistic justification of the approach taken by some practitioners in voxel-based SfS, consisting in letting a voxel reconstruction with only a partial sum of $C-P$ foreground projections, instead of requiring total intersection (see §7.1.3 in the following chapter for more details on this).

6.5 2D Model Update

Once the voxels have been classified with any of the previously discussed procedures, the resulting voxels probabilities are projected to all the views. Note that when the probabilities are projected, special care has to be taken so that pixels are assigned with correct foreground probabilities. We propose to assign pixels the highest foreground probability value among all voxels whose projection belongs to the pixel. Additionally, the corresponding foreground probability that is projected from 3D to 2D has to be adapted to the change of dimensionality:

$$P(\phi_i|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C})) = P(\phi|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C})) = \sqrt[C]{P(\Gamma_0|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C}))}, \tag{6.12}$$

assuming that all the views contributed to the voxel with identical probabilities. This probability can be used to update the 2D background models described in chapter 4, §4.4. In the 2D MAP setting described in the mentioned chapter, background models are updated according to their background probabilities ($P(\beta|\mathbf{I_x})$). The Bayesian setting of both approaches let us easily incorporate this 3D extra probabilistic information to the models update process by redefining the $P(\beta|\mathbf{I_x})$ as follows:

$$P'(\beta|\mathbf{I_x}) = P(2D)P(\beta|\mathbf{I_x}) + (1-P(2D))(1-P(\phi|\mathbf{I}_1(\mathbf{x_1}),\cdots,\mathbf{I}_C(\mathbf{x_C}))), \tag{6.13}$$

where $P(2D)$ is a design parameter (a prior) that determines the influence that 3D information has into the 2D model update process (a value of $P(2D) = 0.5$ has proved to work well in our experiments).

Projecting back 3D probabilities permits to update 2D background models with higher precision. In chapter 4 we proved that, based on EM, the background models should be updated proportionally to the probability that an observation belongs to the background. Thus, the equations derived here are important to provide more robust learning speeds based on the information acquired from multiple cameras. Note that better adaptation speeds also permit to obtain better 2D background models. In this scheme, the background models are constantly updated making use of the redundancy present in a multi-camera system. In addition, the framework presented here can be extended to incorporate other 3D probabilistic values obtained using other techniques. In this regard, the method developed in the following chapter will be used to refine the 3D map obtained with the presented method, leading to even better 2D/3D foreground detections.

6.6 System Implementation

When using a large number of cameras, the class of maximum probability has to be found in a large search-space (K), and computational costs may be too high for certain applications. If this is the case, one can compute $P(\Gamma_0 | \mathbf{I}_i(\mathbf{x_i}), i = 1, \cdots, C)$ and set a threshold on the probability of the 3D-Shape. The probability of the 3D-Shape (P(Γ_0)) can be obtained using (6.2) when working with reliable 2D models, or with (6.7) when considering a certain probability of outliers (e) in the 2D models.

Threshold selection is performed only once per each different type of working environment. The threshold can be simply obtained by inspection of original image confronted to the projected probabilities (see Figure 6.1(a) and (c)). Similarly as stated in §6.5, note that when the probabilities of the 3D-Shape are projected, special care has to be taken so that pixels are assigned the highest probability value among all voxels whose projection belongs to the pixel. Note that this threshold can be set with very high precision, since probabilities are numbers defined in $\mathbb{R}$. On the contrary, in classical SfS, thresholds have to be set in the realm of integer numbers $\mathbb{Z}$, i.e., one has to decide the minimum number of foreground projections in 2D that form a voxel in 3D.

Finally, it has to be remarked that the most reliable classification, with a Bayesian justification, is done using (6.3), when considering error-free 2D models and (6.9) or (6.11), when considering an error model. The drawback is that the probabilities of all the 3D background super classes, which we are not interested in, will have to computed.

6.7 Results

The proposed scheme has been evaluated using 5 synchronized video streams, captured and stored in JPEG format, in the smart-room of our lab at the UPC. Apart from the compression artifacts, the imaging scenes also contain a range of difficult defects, including illumination changes due to a beamer and shadows. Our system has dealt with all these problems successfully, improving the results of conventional 2D-segmentators and standard SfS reconstruction methods.

Figure 6.1 shows an example in a certain view and instant. In this example, we have used foreground priors equal to 0 in those regions which are within $0.4m$ of the walls. The original image (a) can be compared to the resulting mask after performing a conventional 2D-foreground segmentation in (b) and a cooperative 2D-foreground segmentation in (d). In the example, the outlier model in (6.7), without further simplifications is used. In this example, we have used $e = 0.5$. The classification is performed setting a threshold to the probability of 3D-foreground by inspection of (c), as discussed in the previous section.

Inspection of silhouettes (b) and (d) shows that the 2D models learned in the cooperative approach are clearly better than those which are learned using a single-view approach. Note that the silhouettes in (d) do not have as many holes as in (b). Also, in (d) there is not as much clutter as in (b) in the areas close to the walls. Since these areas are out of study, the pixels corresponding to the visible zones close to the walls are given high background probabilities and their corresponding background models are updated with the color information that is observed. In addition, the shadow cast over the floor by the person on the right part of the image was removed. However, this particular improvement cannot be attributed to the proposed method. In fact, the main reason for which this region was integrated into the background is that two out of the five cameras did not capture the part of the room where the shadow was cast since the table was occluding it. Thus, this part was simply not reconstructed (see (c)). Occlusions often introduce this type of effects, which can be detected by checking the 2D/3D consistency. An in-depth analysis of these aspects will be given in the following chapter.

6.8 Conclusion and Future Work

In this chapter, we have presented an improvement for the current 2D and 3D foreground segmentation techniques. The presented method is able to segment the foreground in a view using the evidence present in the rest of cameras.

The work presented here and in chapter 4 form a unit in the sense that both approaches can cooperate in a Bayesian context. As described before in this chapter, the cooperative framework treats 2D probabilities as input values. This improves the 3D classification in case of image noise or background model failures. Once the 3D space is classified, the 3D probabilistic

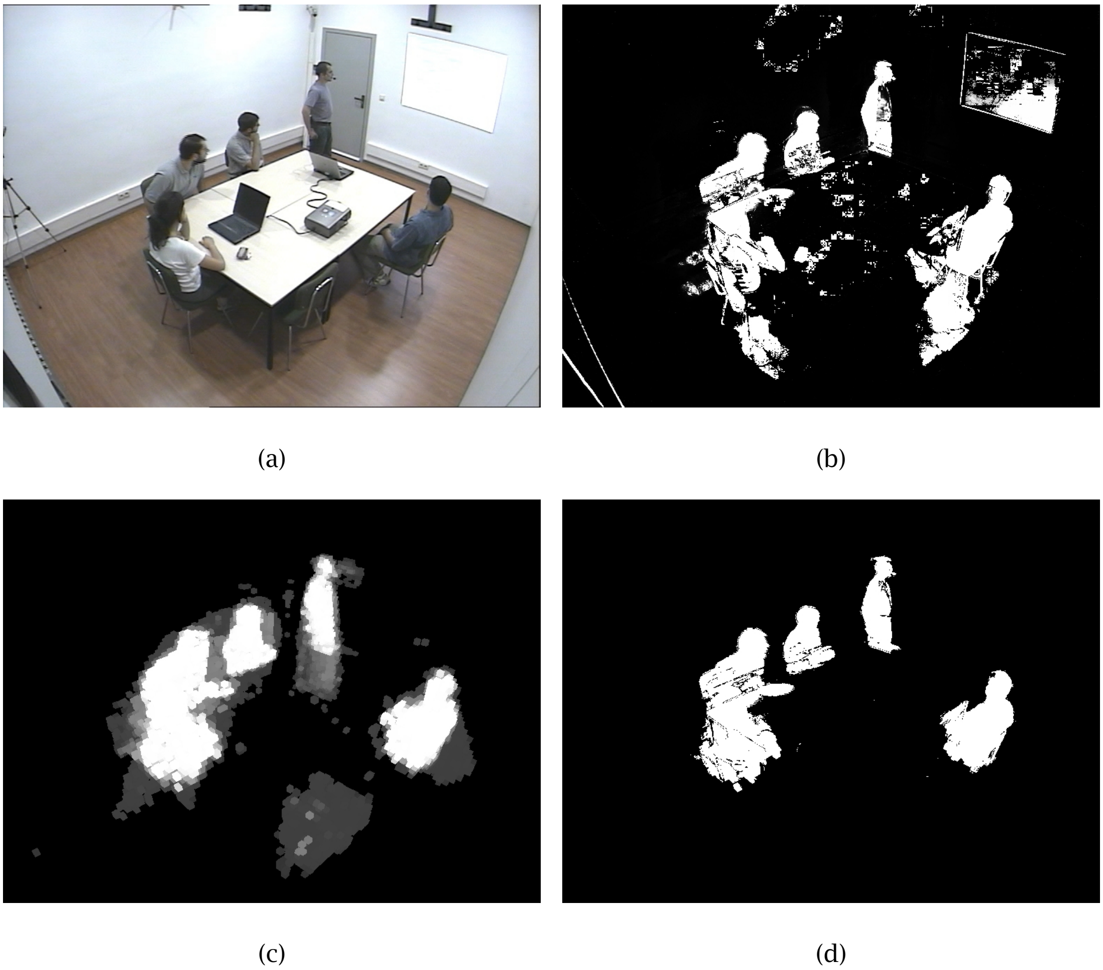


Figure 6.1: The original image is shown in (a). Picture (b), shows the foreground segmentation using conventional Bayesian classification as explained in chapter 4. The segmentation has also been postprocessed to remove shadows as described in §5.4.1, in chapter 5. However, it can be observed that strong shadows were not removed. In (c), the projected probabilities of the 3D-Shape are shown in gray scale. Finally, image (d) shows the foreground segmentation using the cooperative framework.

evidence can be transferred back to 2D images. 2D models are updated based on the probabilities that the pixels form part of the background or foreground. Therefore, it is fairly easy to incorporate 3D information to this update process by projecting these 3D probabilistic datum.

The proposed scheme also permits setting fine-grained thresholds for 3D space classification since the 3D space is classified making use of 2D probabilities instead of 2D *foreground* or *background* binary information. And, being a Bayesian approach, it is also possible to set priors in the 3D space without occlusion problems.

The Bayesian setting presented has proved to work well. However there are a set of problems which this technique cannot alleviate. When errors are systematic, that is, some areas which are completely missed in certain views consistently along the time, then the technique is not able to detect the problem. Indeed, there are other studies that are able to detect some 2D-classification errors based on the geometric constraints of the reconstructed Visual Hull [FVB03, LP07a, LPC06, Won01]. In chapter 7 we present an approach in this direction. The integration of both methods is discussed in §7.5 on page 143.

Finally and to conclude the chapter, in chapter 4 we showed the close relation between the pixel model update and pixel background probability. This relation can be exploited by using more-informed probabilistic sources information that are external to the pixel process. In this chapter, we have proposed to use the projection of probabilistic 3D maps that are created from 2D views. This has allowed to bridge the gap between the planar and volumetric foreground detection tasks, unifying the algorithms presented in this chapter and the ones in chapter 4. Moreover, the framework can be extended to incorporate external 3D probabilistic values which are obtained using other set of techniques. In this respect, the method developed in the following chapter will be used to refine the 3D map obtained with the presented method, leading to even better 2D/3D foreground detections.

Chapter 7

Shape from Inconsistent Silhouette

Shape from Silhouette (SfS) is the general term used to refer to the techniques that obtain a volume estimate from a set of binary images. In a first step, a number of images are taken from different positions around the scene of interest. Later, each image is segmented to produce binary masks, also called silhouettes, to delimit the objects of interest in each view. Finally, the volume estimate is obtained as the maximal volume which could be the cause of the observed silhouettes. The set of silhouettes is usually considered to be consistent which means that there exists at least one volume which completely explains them. The Visual Hull (VH) is then defined as the maximal volume which yields the set of consistent silhouettes. Since SfS is normally used assuming consistency in the silhouettes, it is also often considered that the VH is the result of the SfS algorithm. However, silhouettes are normally inconsistent due to inaccurate calibration or erroneous 2D silhouette extraction techniques. In spite of that, SfS techniques tend to reconstruct only the part of the volume which projects consistently in all the silhouettes. The part of the shape which cannot explain the silhouettes is left not reconstructed, which is one of the reasons why 3D misses happen more often than 3D false alarms. In this chapter, we extend the idea of SfS to be used with sets of inconsistent silhouettes. We propose a fast technique for estimating that part of the volume which projects inconsistently and propose a criteria for classifying it either as part of the shape or not by minimizing the probability of misclassification. A number of theoretical and empirical results, with synthetic and real-world images, are given, showing that the proposed method reduces the probability of volumetric misclassification and giving evidence that the method deals with false alarms and misses in an unbiased way.

7.1 Introduction

Shape extraction from a set of silhouettes (binary masks of the objects of interest in the foreground scene) was firstly introduced by Baugmart [Bau74] in 1974, though it was not until

1991 when Laurentini [Lau91] defined the geometric concept of Visual Hull (VH) as the maximal object silhouette-equivalent to the real object S, i.e., which can be substituted for S without affecting any silhouette [BL03, Lau94, Lau95]. Since then, Shape from Silhouette (SfS) has been considered as the method of obtaining the VH of an object.

The concept of 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, that is, the VH does not exist. Total consistency hardly ever happens in realistic scenarios due to inaccurate calibration or noisy silhouettes caused by errors during the 2D detection process: background learning techniques [EDHD99, HHD99, HHD00, KS00a, LHGT04, MD02, MJD$^+$00, SG00b, WADP97], chroma key techniques [CCSS01, SB96], etc. In spite of that, SfS methods have been designed in the past assuming that the silhouettes are consistent, thus reconstructing only the part of the volume which projects consistently in all the silhouettes, i.e., the volume where the visual cones intersect, without further considerations.

We propose a shape reconstruction method based on the silhouette consistency principle. Our system validates the regions in the silhouettes which are consistent in all the projections and adjusts the regions which are not, dealing with 2D errors, i.e., misses and false alarms, in an unbiased way. By contrast, other SfS systems usually treat differently the 2D errors on the basis of their type.

In the following, we summarize the different techniques available for extracting shapes from a set of silhouettes. Then, we discuss which are the different types of 2D errors and how they affect the reconstructed shape.

7.1.1 Shape from Silhouette

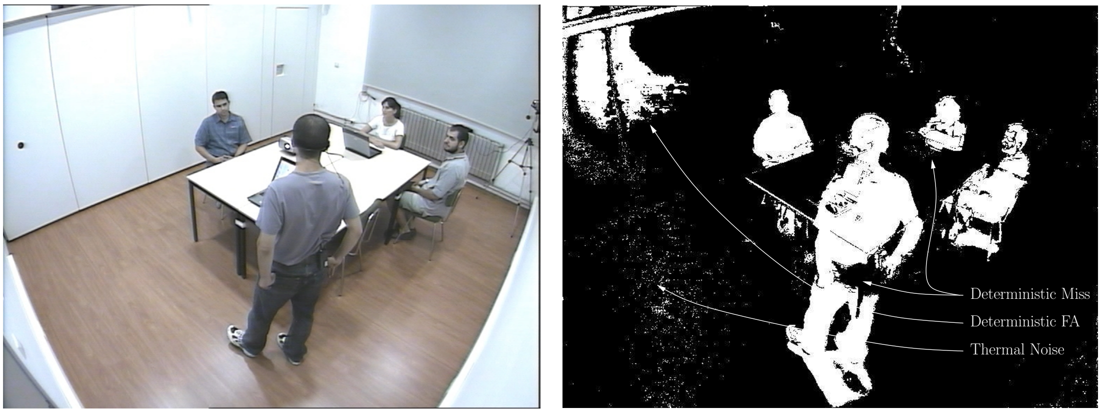
Many algorithms have been developed for constructing volumetric models from a set of silhouette images (see §2.2.3, on page 25). Silhouette images are first extracted by creating statistical models of the background process of every pixel value, i.e., colour [EDHD99, FR97, HHD00, JDWR00, SG00b, WADP97], texture [JS02, LL02, LPX05b, XLL04], or temporal-based information [LHGT02, Wix00]. Then, the foreground segmentation is performed at each pixel, either as an exception to the modeled background [EDHD99, HHD99, HHD00, MJD$^+$00, SG00b, WADP97], or in a Bayesian framework, using a maximum a posteriori classifier as the one presented in chapter 4. Once the silhouettes are extracted, the main step of all the algorithms is the intersection test. Geometric solutions back-project the silhouettes, creating an explicit set of cones that are then intersected in 3D [GHF86, MBR$^+$00, RS97]. Voxel-based solutions divide the volume into voxels [CKBH00, LP05, LP06, MKKJ96, MTG97, SVZ00]. Then each voxel is projected into all the images to test (using a projection test) whether they are contained in every silhouette. More efficient octree-based strategies have also been used to test voxels in a coarse to fine hierarchy [Pot87, Sze93]. See [Dye01, SCMS01] for two surveys on volumetric-based methods.

Accurate silhouette extraction is crucial for good performance of SfS, independently of the algorithm used. In the following, we discuss how errors in the silhouettes affect the reconstructed shape. Based on the outcomes of the issues discussed, a more indepth analysis of the proposed 3D-reconstruction technique will be possible.

7.1.2 Noise Propagation to the 3^{rd} Dimension

Silhouette image noise can be classified in different ways, e.g., according to the observable effects over the silhouettes; or depending on the cause that produced the error:

- Defects observable in the silhouettes can be categorized into two types: false alarms and misses. False alarms correspond to erroneous foreground detections, while misses correspond to erroneous background detections.
- Errors in the silhouettes can be due to different causes: regular noise and non-Gaussian systematic errors. The first type of error is because of the cameras thermal noise, while the second one often consists in large regions missed or falsely detected due to the arrangement of the scene. Systematic misses in a view often occur when, for instance, foreground objects have similar colors and texture to their counterparts in the background. Systematic misses can also be due to background structures, such as the table in Figure 7.1, occluding the foreground objects. Analogously, specular reflections can form large areas of falsely detected pixels.

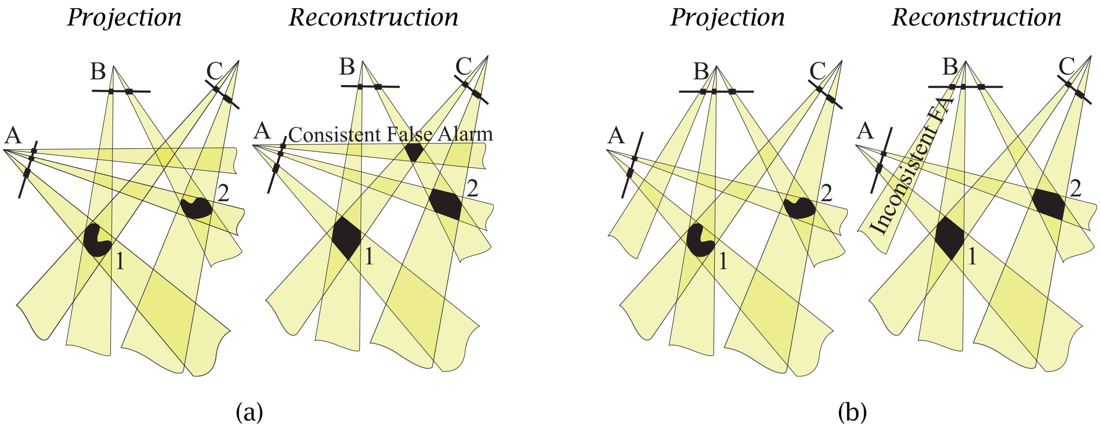


(a) (b)

Figure 7.1: Original image and segmented silhouette showing the different types of errors.

Each technique in the literature has been focused on reducing the effects of either the systematic or the Gaussian nature of the errors. However, since both of them can produce the same effects (false alarms and misses), the study of 3D error propagation can be isolated from the cause.

In SfS, a false alarm in a view does not contribute to a false alarm in 3D unless the visual cone that is erroneously created intersects simultaneously with other $C-1$ visual cones, where C is the total number of cameras (see Figure 7.2(a)). If the intersection is produced, then the volumetric points corresponding to the intersection are wrongly reconstructed. Since the reconstructed shape is consistent because its projection in all the views matches with the silhouettes, then the 2D false alarm is undetectable. However, the shape is not reconstructed in the parts of the volume where at least one of the erroneous visual cones does not intersect simultaneously with other $C-1$ visual cones (see Figure 7.2(b)). This is the most typical case in scenarios where the major part of the volume is unoccupied. In such case, the cones produced by 2D false alarms do not intersect with visual cones from the rest of cameras. Thus, 2D false alarms are inconsistent with the reconstructed shape, allowing their detection as we will show in the following sections.



(a)

(b)

Figure 7.2: In (a) there has been a false detection in camera A. The false visual cone intersects with other $C-1$ visual cones forming a false shape reconstruction. Another false alarm in camera B is depicted in (b). In this case, the false alarm forms an inconsistent cone for not intersecting with other $C-1$ visual cones. This type of false alarm, which is the most common case, does not affect SfS reconstructions.

Contrarily, a miss in a view inhibits the simultaneous intersection of C visual cones in 3D, leading to an ineluctable miss in the shape (see Figure 7.3). This makes the SfS algorithm highly sensitive to this type of errors, whereas 2D false alarms do not produce erroneous reconstructions in most of the cases. 2D misses can also be indirectly detectable, since the projection of a wrongly unreconstructed shape will not match with the rest of correct silhouettes.

As a final thought on the effects of 2D error propagation, it seems clear that the very sensitive response of SfS to 2D misses contradicts the general notion that “as the number of cones increases, the object is reconstructed with higher precision" [Lau94]. While this is true

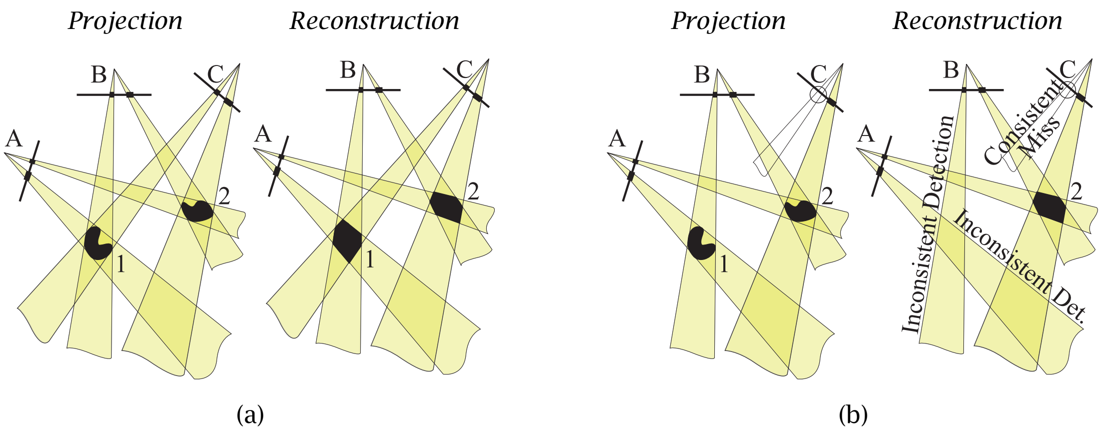


Figure 7.3: In (a), objects 1 and 2 are correctly detected in all the cameras. In (b), object 1 has been missed in camera C. On the right the Visual Hull is depicted. Note that the visual cones which do not intersect with any reconstructed shape are considered to be inconsistent with respect to the Visual Hull.

with perfectly extracted silhouettes, it is not the case when the silhouettes have *non null rates of miss*. In fact, an infinite number of silhouettes with a low but non null rate of randomly distributed misses will not reconstruct any shape (see Figure 7.4[1]). In conclusion, although SfS algorithms are perfectly fine with consistent silhouettes, they tend to penalize 2D misses in front of 2D false alarms when the silhouettes are inconsistent. The Shape from an Inconsistent set of Silhouettes (SfIS) has to be based on a different principle; one that takes decisions in accordance with the probabilities of 2D false alarm and miss; and one which does not imply that the Shape lies only in the intersection of *all* the visual cones.

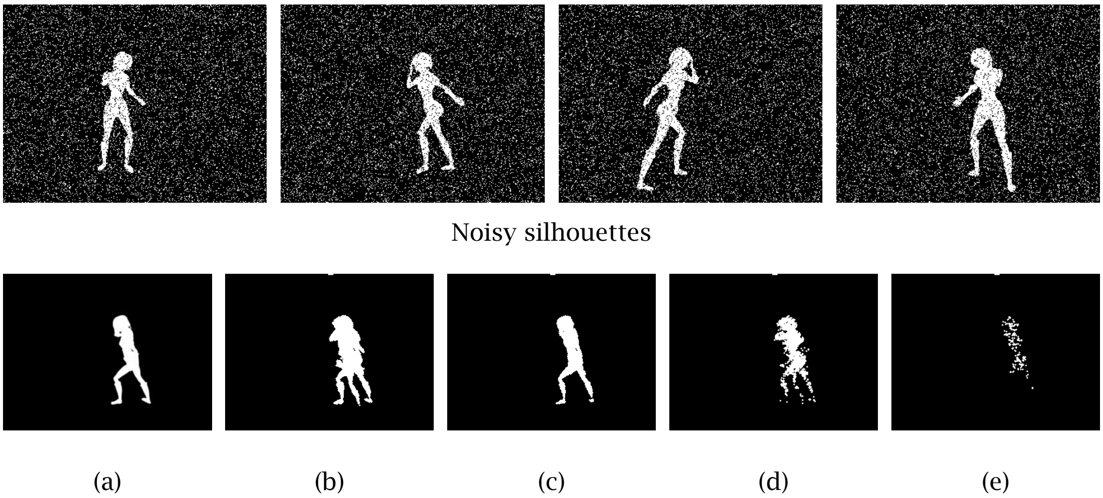
Indeed, SfIS might introduce more false alarms to the Shape than SfS as the payoff for recovering some of the misses. We will show that false alarms will be introduced only to the extent that global error is lower than without them.

7.1.3 Dealing with Noise in Related Works

In the past, efforts have been put in proposing different algorithms for reducing the effects of the propagation of the two-dimensional noise. There are essentially three sorts of approaches to achieve noise reduction.

The first general approach involves using voxel-based reconstructions to reduce the probability of voxel misclassification. In [CKBH00], Cheung et al. propose an algorithm called SPOT.

[1] The *Kung-Fu Girl* dataset is provided by the *Graphics Optics Vision group* of *Max-Planck-Institut fur Informatik*. The dataset consists of frame sequences rendered from 25 different camera views located in a hemisphere around the scene, background images and camera parameters. The recorded scene contains a humanoid figure, animated by motion-captured data. Each frame is saved as a 320x240 image.



Noisy silhouettes

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

Figure 7.4: VH reconstruction and projection.
The row on the top shows some of the original silhouettes which have been used to create Visual Hulls using different approaches. Note that we have added artificial random noise ($P(FA) = P(M) = 0.1$) to the silhouettes. Figure (a), in the bottom row, depicts a silhouette seen from another camera, which has not been used during the construction of any VH. The purpose of silhouette (a) is to compare the projections in this view of the different VHs:
In (b) and (c), 4 and 8 *noise-free* silhouettes have been used to build the VH, respectively. Note that the projected VH in (c) is more accurate, due to the larger number of cameras being used. Figures (d) and (e) show the projection of the VH reconstructed using 4 and 8 *noisy* silhouettes, respectively. This time, note that large parts of the Shape have been missed in the VH, due to 2D misses. The VH specially worsens when more cameras are used. In contrast, the figures also show that 2D false alarms hardly have any effect on the 3D Shape.

In their approach, the voxels are projected into each camera view. Then, their algorithm determines the minimum number of foreground pixels (Z_ϵ) which have to be detected inside each projection of a voxel to consider that the projection test is passed in a certain view. Finally, if the projection test is passed *in all the views*, then the voxel is classified as part of the Shape. The minimum number of foreground pixels Z_ϵ, over the total Z, is determined after minimizing the probability of voxel misclassification considering that the silhouettes are *consistent* (i.e., that a voxel is part of the shape if and only if the projection test is passed in all the views, while it is background otherwise). So, on the one hand, SPOT considers that the masks are consistent while, on the other hand, it accepts that the masks are inconsistent for having misses and false alarms. Under the same assumption of silhouette consistency, SPOT achieves lower voxel misclassification rate compared to other SfS algorithms that use naive projection tests such as testing only one point per voxel and view or testing all the pixels within the projection of the voxel. But even though SPOT and other voxel-based noise reduction methods are an important step forward, none of them have focused on the detection of systematic errors.

The second general approach, used in [LP05] and proposed in [SVZ00] as a reference for comparison with their proposed method, requires the intersection of at least $C - P$ visual cones to allow a reconstruction, where P is the number of acceptable misses among the set C of cameras. Although single misses do not block the reconstruction in this approach, the resulting shape is larger than the real Visual Hull for requiring fewer intersections of visual cones. A drawback of this approach is that larger hulls are reconstructed either if the silhouettes are consistent or not.

The last general approach uses multi camera information in terms of consistency constraints, providing tools for detecting systematic errors. In [FVB03, Won01] the epipolar tangency constraint (testing correspondences of the frontier points) is used as a necessary condition for shape consistency. However, the authors discard using the area of each silhouette that lies outside the visual hull for being slow and not suitable for pose estimation [FVB03].

Our approach is placed in the later context. We propose a fast technique for estimating that part of the volume which projects inconsistently and propose a criteria for classifying it either as part of the shape or not by minimizing the probability of voxel misclassification. Our approach is voxel-based. However we propose a general framework where any projection test can be used.

The remainder of the chapter is structured as follows. In the next section, the voxel-based SfS approach is discussed. Section 7.3 is devoted to discussion of SfIS, including detailed algorithms for its implementation. Section 7.4 presents the conditions in which a very fast implementation of SfIS is possible and section 7.5 describes a collaborative framework between the SfIS and the cooperative Bayesian method presented in chapter 6. In section 7.6, theoretical and experimental studies of the system are presented with various synthetic and real-world test images and video sequences. Finally, the chapter concludes in section 7.7 with an overview of the main contributions presented.

7.2 Voxel-Based Shape from Silhouette

In SfIS, volume classification is achieved after minimization of the probability of misclassification. Since 3D errors depend on the probability of the 2D misclassification technique, it is important to first study which are these 2D-error probabilities. To do so, we focus on the voxel-based approach and discuss the probabilities of error of several projection tests. In addition, we give the probability of error of the voxel-based SfS approach, so that we can compare it later with the probability of error of SfIS.

The voxel-based SfS algorithm for any projection test is the one shown in Algorithm 2.

Algorithm 2 Voxel-based SfS algorithm

```
Require: Silhouettes: $S(c)$, a Projection Test Function: $PT_c(voxel, Silhouette)$
for all $voxel$ do
    $voxel \leftarrow$ Foreground
    for all $c$ do
        if $PT_c(voxel, S(c))$ is false then
            $voxel \leftarrow$ Background
        end if
    end for
end for
```

Since voxel classification errors may be due to either false alarms or misses, the probability that a voxel is misclassified is:

$$P(Err_{3D}) = P_B P(FA_{3D}) + P_S P(M_{3D}), \tag{7.1}$$

where $P_B = P(\beta)$ and $P_S = P(\phi)$ are prior probabilities of a voxel forming part of the Background or Shape, respectively[1], and $P(FA_{3D})$ and $P(M_{3D})$ correspond to the probabilities of false alarm and miss in a voxel. Note that we modify the notation from the previous chapters to reduce the complexity of the expressions that will be derived in the rest of the chapter. $P(M_{3D})$ will be used to denote $P(\hat{\beta}|\phi, \mathbf{P})$ for a certain 3D point $\mathbf{P}$. Also, $P_i(M_{2D})$ will be used to denote $P(\hat{\beta}|\phi, \mathbf{I}_i)$, corresponding to probability of miss of a certain projection test using the pixel values of the projected 3D point. Analog considerations apply to $P(FA_{3D})$ and $P(FA_{2D})$.

Since 3D false alarms happen when a voxel is wrongly classified as part of the Shape in all camera views, while misses happen when a voxel is wrongly classified as part of the Background in at least one camera view, we can write

$$P(Err_{3D}) = P_B \underbrace{\prod_{i=1}^{C} P_i(FA_{2D})}_{P(FA_{3D})} + P_S \underbrace{\left(1 - \prod_{i=1}^{C}(1 - P_i(M_{2D}))\right)}_{P(M_{3D})}, \tag{7.2}$$

[1] Priors P_S and $P_B = 1 - P_S$ can be simply obtained by computing the detected voxel / total voxel occupancy ratio, for instance.

where $P_i(FA_{2D})$ and $P_i(M_{2D})$ correspond to the probabilities that the projection test has been wrongly passed (false alarm) or wrongly failed (miss) in camera i, respectively.

Equation (7.2) can be expressed more compactly when the probabilities of false alarm and miss are equiprobable in all the views ($P_i(M_{2D}) = P_j(M_{2D})$ and $P_i(FA_{2D}) = P_j(FA_{2D})$, for all i and j):

$$P(Err_{3D}) = P_B \underbrace{P(FA_{2D})^C}_{P(FA_{3D})} + P_S \underbrace{\left(1 - (1 - P(M_{2D}))^C\right)}_{P(M_{3D})} \tag{7.3}$$

that besides being more compact, it is also significantly faster to compute than equation (7.2). In the following, we will refer to a test as equiprobable if it has equal error probability in all views, and non-equiprobable if it has different error probability in each view where the test is being carried out.

In the following, we describe some equiprobable and non-equiprobable projection tests. In particular, we rewrite and formalize some of the more popular projection tests in the literature, such as the Single Pixel and the Complete Pixel Projection Tests. We present them in a unified notation, making it possible an easier comparison of them. We also formalize the Incomplete Pixels Projection Test, which has been used in the past with several variants and names. Moreover, we propose a new projection test. This new projection test, which we have named as the Sampled Pixels Projection Test, has proved to be both fast and accurate when combined with the proposed reconstruction method that is described later in this chapter.

In addition, we derive the probabilities of voxel misclassification for all the projection tests, presenting all the expressions in a unified way. This step is a requisite for allowing us to formalize the reconstruction method presented later in this chapter with the more common projection tests in the literature and with the Sampled Pixels Projection Test that is presented here.

7.2.1 Single Pixel Projection Test

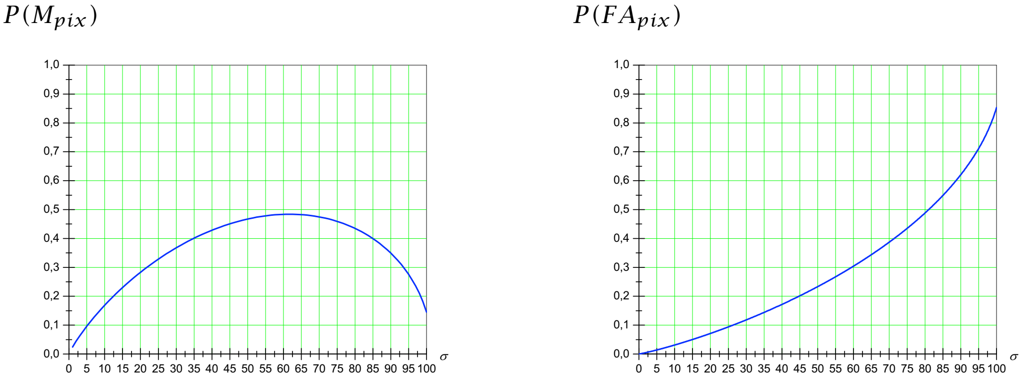
A very fast test consists in projecting the point in the center of a voxel into a pixel in all the camera views. Thus, the probabilities of false alarm and miss of the test are exactly the same as the probabilities of false alarm and miss of the 2D foreground classification technique: $P(FA_{pix})$ and $P(M_{pix})$, respectively. See, for instance, Figure 7.5 for the probabilities of pixel misclassification in an MAP-based foreground segmentation method.

The probabilities of pixel misclassification depend on the foreground segmentation scheme. As an example, the mentioned figure shows the probabilities of false alarm and miss of a pixel using a 1-Dimensional Gaussian to model the luminance of the background process in a pixel and a uniform pdf ($\frac{1}{256}$) to model the foreground as described in §4.3.2, on page 48. Then, the variances of the Gaussians of all the pixels in all the images are averaged at each instant to extract a global measure of the pixel error rate. Thus, the probabilities of pixel misclassification of MAP-based foreground segmentation methods can be expressed with respect to the averaged standard deviation of all Gaussians as follows:

$$P(FA_{pix}) = P(\hat{\phi}|\beta) = 1 - \text{erf}\left(\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma}{256}\right)}\right) \tag{7.4}$$
$$P(M_{pix}) = P(\hat{\beta}|\phi) = \frac{2\sqrt{2}\sigma}{256}\sqrt{-\ln\left(\frac{\sqrt{2\pi}\sigma}{256}\right)},$$

where the $\sigma_{\mathbf{x}}$ of each pixel (denoted with $\mathbf{x}$) is averaged over all the pixels in all the images $\sigma = \frac{\sum_{\forall \mathbf{x}} \sigma_{\mathbf{x}}}{\sum_{\forall \mathbf{x}} 1}$.

The expressions above were presented in detail in §4.3.2, chapter 4, on page 51. Note that $P(FA_{pix})$ and $P(M_{pix})$ could have not been calculated employing the traditional exception-to-background 2D foreground segmentation approach. Thus, in order to derive the previous expressions and the following ones, it is essential to adopt an MAP-based planar foreground segmentation scheme as the one we presented in chapter 4.



(a)

(b)

Figure 7.5: The probabilities of pixel misclassification depend on the foreground segmentation scheme. As an example, the figure shows the probabilities of false alarm and miss of a pixel using a 1-Dimensional Gaussian to model the luminance of the background process in a pixel and a uniform pdf ($\frac{1}{256}$) to model the foreground.

Since only one point is projected to all images, errors of the projection test are equiprobable in all views, and the probability of voxel misclassification can be computed using equation (7.3). Note that we have assumed a global error probability of the foreground segmentation method, not particularizing the error probabilities for each pixel. A more general version of this approach can be employed by considering different error probabilities for each pixel, and therefore using expression (7.2). However, for the sake of clarity, in the rest of the text we only consider a global error probability of the foreground segmentation method. The general version of all the expressions can be easily derived from the expressions we provide but it is important to take into account that more general expressions will often lead to non-real-time implementations of the reconstruction method presented later in this chapter.

7.2.2 Complete Pixels Projection Test

The Complete Pixels Projection Test consists in testing all the pixels (S_i) within the splat of the voxel in camera i, i.e., the number of pixels under the projection of the voxel in the i-th view. The test is passed only if all the pixels within the splat belong to the silhouette, that is, they are foreground pixels.

The size of the splat (S_i) can be estimated by projecting the sphere which contains the voxel, assuming that a sphere in the 3D world projects into a circle in the image plane (see Figure 7.6):

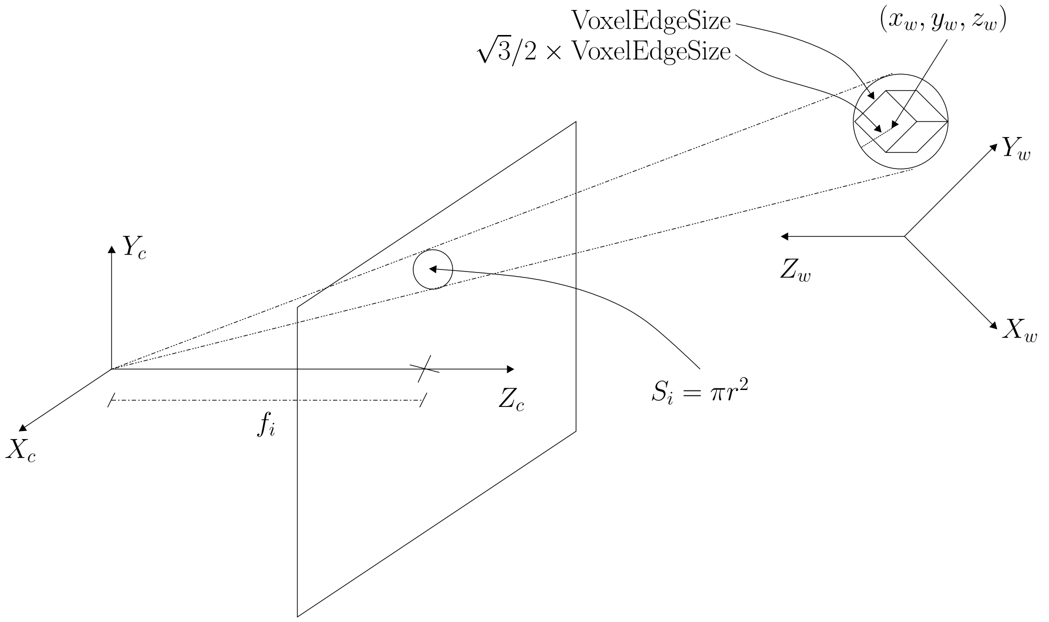


Figure 7.6: Size of the splat of a voxel in camera i.

$$
\begin{aligned}
S_i =& \pi r_i^2 \\
=& \pi \left(\frac{f_i \frac{\sqrt{3}}{2} \text{VoxelEdgeSize}}{[\mathbf{x_i}]_3} \right)^2 ,
\end{aligned}
\tag{7.5}
$$

where f_i corresponds to the focal length of camera i, and $[\mathbf{x_i}]_3$ denotes the third component of vector $\mathbf{x_i}$, which can be obtained by projecting the voxel center $[x_w \; y_w \; z_w]^T$ from the 3D world coordinate system to the image coordinate system:

$$\mathbf{x_i} = \left[\begin{array}{c|c} \mathbf{R_i} & \mathbf{t_i} \end{array} \right] \begin{bmatrix} x_w \\ y_w \\ z_w \\ 1 \end{bmatrix}, \tag{7.6}$$

being $\mathbf{R_i}$ and $\mathbf{t_i}$ the rotation matrix and translation vector of camera i, respectively.

The probability that the test is wrongly passed ($P(FA_{2D})$) is usually low, as this only occurs when all the pixels within the splat are falsely detected. However, the probability of a test-miss ($P(M_{2D})$) is significantly higher since the test is wrongly failed with at least one pixel-miss. Refer to table 7.1 on page 127 for the exact expressions of $P(M_{2D})$ and $P(FA_{2D})$.

Since a voxel is missed with at least one test-miss (which is very probable), voxel-misses happen very often, which is the reason for the rare use of the Complete Pixels Projection Test.

Finally, note that in order to compute the probability of voxel misclassification one has to use equation (7.2), since the projection test is non-equiprobable because it depends on the size of the splat in each view.

7.2.3 Sampled Pixels Projection Test

We have developed the Sampled Pixels Projection Test with SPOT [CKBH00] as principal inspiration. As with SPOT, a number of R points within the voxel are selected. These points may be equidistant among them, or just randomly selected. The test is passed in a view i when at least N projected points, i.e., pixels, over the total R, are within silhouette i.

$$\begin{array}{lcl} \text{Pixels in the Shape} \geq N & \Rightarrow & \text{pass the test} \\ \text{Pixels in the Shape} < N & \Rightarrow & \text{do Not pass the test} \end{array} \tag{7.7}$$

Selection of R points for each voxel makes the test very fast for two reasons:

The first reason is that the number of selected points is chosen independently of the voxel position, and therefore the probabilities of voxel misclassification are the same for all voxels.

The second reason is that since the test is run using exactly R pixels in each projection, the probabilities of false alarm and miss of the test are identical in all views. Thus, the probability of misclassification of a projection test has to be computed only for one view, and therefore $P(Err_{3D})$ can be estimated using faster equation (7.3), instead of (7.2).

The Sampled Pixels Projection Test here proposed differs from SPOT in the expressions used to calculate the probability of voxel misclassification. In SPOT it is assumed that priors P_B and P_S are equiprobable, which is almost never the case, being in some setups P_B several orders of magnitude larger than P_S. Another difference is that SPOT considers that a voxel-miss occurs only when *exactly* one projection test is wrongly failed which is computationally less complex (see equation (7.9)), while the Sampled Pixels Projection Test uses equation (7.3) which considers voxel-misses when the projection test is missed in *at least* one view (see equation (7.8)):

$$P(M_{3D}) = 1 - (1 - P_i(M_{2D}))^C$$

$$= \sum_{i=1}^{C} \binom{C}{i} P(M_{2D})^i (1 - P(M_{2D}))^{C-i} \qquad (7.8)$$

$$\neq P(M_{2D}) \sum_{i=1}^{C-1} (1 - P(M_{2D}))^i, \qquad (7.9)$$

In order to fully express equation (7.3), the probabilities of false alarm ($P(FA_{2D})$) and miss ($P(M_{2D})$) of the test have to be deduced. In the Sampled Pixels Projection Test, these probabilities depend on N in the following manner:

Since the test is passed when at least N pixels lie in the silhouette, false alarms of the projection test happen when there are at least N pixels falsely detected. Contrarily, misses of the projection test occur when there are at least $R - N + 1$ pixels missed.

Based on this reasoning, both misclassification probabilities have to add together the probabilities of all the possible cases which lead to a misclassification. Table 7.1 shows the precise mathematical expressions of $P(FA_{2D})$ and $P(M_{2D})$ of the test.

Once that $P(Err_{3D}[N])$ has been expressed, the following step is to choose the minimum number of points N over R which have to belong to the silhouette so that the test is passed. Indeed, the best N is the one which minimizes the probability of voxel misclassification:

$$N^\star = \operatorname*{argmin}_N P(Err_{3D}[N]) \qquad (7.10)$$

Since $P(Err_{3D}[N])$ is not continuous, it cannot be minimized by differentiating it. However, the optimal $N^\star$ can be obtained by doing an exhaustive search over all possible $N \in [0, R]$, as shown in Algorithm 3. Note that even though being computationally demanding, the calculation does not entail a problem since it only has to be performed once for all views and voxels.

Algorithm 3 Optimal $N^\star$

```
$MinPerr \leftarrow 1$
for all Possible $N$: $n = 0 \cdots R$ do
    if $P(Err_{3D}\,[N = n]) \leq MinPerr)$ then
        $N^\star \leftarrow n$
        $MinPerr \leftarrow P(Err_{3D}\,[N = n])$
    end if
end for
```

7.2.4 Incomplete Pixels Projection Test

We define the Incomplete Pixels Projection Test as the soft version of the Complete Test, in which not all the pixels within the splat of the voxel must belong to the silhouette to pass the test. Formally, the Incomplete Pixels Projection Test is passed in a view i when a minimum number of pixels M_i over all pixels belonging to the splat (S_i) lie in the silhouette:

$$\begin{array}{lcl} \text{Pixels in the Shape} \geq M_i & \Rightarrow & \text{pass the test} \\ \text{Pixels in the Shape} < M_i & \Rightarrow & \text{do Not pass the test} \end{array} \tag{7.11}$$

Similarly as with the Sampled Test, a false alarm of a test in a view i is produced when there are at least M_i pixels falsely detected over the total S_i, while misses are produced with at least $S_i - M_i + 1$ pixel misses over the total S_i. Refer to table 7.1 for the precise mathematical expressions of the probabilities of false alarm and miss of the test.

M_i can be chosen so that at least a fraction of pixels ($p = M_i/S_i$) belong to the silhouette. But even if the same p is forced in all views, the tests are not equiprobable, and therefore equation (7.2) has to be used to compute the probability of voxel misclassification.

One can also search the optimal $M_1^\star, M_1^\star, \cdots M_C^\star$ for each voxel so that $P(Err_{3D}[M_1, M_2, ..., M_C])$ is minimized also using equation (7.2):

$$\{M_1^\star, M_2^\star, \cdots, M_C^\star\} = \underset{M_1, M_2, \cdots, M_C}{\operatorname{argmin}} \; P(Err_{3D}[M_1, M_2, \cdots M_C]), \tag{7.12}$$

where an exhaustive search approach is unfeasible due to the very large size of the search-space where the solution has to be found:

$$Size = \prod_{i=1}^{C} S_i \tag{7.13}$$

Note that $Size$ depends on the position of the voxel with respect to the cameras. The distribution of the splat for the smart-room in our lab[1], using voxels of 3 cm edge size, has a mean of 55 pixels. Thus, $Size$ can be approximated to 55^5 in our working environment (see Figure 7.7[2]).

However, there exists a suboptimal solution for the M_i of the projection test of each view i:

$$M_i^\star = \underset{M_i}{\operatorname{argmin}} P(Err_{3D}[M_1 = M_i, \cdots, M_C = M_i, S_1 = S_i, \cdots, S_C = S_i]), \tag{7.14}$$

which can be solved using Algorithm 3 with $N = M_i$ and $R = S_i$.

The solution is suboptimal in the sense that we select in an optimal way the M_i in each view, assuming that the size of the splat is identical in the rest of views. But even though this

[1] The smart-room at the UPC includes 5 fully calibrated wide angle lens cameras with a resolution of 768 × 576 pixels. Video acquisition is done at 25 fps. Four of the cameras are positioned at the room corners, whereas the fifth camera is located at the ceiling. The room dimensions are 3966 × 5245 × 4000 mm.

[2] This figure has been a courtesy of J. Salvador, MSc student of the image group of the UPC.

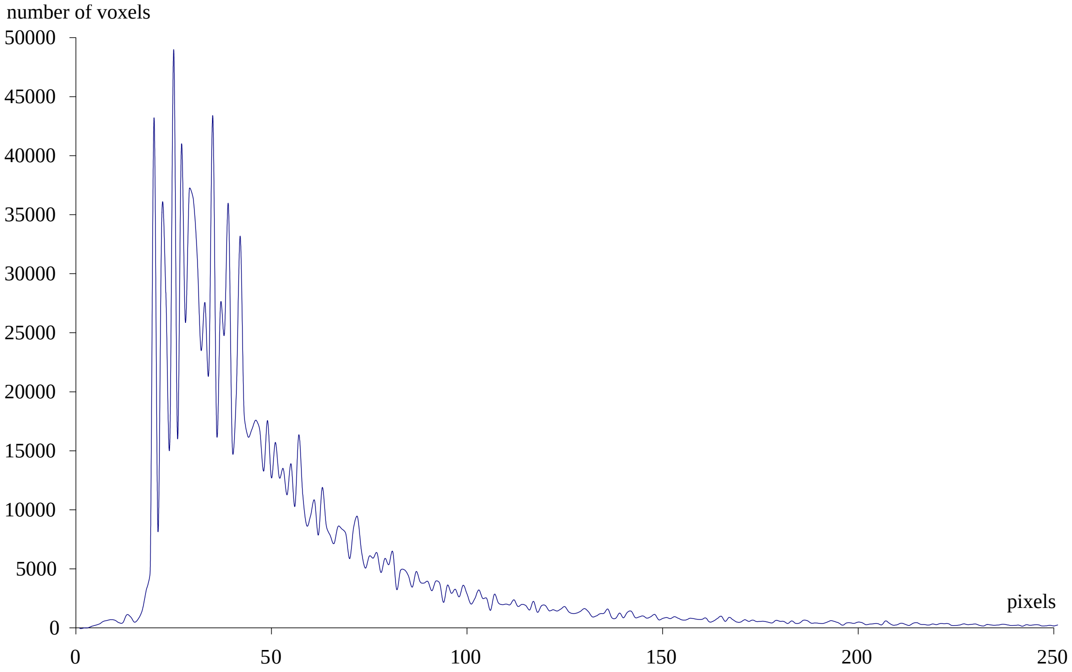


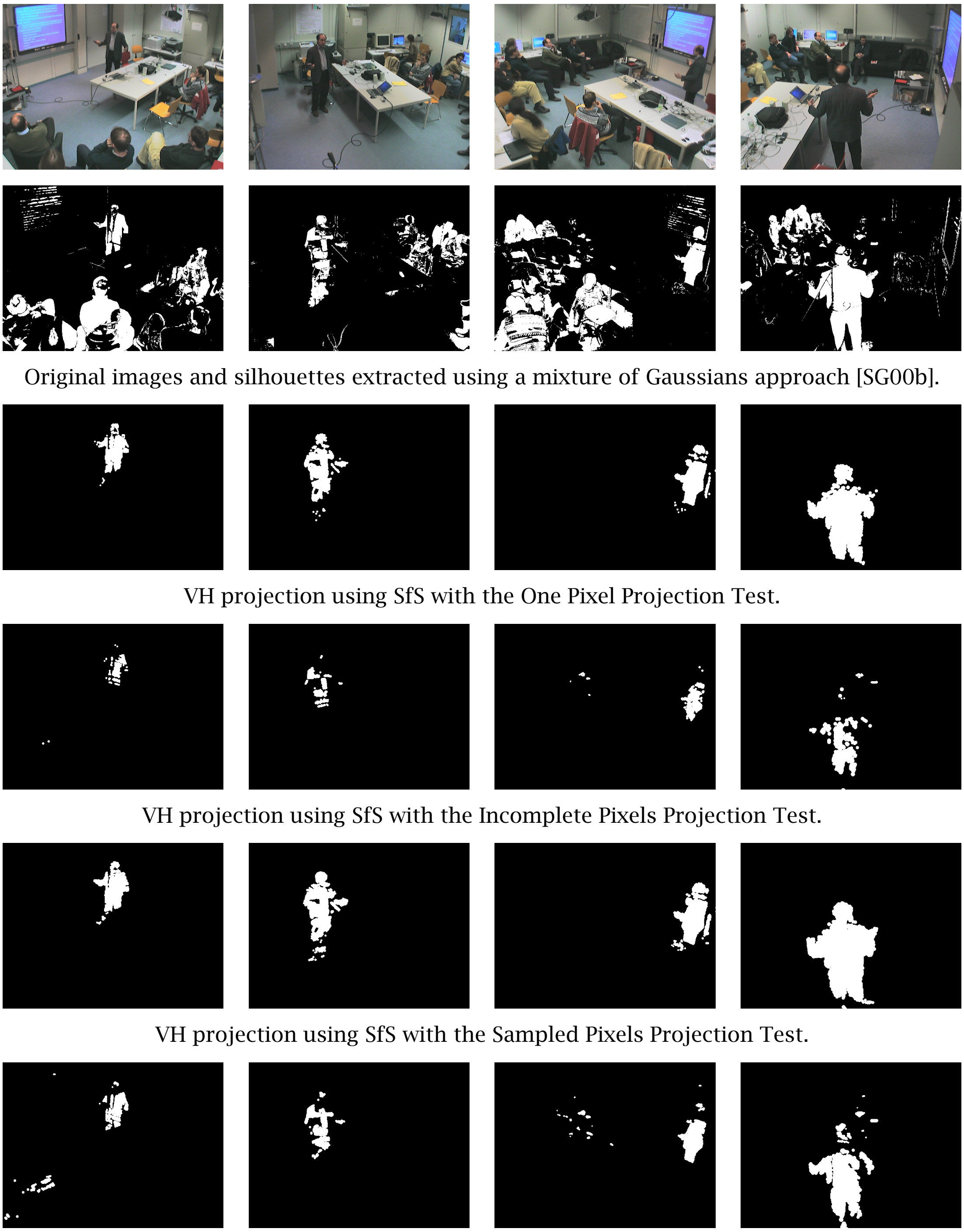
Figure 7.7: Distribution of the size of the splat, with a mean size of 55 pixels.

solution can be obtained much faster than the optimal solution, the set of resulting solutions $\{M_1^\star, M_2^\star, \cdots, M_C^\star\}$ may not be globally optimal.

Finally, in Figure 7.8[1] we depict some visual examples of the behavior of the presented projection tests. Note that One Pixel Test is not as good performer as the Sampled Pixels Test, in this example. However this approach is suitable for low-error systems focused to real-time operation. In the figure, the Complete Pixels Test is the worst performer. In fact it often has the highest probability of voxel misclassification when there are 2D misses. Finally, we also show the Incomplete Pixels Test using a fixed factor of $p = 80\%$ pixels. Note that the factor has not been set in an optimal way ($p \times S_i \neq M_i^\star$), which justifies the poor performance of the method in this case.

In summary, we have presented four different projection tests which can be combined with the standard voxel-based SfS algorithm. We have also provided the error probabilities of each test, and given some indications on the computational complexity of each one. In the following, we propose the SfIS algorithm, which also makes use of the proposed tests. In SfIS, the error probabilities of the employed test are important, and therefore Table 7.1 will be a

[1] The images correspond to the 2005 evaluation dataset used within the framework of the CHIL *Computers in the Human Interaction Loop* project [Com]. The images were acquired in the smart-room of the Interactive Systems Labs at the University of Karlsruhe, Germany. The setup includes 4 fully calibrated wide angle lens cameras with a resolution of 768 × 576 pixels, positioned at the room corners.



Original images and silhouettes extracted using a mixture of Gaussians approach [SG00b].

VH projection using SfS with the One Pixel Projection Test.

VH projection using SfS with the Incomplete Pixels Projection Test.

VH projection using SfS with the Sampled Pixels Projection Test.

VH projection using SfS with the Complete Pixels Projection Test ($p = 80\%$).

Figure 7.8: The first row of images shows the original images acquired in the smart-room of the Interactive Systems Labs at the University of Karlsruhe, Germany. The second row of images correspond to the set of extracted silhouettes. The rest of rows show the projection of the VH using the One Pixel, Incomplete Pixels, Sampled Pixels and Complete Pixels Projection Tests. All the VHs have been reconstructed in the area of the presenter, using voxels with edge size of 2.5 cm.

useful resource for the implementation of the algorithm.

Table 7.1: Projection Tests Error Probabilities

Type	Proj. Test	Error Probability	Equiprobable
F.A.: $P_i(FA_{2D})$	Single	$P(FA_{pix})$	✓
	Complete	$P(FA_{pix})^{S_i}$	✗
	Sampled	$\sum_{i=N}^{R} \binom{R}{i} P(FA_{pix})^i (1 - P(FA_{pix}))^{R-i}$	✓
	Incomplete	$\sum_{i=M_i}^{S_i} \binom{S_i}{i} P(FA_{pix})^i (1 - P(FA_{pix}))^{S_i-i}$	✗
Miss: $P_i(M_{2D})$	Single	$P(M_{pix})$	✓
	Complete	$\sum_{i=1}^{S_i} \binom{S_i}{i} P(M_{pix})^i (1 - P(M_{pix}))^{S_i-i}$	✗
	Sampled	$\sum_{i=R-N+1}^{R} \binom{R}{i} P(M_{pix})^i (1 - P(M_{pix}))^{R-i}$	✓
	Incomplete	$\sum_{i=S_i-M_i+1}^{S_i} \binom{S_i}{i} P(M_{pix})^i (1 - P(M_{pix}))^{S_i-i}$	✗

7.3 Shape from Inconsistent Silhouette (SfIS)

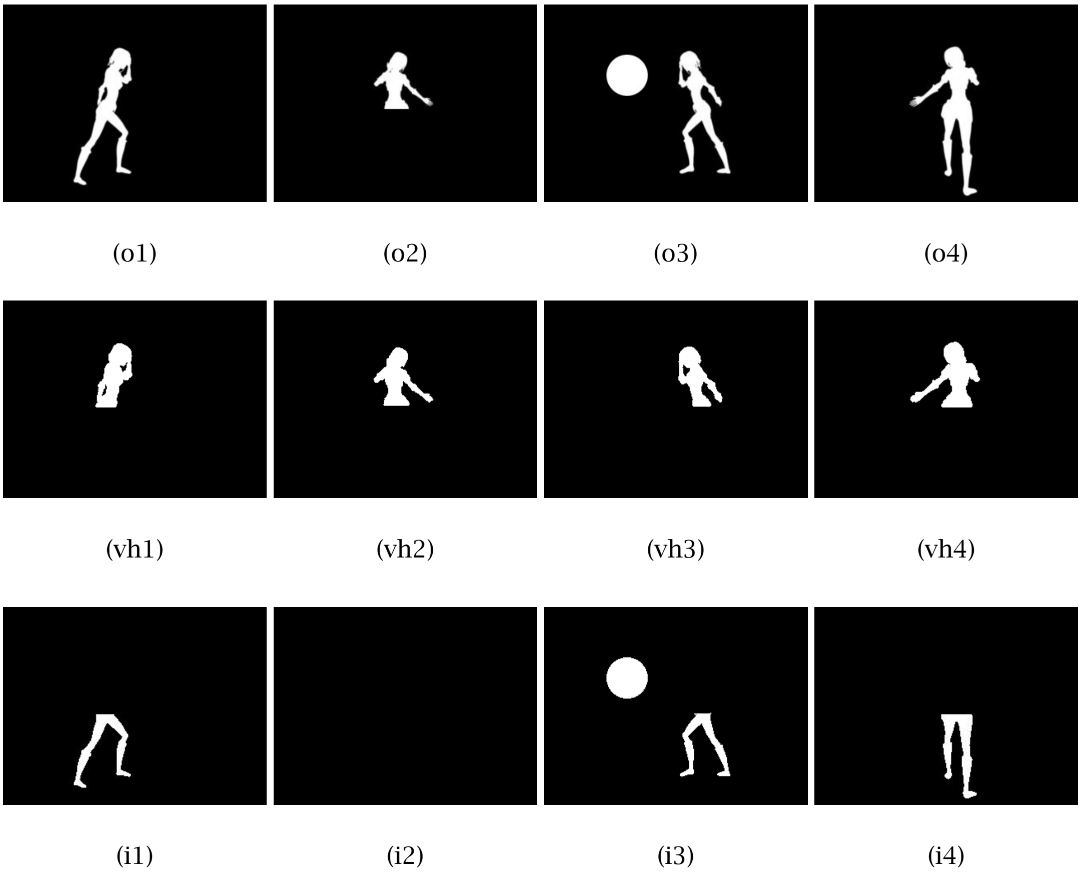
In SfIS, the VH is reconstructed using SfS methods and corrected later with those parts of the volume which were not correctly classified. 3D misclassifications can be detected by examining the inconsistent regions of the silhouettes. To detect inconsistent regions, one can project back the VH and test whether 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:ISs*). Preliminary work on SfIS for equiprobable projection tests was presented in [LPC06]. In the following, we provide the generalization of SfIS for any type of projection test. First, we formalize the concepts of IH and ISs and propose a procedure for estimating them. Then, we propose a method for optimally classifying the IH into Shape or Background.

7.3.1 Inconsistent Hull (IH)

The geometric concept of IH is introduced as the volume where there does not exist a shape which could possibly explain the observed silhouettes. The ISs are the resulting silhouettes after subtracting the original silhouettes with the projection of the visual hull (see Figure 7.9 for an example using the Kung-Fu Girl dataset).

The IH can be defined as the union of all the inconsistent cones, formed by the back-projection of the ISs into the 3D scene. Thus, when the set of silhouettes is consistent then *all* the ISs are empty, and the IH is also empty. However, when *a single* inconsistency appears in at least one silhouette then the IH will not be empty.

From the above equivalent definitions of the IH, it follows that the IH is disjoint from the VH ($VH \cap IH = \emptyset$). This can be observed in Figure 7.10, where different situations with consistent



(o1) (o2) (o3) (o4)

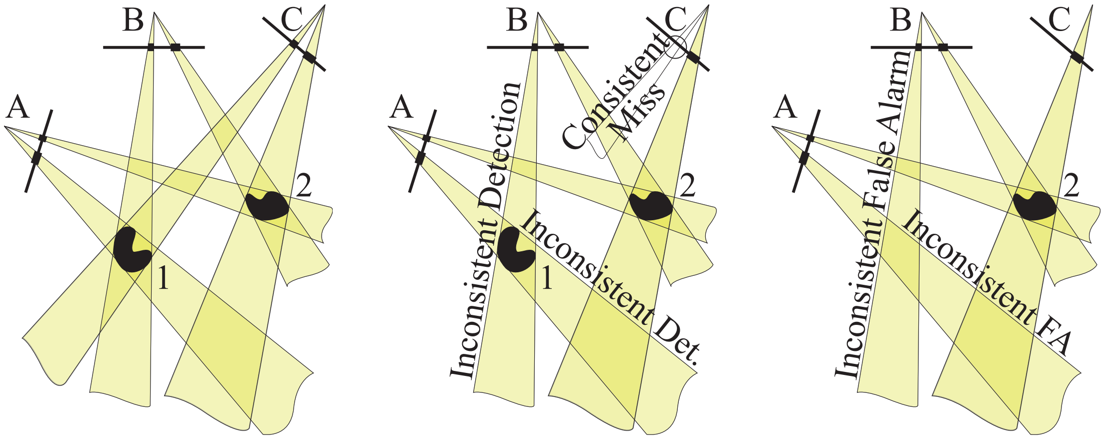
(vh1) (vh2) (vh3) (vh4)

(i1) (i2) (i3) (i4)

Figure 7.9: The first row of images shows four synthetic silhouettes, corresponding to the *Kung-Fu Girl* dataset, where some errors have been intentionally introduced: In (o2), the bottom part of the silhouette has been deliberately removed and, in (o3), a false alarm has been incorporated. The second row of images shows the projection of the VH reconstructed using SfS from the silhouettes above. Note that the 2D false alarm does not propagate to 3D, while a single miss propagates to 3D preventing a proper reconstruction of the VH. Finally, in the bottom row, the ISs are shown. The IH is the union of the back-projected cones of the inconsistent silhouettes.

and inconsistent sets of silhouettes have been depicted: In (a), there are two foreground objects which are correctly detected in all the cameras. In (b), camera C misses foreground object 1. The misdetection entails an inconsistent set of silhouettes in cameras A and B. However, projections of object 2 are consistent, and therefore the object can still be correctly detected inside the VH which will be reconstructed using any standard SfS algorithms. Further inspection of the figure indicates that the IH in this case corresponds to the union of the visual cones $camA\rightarrow obj1$ and $camB\rightarrow obj1$, confirming that is disjoint from the VH reconstructed around object 2. In (c), object 2 is correctly reconstructed but there have been two false alarms in cameras A and B. These false alarms coincide with the regions of the projection of object 1 in (b). The IH in this case is the same as in (b), again confirming that the IH is disjoint from the VH.

A closer look at Figure 7.10(b) & Figure 7.10(c) reveals some preliminary conclusions about how the IH could be classified. Observe that both figures depict different situations that could have been the cause of the same observed silhouettes. Note that it is impossible to guarantee whether there has been a single miss in camera C or two false alarms in cameras A and B. However, the figures suggest that the more inconsistent cones intersect, the higher the chances that the rest of cameras have missed an object in the area of inconsistent cone intersection. Of course, the exact chances of missing an object will also depend on the probabilities of 2D misclassification. The main problem to solve will be how to choose the minimum number of inconsistent intersections ($T^\star$) that have to be produced so that it can be determined that a part of the Shape was missed during the reconstruction process.



(a) (b) (c)

Figure 7.10: In (a), objects 1 and 2 are correctly detected in all the cameras. In (b) object 2 is correctly detected in all the cameras, but object 1 is missed in camera C. Former visual cones $camB\rightarrow obj1$ and $camA\rightarrow obj1$ are now inconsistent cones, whose union forms the IH. Note that it is impossible to know, from the observed silhouettes, whether there has been a single miss in camera C or two false alarms in cameras A and B, as depicted in (c).

There is yet another factor which will have to be considered in the choice of the optimal $T^\star$. As Figure 7.10 suggests, the number of intersecting inconsistencies is apparently tied to either the number of false alarms, or to the number of cameras minus the misses. However, there is a special case for which this is not true. This special situation is due to the fact that inconsistencies can be hidden by occluding objects. Figure 7.11 shows a typical situation with inconsistencies and occlusions. In the figure, a new object (object 3) has been deliberately placed in the same visual cone of $camB{\rightarrow}obj1$. Thus, object 3 prevents the inconsistent cone $camB{\rightarrow}obj1$ when camera C misses object 1. The figure clearly indicates that the number of inconsistent cone intersections is not a sufficient piece of information for deciding whether there have been misses in some silhouettes or not. Furthermore, the figure also suggests that all views where an object occludes a point of the IH will have to be ignored when determining its $T^\star$.

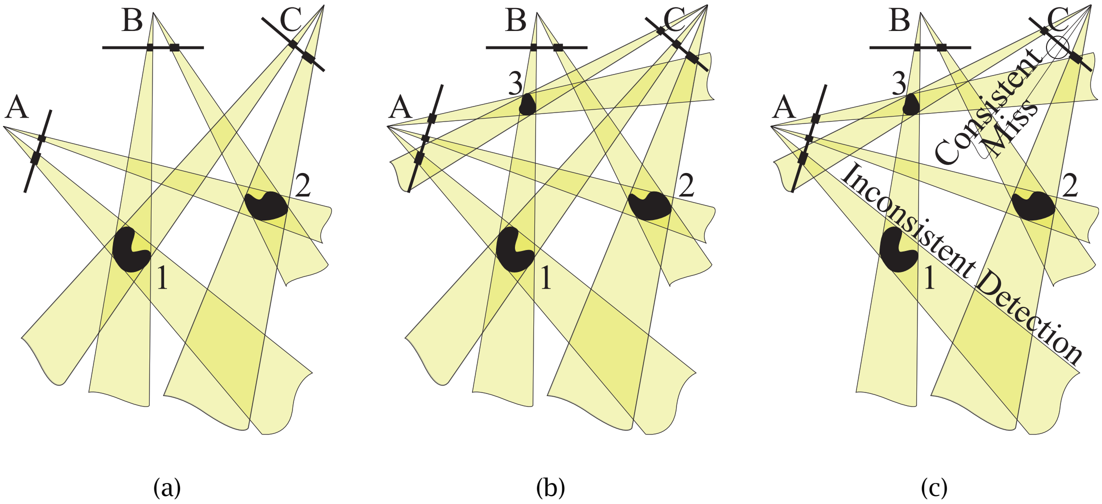


Figure 7.11: In (b) objects 1, 2 and 3 are correctly detected in cameras A, B and C. In (c), although objects 2 and 3 are correctly reconstructed, object 1 is not. The IH in this figure is smaller than its counterpart in Figure 7.10(b) due to the occlusion of object 1. The figure suggests that the number of occlusions will be an important determinant for proper classification of the IH.

In the following, we propose a method to determine the IH. Then, we describe how we choose the minimum number of inconsistent intersections that have to be produced so that it can be determined that an object was missed. The presented method will take into account previous considerations regarding occlusions.

7.3.2 IH Voxelization

Prior to deriving the expressions for the IH classification, first we need a method to reconstruct it.

In order to estimate the IH, we need to determine the *union* of the inconsistent cones (corresponding to the back-projection of the ISs) analogously as SfS methods determine the *intersection* of the visual cones (corresponding to the back-projections of the silhouettes). As it has been previously reviewed in section 7.1.1, determining visual cones intersecting can be performed in different ways. For instance, some SfS techniques project back the silhouettes, creating the set of visual cones which are intersected in the 3D space. In other approaches, the volume is divided in voxels which are then projected to the images to find out (using a projection test) whether they are contained in every silhouette or not.

In this paper, we develop the concept of Shape from Inconsistent Silhouette using a voxel-based approach, although similar considerations can be derived with the geometrical approach.

Algorithm 4 Voxelization of the IH

```
Require: Silhouettes: $S(c)$, Proj. Test: $PT_c(voxel, Silhouette)$
for all $voxel$ do
    $VH(voxel) \leftarrow$ true
    for all $c$ do
        if $PT_c(voxel, S(c))$ is false then
            $VH(voxel) \leftarrow$ false
        end if
    end for
end for
Project the $VH$ to all the camera views: $VH_{proj}(c)$
for all $voxel$ do
    $IH(voxel) \leftarrow$ false
    for all $c$ do
        if $PT_c(voxel, S(c))$ is true then
            if $PT_c(voxel, S(c)) \neq PT_c(voxel, VH_{proj}(c))$ then
                $IH(voxel) \leftarrow$ true
            end if
        end if
    end for
end for
```

The detailed process for the IH voxelization is shown in Algorithm 4. Note that in the voxel-based approach, the role of the inconsistent silhouettes (difference between *silhouettes* and *VH projection*) is replaced by the nonequivalence of their projection tests: $PT_c(voxel, S(c)) \neq PT_c(voxel, VH_{proj}(c))$.

7.3.3 Unbiased Hull (UH)

The IH contains all the volumetric points which cannot justify 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. We define the Unbiased Hull (UH) as the subset of the IH which is better explained as Shape for minimizing the probability of voxel misclassification. We call it *unbiased*, since the volumetric points of the IH are classified as either Shape or Background based on the lowest probability of error, while VH reconstruction methods are *biased* for always classifying the IH as Background. In the final stage of the process, the union of the VH and UH will form the best apparent hull in terms of lower misclassification probability.

The classification of the voxels in the IH has to be optimal based on all the characteristics we can gather from them:

- Firstly, each voxel in the IH has an associated number of foreground projections ($\mathcal{F}$), which corresponds to the number of visual cone intersections in the voxel. $\mathcal{F}$ can be simply calculated by counting the number of silhouettes ($S(c)$) where the projection test is passed: $PT_c(voxel, S(c)) = true$, being $voxel \in$ IH. For example, in Figure 7.11(c), all voxels corresponding to object 1 have $\mathcal{F} = 2$, for being in the visual cones $camA{\rightarrow}obj1$ and $camB{\rightarrow}obj3$.

  In the IH, the number of foreground projections is bounded by: $1 \leq \mathcal{F} \leq C - 1$, since 0 foreground projections would correspond to a consistent background detection in the VH and C detections would correspond to a consistent foreground detection in the VH.

- A voxel in the IH can also be characterized by the number of consistent foreground projections ($\mathcal{O}$), corresponding to the number of views where the voxel has been occluded. $\mathcal{O}$ can be computed as the number of times that the projection test is passed both in a silhouette ($S(c)$) and in the projection of the VH ($VH_{proj}(c)$): $PT_c(voxel, S(c)) = true = PT_c(voxel, VH_{proj}(c))$, being $voxel \in$ IH. For instance, voxels corresponding to object 1 in Figure 7.11(c) have $\mathcal{O} = 1$, for intersecting with the consistent occluding cone: $camB{\rightarrow}obj3$.

  The number of occlusions in the IH is bounded by $0 \leq \mathcal{O} \leq C - 1$, as C occlusions would correspond to a foreground detection in the VH.

- A voxel in the IH also has an associated number of inconsistencies ($\mathcal{I}$), which corresponds to the number of inconsistent foreground projections. Note that $\mathcal{I}$ is such that $\mathcal{F} = \mathcal{I} + \mathcal{O}$. From a practical point of view, the $\mathcal{I}$ of each voxel corresponds to the number of times that the projection test is passed in a silhouette ($S(c)$) but not passed in the projection of the VH ($VH_{proj}(c)$): $PT_c(voxel, S(c)) = true \neq PT_c(voxel, VH_{proj}(c))$, being $voxel \in$ IH. For instance, voxels corresponding to object 1 in Figure 7.11(c), have $\mathcal{I} = 1$, for being in the inconsistent cone $camA{\rightarrow}obj1$.

In the IH, the number of inconsistencies ($\mathcal{I}$) is bounded by:

$$1 \le \mathcal{I} \le C - \mathcal{O} - 1, \tag{7.15}$$

where the lower bound is due to the fact that all the voxels of the IH have been intersected with at least one inconsistent cone; and where the upper bound has to be lower or equal to $C-1$, since C inconsistencies would correspond to a foreground detection of the Visual Hull. Moreover, the number of occlusions ($\mathcal{O}$) also has to be subtracted ($C - \mathcal{O} - 1$), since occlusions are only produced when voxels are intersected with consistent visual cones.

- Finally, a voxel can also be associated with the number of views where it projects to background ($\mathcal{B}$). Note that $\mathcal{B} = C - \mathcal{F}$, and therefore $\mathcal{B} = C - \mathcal{I} - \mathcal{O}$. The number of background projections ($\mathcal{B}$) can be computed by counting the number of silhouettes ($S(c)$) where $PT_c(voxel, S(c)) = false$, being $voxel \in$ IH. For instance, in Figure 7.11(c), voxels corresponding to object 1 have $\mathcal{B} = 1$.

  The bounds on the number of background detections ($\mathcal{B}$) are: $1 \le \mathcal{B} \le C - \mathcal{O} - 1$. Besides using a similar reasoning as with the number of inconsistencies ($\mathcal{I}$), the expression can also be simply deduced by using inequality (7.15):

$$\begin{aligned}
&1 \le \mathcal{I} \le C - \mathcal{O} - 1 \Rightarrow \overbrace{1 \le \rlap{$\underbrace{\phantom{C - \mathcal{B} - \mathcal{O} \le C - \mathcal{O} - 1}}_{b}$}C - \mathcal{B} - \mathcal{O}}^{a} \le C - \mathcal{O} - 1 \\
&a : 1 \le C - \mathcal{B} - \mathcal{O} \Rightarrow \mathcal{B} \le C - \mathcal{O} - 1 \\
&b : C - \mathcal{B} - \mathcal{O} \le C - \mathcal{O} - 1 \Rightarrow \mathcal{B} \ge 1
\end{aligned} \tag{7.16}$$

Some further considerations regarding $\mathcal{F}$, $\mathcal{I}$, $\mathcal{O}$ and $\mathcal{B}$ can be derived: Interestingly, the number of inconsistent projections ($\mathcal{I}$) in a voxel are due to either having had false alarms in $\mathcal{I}$ silhouettes or due to having had misses in $\mathcal{B}$ silhouettes, where $\mathcal{B} = C - \mathcal{I} - \mathcal{O}$.

As $\mathcal{I}$ rises, $\mathcal{B}$ falls, and therefore the probability of having $\mathcal{B}$ simultaneous misses is increased while the probability of having $\mathcal{I}$ simultaneous false alarms is decreased (see Figure 7.3(a) & Figure 7.3(c), respectively). Based on this reasoning, optimal threshold $T^\star$ has to be such that if $\mathcal{I} \ge T^\star$, the voxel is better explained as Shape (with $C - \mathcal{I} - \mathcal{O}$ misses) than Background (with $\mathcal{I}$ false alarms):

$$\begin{array}{lcl}
\mathcal{I} \ge T^\star & \Rightarrow & \text{decide } \textit{Shape} \\
\mathcal{I} < T^\star & \Rightarrow & \text{decide } \textit{Background}
\end{array} \tag{7.17}$$

In order to find $T^\star$, first we have to express which is the probability of voxel misclassification for any $P(Err_{3D}[T])$ so that $T^\star$ is that one which minimizes it:

$$T^\star = \underset{T}{\operatorname{argmin}}\, P(Err_{3D}[T]) \tag{7.18}$$

Similarly as with the voxels in the VH, a voxel in the IH is misclassified if it is wrongly classified as Shape (false alarm) or if it is wrongly classified as Background (miss), as expressed in (7.1).

We first examine the probability of false alarm ($P(FA_{3D})$). A false alarm in a voxel happens when a voxel is classified as part of the Shape, while in fact it forms part of the Background. If the voxel forms part of the Background, then all inconsistencies correspond to false alarms of the projection test. Since shape classification occurs when $\mathcal{I} \geq T$:

$$P(FA_{3D}) = \sum_{i=max(T,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(FA_{2D})^{i}(1-P(FA_{2D}))^{C-i}, \tag{7.19}$$

corresponding to the summation of all possible combinations that trigger a false alarm in a voxel, and assuming equiprobable $P_i(FA_{2D}) = P(FA_{2D})$ in all views (i).

Note that the combinations are bounded by the upper ($C - \mathcal{O} - 1$) and lower (1) bounds on the number of possible inconsistencies (see inequality 7.15), confirming previous considerations regarding the influence of occlusions. Also note that the expression is correctly defined independently of the chosen T, even if the chosen value is out of the interval where the number of inconsistencies are possible.

The expression for $P(FA_{3D})$ when the assumption of equiprobable $P_i(FA_{2D})$ does not hold is:

$$P(FA_{3D}) = \sum_{i=max(T,1)}^{C-\mathcal{O}-1} \left(\overbrace{\sum_{n_1=1}^{C-i+1} \sum_{n_2>n_1}^{C-i+2} \cdots \sum_{n_i>n_{i-1}}^{C}}^{\substack{\text{Equivalent to } \binom{C}{i} \text{ for} \\ \text{non-equiprobable } P_i(FA_{2D})}} \left(\overbrace{P_{n_1}(FA_{2D}) \cdots P_{n_i}(FA_{2D})}^{i \text{ products}} \overbrace{\prod_{\substack{m=1 \\ m \neq \{n_1,\cdots,n_i\}}}^{C} (1 - P_m(FA_{2D}))}^{C-i \text{ products}} \right) \right),$$

which is the expression which has to be used when using the Complete and Incomplete Pixels Projection Test, for instance.

The opposite misclassification case in the IH is having a miss in a voxel. This is the case when a voxel is classified as part of the Background, while in fact it forms part of the Shape. Since a voxel is wrongly classified as Background if $\mathcal{I} < T$:

$$\mathcal{I} < T \quad \underset{\left\{\begin{array}{c} C = \mathcal{F} + \mathcal{B} \\ \mathcal{F} = \mathcal{I} + \mathcal{O} \end{array}\right\}}{\iff} \quad \mathcal{B} \geq C - \mathcal{O} - T + 1 \tag{7.20}$$

Then, the probability of miss $P(M_{3D})$ in the IH can be expressed in a similar manner as with false alarms:

$$P(M_{3D}) = \sum_{i=max(C-\mathcal{O}-T+1,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(M_{2D})^{i}(1-P(M_{2D}))^{C-i}, \tag{7.21}$$

where $P(M_{2D})$ corresponds to the probability that the projection test has not been passed by error, and assuming equiprobable $P_i(M_{2D}) = P(M_{2D})$ in all views (i).

If we cannot assume that the probabilities of miss are equiprobable in all the cameras, then:

$$P(M_{3D}) = \sum_{i=max(C-\mathcal{O}-T+1,1)}^{C-\mathcal{O}-1} \left(\overbrace{\sum_{n_1=1}^{C-i+1} \sum_{n_2>n_1}^{C-i+2} \cdots \sum_{n_i>n_{i-1}}^{C}}^{\substack{\text{Equivalent to } \binom{C}{i} \text{ for} \\ \text{non-equiprobable } P_i(M_{2D})}} \left(\overbrace{P_{n_1}(M_{2D}) \cdots P_{n_i}(M_{2D})}^{i \text{ products}} \overbrace{\prod_{\substack{m=1 \\ m \neq \{n_1,\cdots,n_i\}}}^{C} (1 - P_m(M_{2D}))}^{C-i \text{ products}} \right) \right)$$

Once the probability of voxel misclassification has been expressed, $T^\star$ can be easily obtained by doing an exhaustive search of the minimum $P(Err_{3D})$ over all possible T for each case of occlusion as shown in Algorithm 5.

Algorithm 5 Optimal thresholds for all cases of occlusion: $T^\star[o]$. Note that $T^\star[o]$ will take different values for each voxel depending upon whether $P(M_{2D})$ or $P(FA_{2D})$ also depend on the voxel.

```
for all Cases of Occlusion: $o = 0 \cdots C - 1$ do
    $MinPerr \leftarrow 1$
    for all Possible Number of Inconsistencies: $i = 1 \cdots C - o - 1$ do
        if $P(Err_{3D}[T = i, \mathcal{O} = o)]) \leq MinPerr)$ then
            $T^\star[o] \leftarrow i$
            $MinPerr \leftarrow P(Err_{3D}[T = i, \mathcal{O} = o)])$
        end if
    end for
end for
```

Once $T^\star$ has been obtained, SfIS can be implemented as shown in Algorithm 6.

7.4 Real-Time SfIS

SfIS can be very fast, once the optimal thresholds have been computed for each possible case of occlusion and stored in a lookup table (LUT). Real-time operation of SfIS can be achieved when using it in combination with fast projection tests. Often, the One Pixel Projection Test is used for being fast and simple. However, LUTs cannot be used when probabilities of 2D miss and false alarm of the projection test change over time ($P(FA_{Pix}(t)$ and $P(M_{Pix}(t))$. For example, when a mixture of Gaussians is used to model the Background, the probabilities of miss and false alarm of the pixels depend on the variances of the Gaussians, which are constantly changing over time.

Algorithm 6 SfIS algorithm

```
Require: Silhouettes: $S(c)$, $T^{\star}[o]$, VH, a Proj. Test Function: $PT_c(voxel, Silhouette)$
Project the $VH$ to all the camera views: $VH_{proj}(c)$
for all $voxel$ do
    $i \leftarrow 0$
    $o \leftarrow 0$
    for all $c$ do
        if $PT_c(voxel, S(c))$ is true then
            if $PT_c(voxel, VH_{proj}(c))$ is false then
                $i \leftarrow i + 1$
            else
                $o \leftarrow o + 1$
            end if
        end if
    end for
    if $i > 0$ then
        if $i \geq T^{\star}[o]$ then
            $UH(voxel) \leftarrow$ 3D Foreground
        else
            $UH(voxel) \leftarrow$ 3D Background
        end if
    end if
end for
```

Under these circumstances, it is important to have a fast search strategy that can compute the optimal thresholds online.

7.4.1 A Fast Threshold Search Approach

The method presented here is focused on a fast implementation of SfIS using the One Pixel Projection Test. However, we develop the equations for the more general case of any projection test which is equiprobable with respect to all views (refer to table 7.1).

To begin with, since $P(Err_{3D}[T])$ is not continuous, it cannot be minimized by differentiating it. However, fast search of $T^\star$ can be achieved if the problem can be constrained into finding the minimum of $P(Err_{3D}[T])$ in a strictly convex interval L. In other words, if we can guarantee that $P(Err_{3D}[T])$ is strictly convex in L under certain conditions, and provided that these conditions are reasonable, then there will always exist a global minimum in L, which will be fast to obtain.

In the following, we propose some sufficient conditions which guarantee the strict convexity of $P(Err_{3D}[T])$, with respect to $T \in \mathbb{Z}$ in the interval of interest L: first, we find which is the interval, and then, we obtain the conditions.

In order to find the interval L of interest, it is important to remember that the range of possible inconsistencies in a voxel in the IH is $\mathcal{I} \in [1, C - \mathcal{O}]$. This is the reason why, in the IH, $P(Err_{3D}[T])$ has constant values for $T \leq 1$ and $T \geq C - \mathcal{O}$, corresponding to the probabilities of always deciding Shape or always deciding Background, respectively. Since strict convexity of a function can only occur in the interval where the function is not constant, the interval of convexity of $P(Err_{3D}[T])$ has to be: $L \in]1, C - \mathcal{O}[$.

Once that L has been determined, we only have to seek the conditions that make $P(Err_{3D}[T])$ strictly convex in the interval.

In general, a function $f[x]$ in $\mathbb{Z}$ is strictly convex [BV04, Mur99] if it can be expressed as in inequality (7.22) (see Figure 7.12).

$$f[x-1] + f[x+1] > 2f[x] \tag{7.22}$$

And since conditions of strict convexity have to be found assuming that the projection tests are equiprobable in all camera views, the working expression of $P(Err_{3D}[T])$ is:

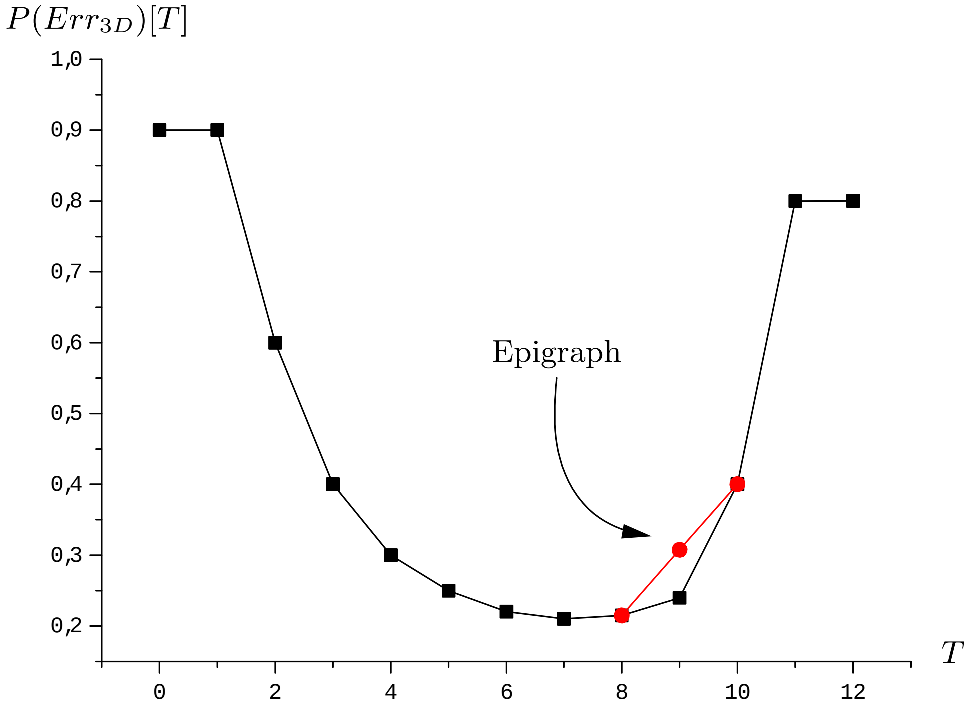


Figure 7.12: A function is strictly convex if and only if for every two points p, q, a point in the middle lies below the segment (epigraph) pq.

$$\begin{aligned} P(Err_{3D}[T]) = \quad & P_S \underbrace{\sum_{i=max(C-\mathcal{O}-T+1,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(M_{2D})^i (1-P(M_{2D}))^{C-i} +}_{P(M_{3D})} \\ & (1-P_S) \underbrace{\sum_{i=max(T,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(FA_{2D})^i (1-P(FA_{2D}))^{C-i}}_{P(FA_{3D})} \end{aligned} \tag{7.23}$$

Then, strict convexity of $P(Err_{3D}[T])$ occurs if (7.23) satisfies (7.22):

$$\begin{aligned} & P(Err_{3D}[T-1]) + P(Err_{3D}[T+1]) = \\ & \quad \underbrace{P_S \sum_{i=max(C-\mathcal{O}-T+2,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(M_{2D})^i (1-P(M_{2D}))^{C-i} +}_{M_a} \\ & \quad (1-P_S) \underbrace{\sum_{i=max(T-1,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(FA_{2D})^i (1-P(FA_{2D}))^{C-i} +}_{FA_a} \\ & \quad \underbrace{P_S \sum_{i=max(C-\mathcal{O}-T,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(M_{2D})^i (1-P(M_{2D}))^{C-i} +}_{M_b} \\ & \quad (1-P_S) \underbrace{\sum_{i=max(T+1,1)}^{C-\mathcal{O}-1} \binom{C}{i} P(FA_{2D})^i (1-P(FA_{2D}))^{C-i}}_{FA_b} \\ & \qquad > 2P(Err_{3D}[T]) = 2P_S P(M_{3D}) + 2(1-P_S)P(FA_{3D}) \end{aligned} \tag{7.24}$$

Let us rewrite the terms M_a, FA_a, M_b and FA_b, into:

$$\begin{aligned} M_a: \quad & P(M_{3D}) - \underbrace{\binom{C}{C-\mathcal{O}-T+1} P(M_{2D})^{C-\mathcal{O}-T+1} (1-P(M_{2D}))^{\mathcal{O}+T-1}}_{M'_a} H(T-2)H(C-\mathcal{O}-T) \\ FA_a: \quad & P(FA_{3D}) + \underbrace{\binom{C}{C-\mathcal{O}-T} P(M_{2D})^{C-\mathcal{O}-T} (1-P(M_{2D}))^{\mathcal{O}+T}}_{FA'_a} H(T-1)H(C-\mathcal{O}-1-T) \\ M_b: \quad & P(M_{3D}) + \underbrace{\binom{C}{T-1} P(FA_{2D})^{T-1} (1-P(FA_{2D}))^{C-T+1}}_{M'_b} H(T-2)H(C-\mathcal{O}-T) \\ FA_b: \quad & P(FA_{3D}) - \underbrace{\binom{C}{T} P(FA_{2D})^{T} (1-P(FA_{2D}))^{C-T}}_{FA'_b} H(T-1)H(C-\mathcal{O}-1-T), \end{aligned}$$

where $H(T)$ is the Heaviside step function:

$$H(T) = \begin{cases} 0 & T < 0 \\ 1 & T \geq 0 \end{cases} \tag{7.25}$$

As we are only interested in the interval $L \in]1, C - \mathcal{O}[$, inequality (7.24) can be expressed as:

$$\underbrace{P_S(M'_b - M'_a)}_{T_1} + \underbrace{(1 - P_S)(FA'_a - FA'_b)}_{T_2} > 0, \tag{7.26}$$

where no terms are affected by the Heaviside step functions in L, and where $P(M_{3D})$ and $P(FA_{3D})$ are canceled.

As a matter of fact, since we only need to find a *sufficient* condition of strict convexity of $P(Err_{3D}[T])$, we can separate the left term of the inequality into two terms (T_1, T_2), and seek the conditions that make both terms larger than 0. Forcing the first term ($T_1 = P_S(M'_b - M'_a)$) to be greater than 0, can be expressed as follows:

$$\frac{\binom{C}{C-\mathcal{O}-T} P(M_{2D})^{C-\mathcal{O}-T}(1 - P(M_{2D}))^{\mathcal{O}+T}}{\binom{C}{C-\mathcal{O}-T+1} P(M_{2D})^{C-\mathcal{O}-T+1}(1 - P(M_{2D}))^{\mathcal{O}+T-1}} > 1, \tag{7.27}$$

that, after isolating T, is equivalent to:

$$T < \frac{(C - \mathcal{O} + 1)P(M_{2D})^{-1}(1 - P(M_{2D})) - \mathcal{O}}{1 + P(M_{2D})^{-1}(1 - P(M_{2D}))} \tag{7.28}$$

In order to guarantee that inequality (7.28) is satisfied in L, we can impose a stricter condition on T, by replacing it with $C - \mathcal{O}$ which is larger than the largest possible value that T can take in the interval L:

$$C - \mathcal{O} < \frac{(C - \mathcal{O} + 1)P(M_{2D})^{-1}(1 - P(M_{2D})) - \mathcal{O}}{1 + P(M_{2D})^{-1}(1 - P(M_{2D}))} \Leftrightarrow C < \frac{1}{P(M_{2D})} - 1, \tag{7.29}$$

which is stricter than condition (7.28).

Note that if condition (7.29) is satisfied, then condition (7.28) is also satisfied, and therefore T_1 is greater than 0 in L.

We can do a similar reasoning with $T_2 = (1 - P_S)(FA'_a - FA'_b)$, which is larger than 0 if:

$$\frac{\binom{C}{T-1} P(FA_{2D})^{T-1}(1 - P(FA_{2D}))^{C-T+1}}{\binom{C}{T} P(FA_{2D})^{T}(1 - P(FA_{2D}))^{T}} > 1, \tag{7.30}$$

that, after isolating T:

$$T > \frac{(C + 1)}{1 + P(FA_{2D})^{-1}(1 - P(FA_{2D}))} \tag{7.31}$$

This time, we replace T with 1, which is smaller than the smallest possible value that T can take in L:

$$1 > \frac{(C+1)}{1 + P(FA_{2D})^{-1}(1 - P(FA_{2D}))} \Leftrightarrow C < \frac{1}{P(FA_{2D})} - 1, \tag{7.32}$$

which is a stricter condition than the one expressed in inequality (7.31).

Similarly as with T_1, if (7.32) is satisfied, then condition (7.31) is also satisfied, and therefore T_2 is greater than 0.

Finally, $P(Err_{3D}[T])$ can be said to be strictly convex in $L \in]1, C - \mathcal{O}[$ if conditions (7.29) and (7.32) hold together, which can be expressed in a single inequality as follows:

$$C < \frac{1}{max(P(FA_{2D}), P(M_{2D}))} - 1, \tag{7.33}$$

which is usually satisfied in all typical scenarios. For instance, $P(Err_{3D}[T])$ is strictly convex with respect to T, when less than 9 cameras are used even if misclassification probabilities are high ($P(FA_{2D}) = P(M_{2D}) = 0.1$).

Note that if condition (7.33) is not satisfied, then we cannot guarantee whether $P(Err_{3D}[T])$ is convex or not, and Algorithm 5 has to be used. However, when the condition holds, which is often the case, we can make use of some of the properties of strictly convex functions:

If $P(Err_{3D}[T])$ can be said to be strictly convex, then its central difference $\delta P(Err_{3D}[T])$ (see Figure 7.13) will correspond to a sequence of sorted elements in increasing order. And the element in the sorted sequence which is closest to zero, will correspond to the minimum of $P(Err_{3D}[T])$:

$$\delta P(Err_{3D}[T]) = P(Err_{3D}[T+1] - P(Err_{3D}[T-1]) \tag{7.34}$$

where δ is the central difference operator: $\delta f[x] = f[x+1] - f[x-1]$.

Observe that if the sequence $\delta P(Err_{3D}[T])$ is sorted, we can approach the closest value of 0 from the left side, by checking whether the midpoint of the sequence is larger than 0, eliminating half of the sequence from further consideration (see Figure 7.14). The binary search [CLRS01] is an algorithm that repeats this procedure, halving the size of the remaining portion of the sequence each time.

The complexity of the search operation in the binary search is $O\log_2(n)$, because at each test one half of the search space is discarded. Furthermore, $\delta P(Err_{3D}[T])$ can be computed $\frac{4}{C-\mathcal{O}-1}$ times faster than $P(Err_{3D}[T])$, since the sum over all possible cases of 3D false alarm and miss does not have to be computed:

$$\delta P(Err_{3D}[T]) = P_S(M'_b - M'_a) + (1 - P_S)(FA'_a - FA'_b), \tag{7.35}$$

where there are no terms of the type: $\sum^{C-\mathcal{O}-1}$.

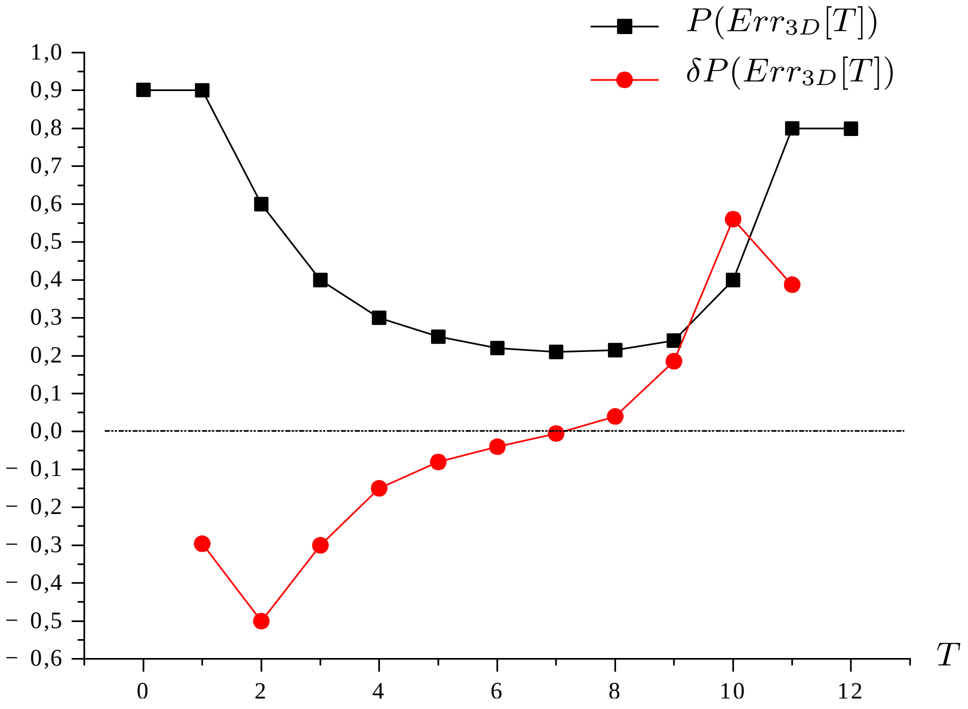


Figure 7.13: $P(Err_{3D}[T])$ and $\delta P(Err_{3D}[T])$ showing that $\delta P(Err_{3D}[T])$ is a strictly increasing function in the interval of strict convexity of $P(Err_{3D}[T])$.

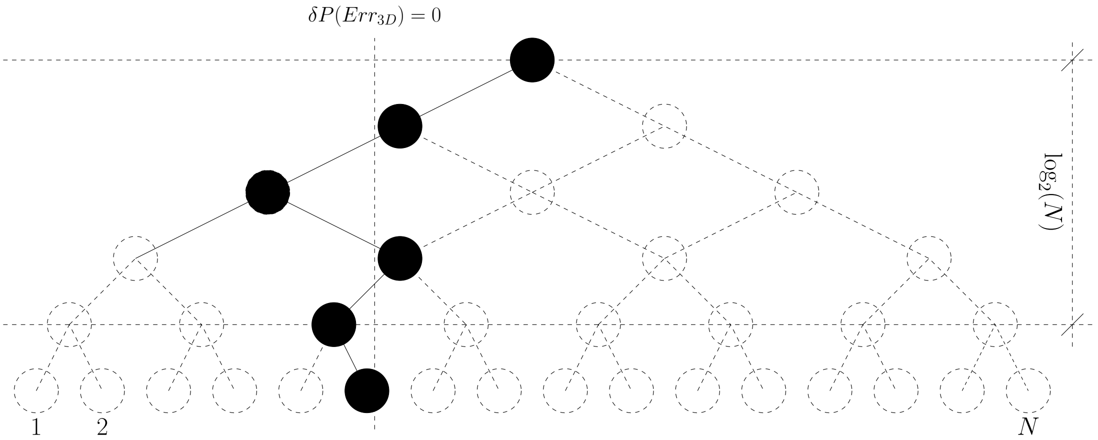


Figure 7.14: Binary Tree

In conclusion, the method described achieves the optimal solution $T^{\star}$ in $O\log_2(\frac{4n}{C-\mathcal{O}-1})$ time, which is faster than the linear search approach described in Algorithm 5, which only achieves the solution in $O(n)$ time. It is noteworthy that the method proposed improves drastically as the size of the array, i.e., the number of cameras, is increased.

Finally, refer to Algorithm 7 for the detailed implementation of the method, considering a left approach to the optimal solution $T^{\star}$. Note that the optimal solution has to be found for every possible case of occlusion that may occur in the scene.

Algorithm 7 Binary search of $T^{\star}[o]$ for all cases of occlusion (o).

```
Require: $\delta P(Err_{3D}[T,o])$
for all Cases of Occlusion: $o = 0 \cdots C-1$ do
    $left \leftarrow 1$
    $right \leftarrow C - \mathcal{O}$
    while $left \leq right$ do
        $index \leftarrow \left\lceil \frac{left+right}{2} \right\rceil$
        if $\delta P(Err_{3D}[T = index, \mathcal{O} = o]) > 0$ then
            $right \leftarrow index - 1$
        else
            $left \leftarrow index + 1$
        end if
    end while
    return $T^{\star}[o] = \text{argmin}_{T=\{index,index+1\}} \delta P(Err_{3D}[T, \mathcal{O} = o])$
end for
```

7.5 A Unified Cooperative-SfIS Bayesian Framework

Previous chapters described a set of probabilistic methods for obtaining 2D silhouettes and a cooperative framework that allowed obtaining 3D classifications using 2D probabilities. In addition, the bases for 2D model update using probabilistic 3D information were also established. In this last chapter, we have presented a new tool that allows to reclassify an initial volumetric estimate making use of the geometrical constraints of the problem. In this section we present a complete system which incorporates the geometrical constraints into the integrated 2D-3D Bayesian framework we described in the previous chapter. The schematic diagram is shown in Figure 7.15.

In fact, both Bayesian and geometrical approaches can cooperate to the benefit of the system. After a close examination of both approaches, the integration comes as the natural outcome of choosing the aspects where both approaches excel. The details on this system integration are presented in the following.

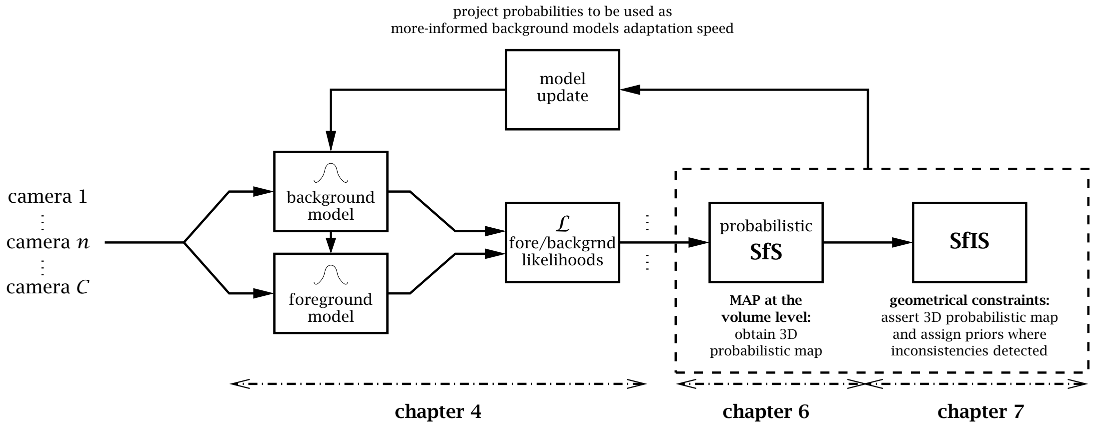


Figure 7.15: The complete schematic diagram of the unified framework for consistent 2D/3D foreground object detection system.

To obtain a volumetric estimate using geometrical constraints in our 2D-3D Bayesian framework, first, a set of probabilistic pixel models are created for each image in the set-up. Pixel models can be used for Bayesian classification of 2D silhouettes, which can be later employed for obtaining 3D reconstructions. In addition, it is possible to maintain those pixel models with new observations so that their posterior probabilities are always maximum. The details on this approach were described in chapter 6.

This 2D probabilistic information is incorporated to the SfIS approach as follows. Once the pixel models have been estimated, an initial version of the SfS is obtained using the SfS cooperative approach described in chapter4. The cooperative approach makes use of the two-dimensional probabilistic methods previously mentioned to obtain foreground and background probabilities for each voxel. These probabilities are then used to classify all the voxels, as it was described in detail in the mentioned chapter. Contrarily to the classical SfS approach, in the cooperative approach, 2D probabilistic values are transferred to 3D and used there to classify the volume.

Once a volumetric Shape has been classified, then it is projected back to each one of the views and compared there with a set of silhouettes that are temporarily classified using only the pixel models. The inconsistencies between Shape and silhouettes are determined and then the SfIS algorithm is applied as usual so that a refined 3D model is obtained.

The last step corresponds to the process in which the images' pixels models are updated. Note that in Bayesian 2D foreground segmentation, the models' adaptation speed varies according to the probabilities of each class, as shown in chapter 4. In the cooperative approach, the voxel probabilities are projected and used as the adaptation speed of the pixels models. However, at this point, SfIS provides an extra source of information which can be incorporated to the probabilistic voxel representation obtained with the cooperative SfS approach. Those voxels which SfIS reclassifies are therefore reassigned with prior fixed probabilistic values (0.9^C for a 3D miss and 0.1^C for a 3D false detection[1]).

Finally, the 2D pixel models are update using (4.66), after projecting the reassigned 3D probabilities with the projection rules defined in §4.4.2, on page 62.

Incorporating SfIS to the cooperative Bayesian framework clearly improves the overall system accuracy. Indeed, it is important not to update models with wrong observation values due to erroneous classifications. SfIS helps in detecting some of these errors in the silhouettes making use of the existing redundancy in a multi-camera setup.

It is relevant to observe that this integration of probabilistic and geometrical compiles all the different aspects that this thesis has addressed. First, the 2D Bayesian approach is used, then the 3D cooperative background learning is employed to obtain a preliminary set of 3D probabilistic values. This initial volumetric probabilistic representation is refined using SfIS and, finally, the 2D models are updated using our unified Bayesian framework.

7.6 Results

In order to fully evaluate SfIS, two types of results are presented. First, we present the theoretical improvements of SfIS over SfS in the IH. Second, we show some $VH \cup IH$ reconstructions and projections using synthetic and real data.

7.6.1 Theoretical Improvements

It is important to keep in mind that SfIS is focused on minimizing the probability of Shape misclassification in the IH in terms of *consistency*. This is the reason why all points which belong to consistent zones are considered to be error-free.

So, in order to compare the errors of SfS and SfIS as fairly as possible, let us first rewrite the expressions of error of SfS in the IH, assuming that there cannot be consistent misclassifications. The new error rate is lower than the one presented in section 7.2, which considered that misclassifications could also be consistent. However the reformulation is necessary in order to not unfairly worsen the results of SfS in front of SfIS:

$$\begin{aligned} P_{SfS}(Err_{3D}) &= P_B P_{SfS}(FA_{3D}) + P_S P_{SfS}(M_{3D}) \\ P_{SfS}(FA_{3D}) &= 0 \\ P_{SfS}(M_{3D}) &= \sum_{i=1}^{C-\mathcal{O}-1} \binom{C}{i} P(M_{2D})^i (1 - P(M_{2D}))^{C-i}, \end{aligned} \tag{7.36}$$

[1] The proposed probabilities are raised to the power of C to adapt them to the change of dimensionality (see §6.5, on page 6.5, for the details).

where $P(M_{2D})$ corresponds to the probability that the projection test has not been passed by error, and assuming equiprobable $P_i(M_{2D}) = P(M_{2D})$ in all views (i). Note that the upper bound is $C - \mathcal{O} - 1$, corresponding to the maximum number of background detections possible in the IH.

If we cannot assume that the probabilities of miss are equiprobable in all the cameras, then:

$$P_{SfS}(M_{3D}) = \sum_{i=1}^{C-\mathcal{O}-1} \left(\overbrace{\sum_{n_1=1}^{C-i+1} \sum_{n_2>n_1}^{C-i+2} \cdots \sum_{n_i>n_{i-1}}^{C}}^{\substack{\text{Equivalent to } \binom{C}{i} \text{ for} \\ \text{non-equiprobable } P_i(M_{2D})}} \left(\overbrace{P_{n_1}(M_{2D}) \cdots P_{n_i}(M_{2D})}^{i \text{ products}} \overbrace{\prod_{\substack{m=1 \\ m \neq \{n_1, \cdots, n_i\}}}^{C} (1 - P_m(M_{2D}))}^{C-i \text{ products}} \right) \right)$$

Figure 7.16(a) shows the $\frac{P_{SfIS}(Err_{3D})}{P_{SfS}(Err_{3D})}$ probability ratio, assuming that an equiprobable projection test is being used and that there have not been occlusions. Note that the ratio is always below 1, meaning that the probability of voxel misclassification in SfIS is always lower than in SfS.

An aspect of interest of SfIS is that it behaves as traditional SfS when (1) $P(FA_{2D})$ is high or (2) $P(M_{2D})$ is low: In the first case, when there are high chances of 2D false alarm, SfIS mimics SfS in order not to incorporate 3D false alarms. In the second case, when $P(M_{2D})$ is very low, SfIS does not interfere either, since SfS is the best reconstruction method when there are not misses. In both cases, $T^{\star} = C - \mathcal{O}$ (see Figure 7.16(b)), forcing the system to always decide Background, and leaving an empty IH.

Figure 7.17 shows how $\frac{P_{SfIS}(Err_{3D})}{P_{SfS}(Err_{3D})}$ varies with different number of occlusions. Note that as $\mathcal{O}$ rises, SfIS has less room for maneuver. In any case, even when $\mathcal{O} = C - 1$, the probability of misclassification of SfIS is never worse than with SfS.

7.6.2 Empirical Results

In the following, we present two different experiments showing the performance of SfIS in front of SfS. The first experiment consists in the reconstruction of the VH and $VH \cup UH$ from a set of synthetic images using different projection tests. The experiment includes quantitative results of the algorithms. In the second experiment, we use real-word images obtained in the smart-room of our lab to show results which can be straight away evaluated from simple observation.

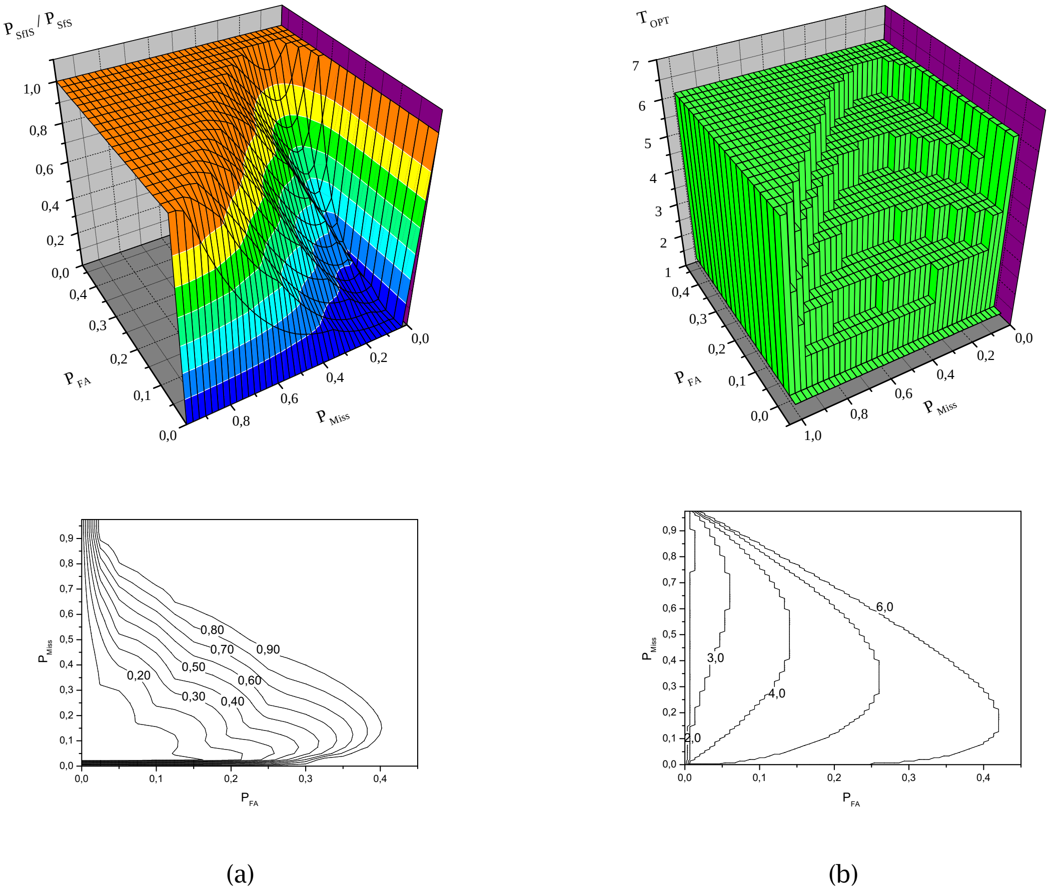
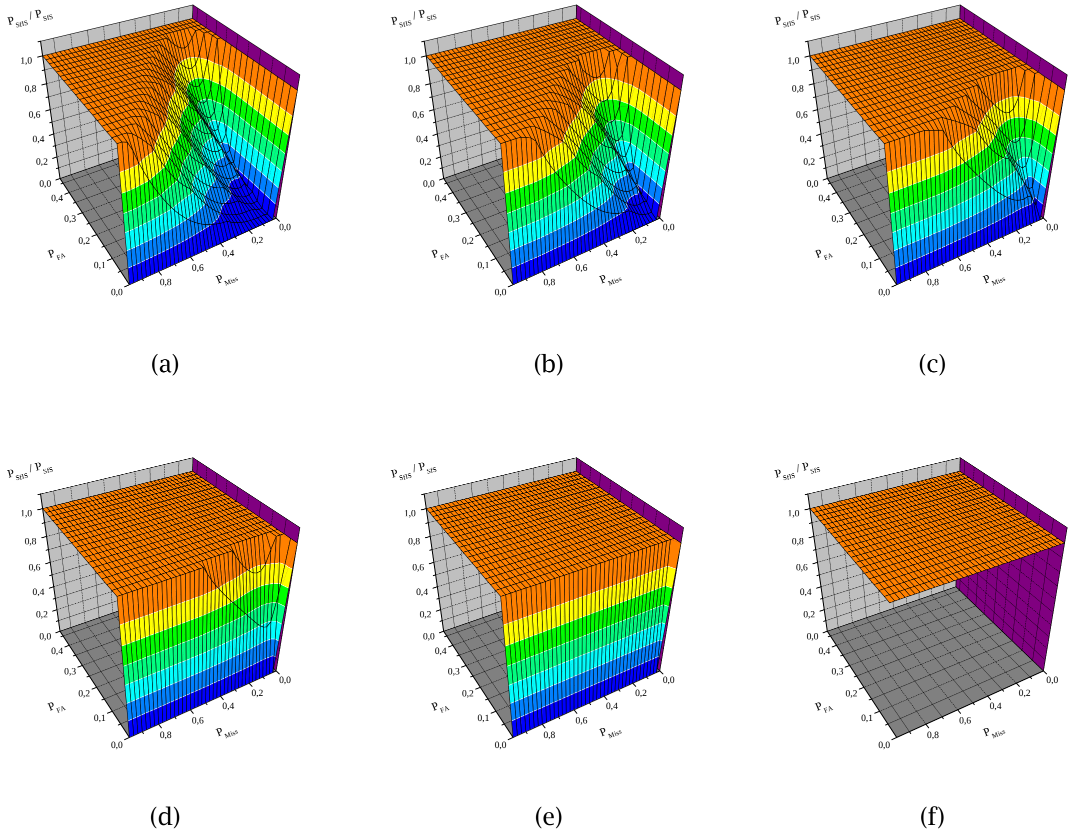


Figure 7.16: The ratio $\frac{P_{SfIS}(Err_{3D})}{P_{SfS}(Err_{3D})}$, and $T^\star$ for different values of $P(FA_{2D})$ and $P(M_{2D})$. Results are shown considering a set-up of 6 cameras with $P_B = 0.9, P_S = 0.1$. In this case, it is assumed that there have not been occlusions ($\mathcal{O} = 0$).

Note that $T^\star = C$ when there are not misses or when the probability of 2D false alarm is high. If $T^\star = C$ then $P_{SfS}(Err_{3D})$ is equivalent to $P_{SfIS}(Err_{3D})$.



(a)

(b)

(c)

(d)

(e)

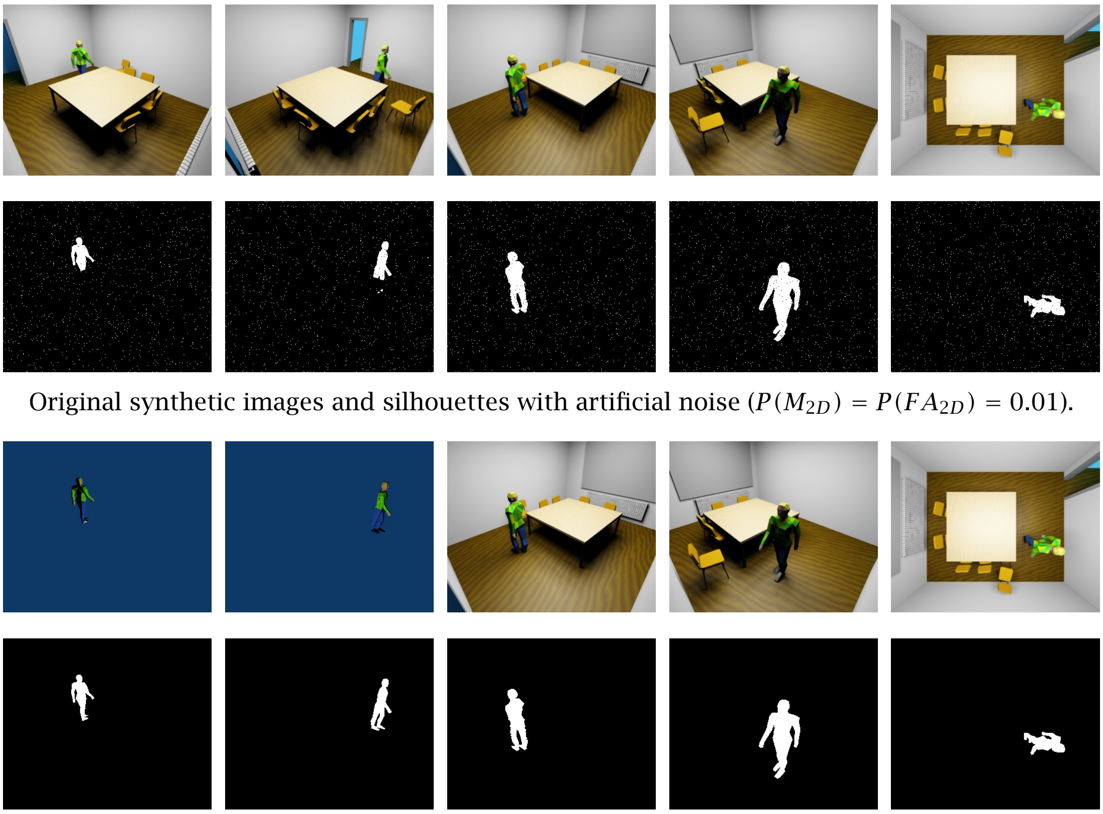
(f)

Figure 7.17: $\frac{P_{SfIS}(Err_{3D})}{P_{SfS}(Err_{3D})}$ probability ratio when there have been 0, 1, 2, 3, 4 and 5 occlusions in (a), (b), (c), (d), (e) and (f), respectively. Results are shown using the same camera set-up as in Figure 7.16.

7.6.2.1 Results with Synthetic Images

In order to obtain quantitative results of the algorithm, we have employed a set of synthetic images because we can arbitrarily add/remove random noise and occlusions to them. In addition, it is possible to easily calculate the ground truth.

The set of images which we have used are shown in Figure 7.18. The first row of images depicts the synthesized scene used in the experiments. In the pictures, a table partially occludes the bottom part of a person in the first two views. The second row of images shows the corresponding set of silhouettes. Note that the silhouettes have some misses and false alarms which have been artificially added. Finally, in the two rows at the bottom, the images and silhouettes corresponding to the noise-free and occlusion-free consistent scene are shown.



Original synthetic images and silhouettes with artificial noise ($P(M_{2D}) = P(FA_{2D}) = 0.01$).

Original synthetic images and silhouettes without occlusions or artificial noise.

Figure 7.18: Set of synthetic images and silhouettes. The dataset, which is a courtesy of J.C. Pujol from the Carlos III University of Madrid, consists of 5 sequences of frames of 352x288 pixels.
In the scene, the cameras, table and chairs are positioned resembling the set-up of the smart-room of the UPC.

The evaluation process is performed as follows. First, a reconstruction from the set of consistent silhouettes, corresponding to the fourth row in Figure 7.18, is obtained to be used as the Ground Truth (*G.T.*). Then, SfS and SfIS algorithms are employed to reconstruct 3D Shapes using the bogus silhouettes of the second row in Figure 7.18. Finally, these Shapes are confronted with the one reconstructed using the consistent set of silhouettes.

In addition to the false alarm rate, miss rate and total error rate, we propose to evaluate the performance of the system employing the verification measures that are commonly used in the information retrieval field:

The Recall, also known as hit-rate or sensitivity, is the ability of a system to recognize all or most of the relevant detections (foreground voxels) in the collection. In our context, the recall is the percentage of foreground voxels that were detected and is based on the number of correct detections and misses:

$$\text{Recall} = \frac{\text{number of True positives}}{\text{number of True positives} + \text{number of False negatives}}, \tag{7.37}$$

that in our particular case corresponds to

$$\text{Recall} = \frac{\text{\#correct Shape detections}}{\text{\#correct Shape detections} + \text{\#misses}}. \tag{7.38}$$

The recall alone does not tell how well the classifiers detect other classes (background). Note that by classifying all the voxels in the room as Shape, the system would have had a recall of 100%.

The Precision is used to indicate the probability that in case that a voxel is detected as part of the Shape, that the voxel really forms part of the specified class.

$$\text{Precision} = \frac{\text{number of True positives}}{\text{number of True positives} + \text{number of False positives}}. \tag{7.39}$$

In our particular case, this rate corresponds to

$$\text{Precision} = \frac{\text{\#correct Shape detections}}{\text{\#correct Shape detections} + \text{\#false Shape detections}}. \tag{7.40}$$

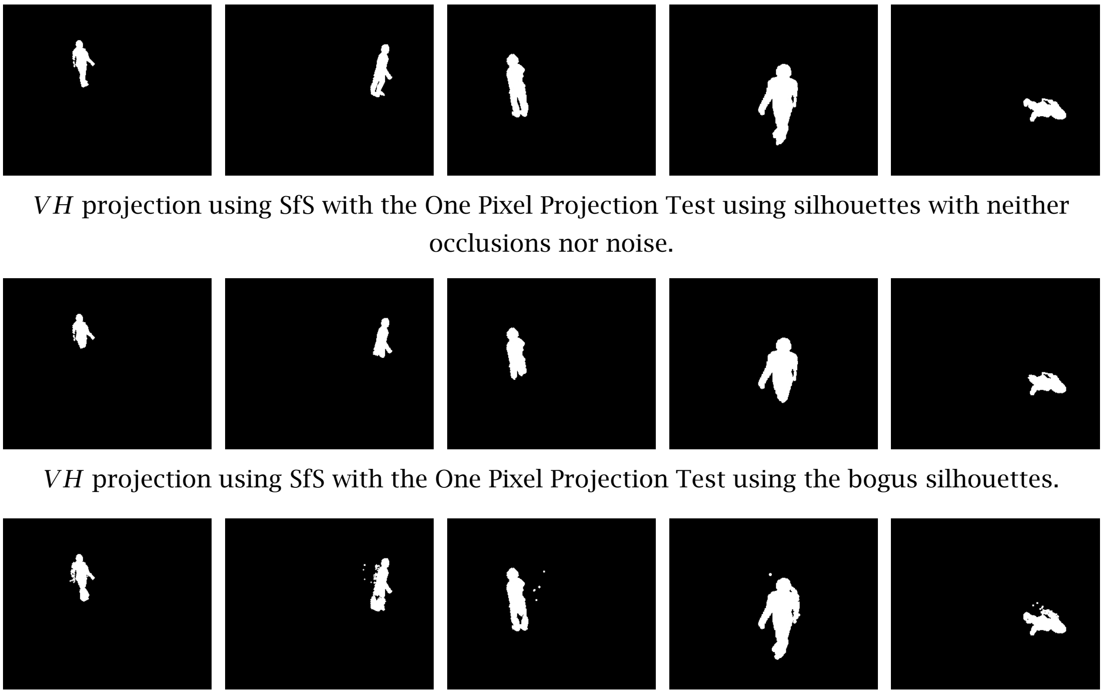
Note that by detecting few foreground voxels, but detecting them correctly, the system would give high precision rates even if the sytem missed most of them. Therefore, a measure that combines precision and recall is needed.

The F-measure, also known as the harmonic mean of precision and recall, measures the combination of the previous evaluation measures and is the measure that we use to evaluate the overall performance of the system:

$$\text{F-measure} = \frac{2 \times \text{Recall} \times \text{Precision}}{\text{Recall} + \text{Precision}}. \tag{7.41}$$

To sum up, recall measures how well the classifier detects the voxels that form part of the Shape and precision measures how well it weeds out the voxels in the background. A well balanced system should have high, similar values of both recall and precision.

We have performed two parallel experiments using the evaluation process and metrics described above. In the first experiment, the One Pixel Projection Test using SfS and SfIS is evaluated. In the second experiment, the Sampled Pixels Projection Test is examined, instead. In the experiments, we have chosen a large size of the edge of the voxel (equivalent to 2.5 cm) so that accurate Shape reconstruction is disregarded in favor of fast reconstructions.



VH projection using SfS with the One Pixel Projection Test using silhouettes with neither occlusions nor noise.

VH projection using SfS with the One Pixel Projection Test using the bogus silhouettes.

$VH \cup UH$ projection using SfIS with the One Pixel Projection Test using the bogus silhouettes.

Figure 7.19: SfIS vs. SfS using 5 cameras, and the One Pixel Projection Test.

Figure 7.19 shows three different types of reconstructions using the One Pixel Projection Test. (1) In the first row, the projections of the Shape reconstructed from the noise-free and occlusion-free images are observed. In this case, the standard SfS algorithm has been used to obtain the voxelized scene. Since the Shape has been obtained from the set of consistent silhouettes, the labeled voxels are used as the Ground Truth (*G.T.*) for comparison in Table 7.2, on page 152. (2) The second row of images corresponds to results of SfS using the noisy and partially occluded silhouettes. Note that most of the errors correspond to 3D misses, as shown in Table 7.2. Therefore, SfS is extremely precise, in this sense. In contrast, it has a low precision rate, meaning that it may detect not all the voxels forming part of the Shape, but those which are detected are usually correctly detected. (3) The row at the bottom shows the projection of

Table 7.2: Results using the One Pixel Projection Test over 96000 voxels

	Ground truth	SfS	SfIS
# Correct detections	5381	4371	**5071**
# False alarms	0	**2**	391
# Misses	0	1010	**310**
F.A. rate	0	$\mathbf{2.08 \times 10^{-5}}$	4.07×10^{-3}
Miss rate	0	0.01	$\mathbf{3.23 \times 10^{-3}}$
Error rate	0	0.01	$\mathbf{7.29 \times 10^{-3}}$
Recall	1	0.81	**0.94**
Precision	1	**0.99**	0.93
F-measure : $\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$	1	0.90	**0.93**

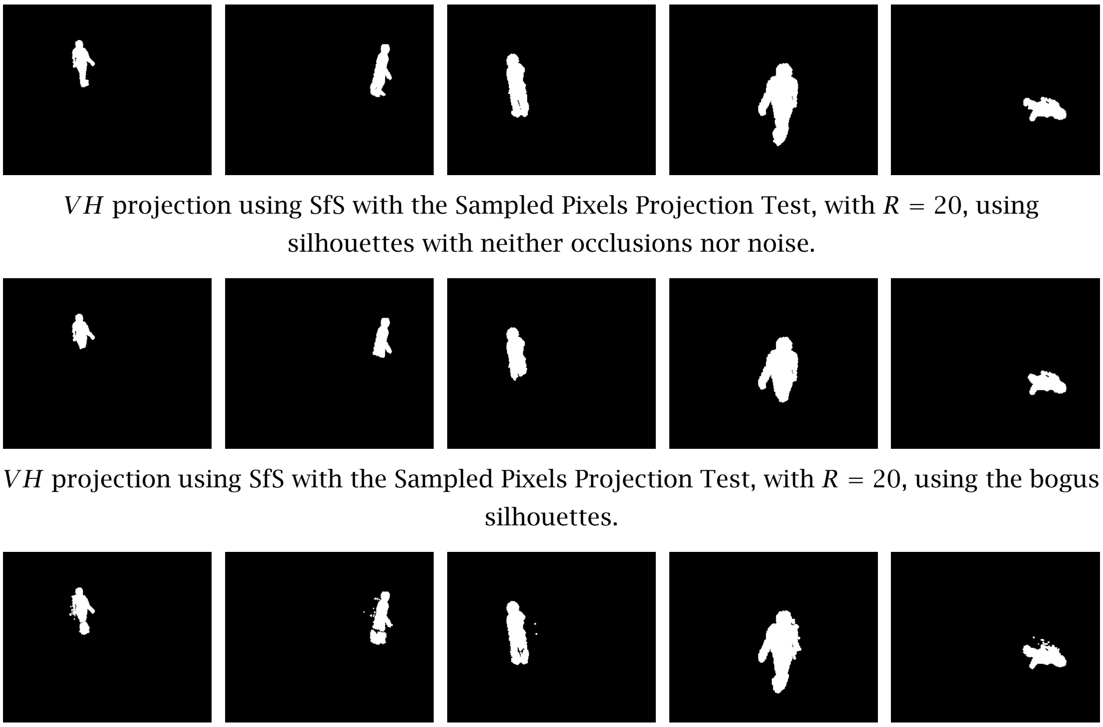
the Shape using the proposed SfIS method. Note that SfIS is able to recover that part of the Shape that the table occluded in the first two views. Besides, observe that SfIS is also successful in not incorporating more false alarms than the recovered misses, as shown in Table 7.2. Also note that some of the wrongly reconstructed voxels can be easily removed in a further stage using simple morphological operations.

Table 7.2 offers another interesting result. Note that SfIS is not only better than SfS w.r.t. the F-measure, but it also has similar precision and recall rates implying and unbiased treatment of error types in 3D.

We want to remark that the data on Table 7.2 (and 7.3, which will be shown later) is only provided to validate the implemented system. That is, we confirm, that, as imposed in the design of the system, we reduce the total error and we can balance the two kind of errors (false alarms and misses). We could consider SfS as a version of SfIS where the threshold taken for the voxels of the Inconsistent Volume was set to a fixed number (C, the number of cameras). We claim that, in most applications, the best threshold is the one than minimizes the total error, that is the one proposed in the previous section.

Table 7.3: Results using the Sampled Pixels Projection Test over 96000 voxels

	Ground truth	SfS	SfIS
# Correct detections	6470	5136	**5802**
# False alarms	0	**4**	571
# Misses	0	1334	**668**
F.A. rate	0	$\mathbf{4.17 \times 10^{-5}}$	5.95×10^{-3}
Miss rate	0	0.01	$\mathbf{6.96 \times 10^{-3}}$
Error rate	0	0.014	**0.013**
Recall	1	0.79	**0.90**
Precision	1	**0.99**	0.91
F-measure : $\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$	1	0.88	**0.91**



VH projection using SfS with the Sampled Pixels Projection Test, with $R = 20$, using silhouettes with neither occlusions nor noise.

VH projection using SfS with the Sampled Pixels Projection Test, with $R = 20$, using the bogus silhouettes.

$VH \cup UH$ projection using SfIS with the Sampled Pixels Projection Test, with $R = 20$, using the bogus silhouettes.

Figure 7.20: SfIS vs. SfS using 5 cameras, and the Sampled Pixels Projection Test.

Figure 7.20 shows analog images to those shown in Figure 7.19. This time, however, the reconstructions have been obtained using the Sampled Pixels Projection Test. (1) In the first row, the projections of the Shape reconstructed from the consistent silhouettes are shown. This Shape is used as the reference in Table 7.3. (2) The second row of images corresponds to results of SfS using the bogus silhouettes. (3) And finally, the row at the bottom shows the projection of the Shape using the proposed SfIS method. The empirical results using the Sampled Pixels Projection Test are shown in Table 7.3.

Table 7.3 confirms that F-rate in SfIS is lower than in SfS also when combined with the Sampled Pixels Projection Test. The table also confirms that errors are equitably distributed between recall and precision when using the SfIS algorithm.

However, interestingly, the F-measure using the One Pixel Projection Test is larger than the one using the Sampled Pixels Projection Test (see Tables 7.2 and 7.3, respectively). This has the following explanation: Since the random noise that we have added is very low, the Sampled Pixels Projection Test takes values of N close to 1, when R is not too high (see Table 7.1, on page 127 and refer to similar considerations for SPOT in [CKBH00]). This makes the Sampled Pixels Projection Test to have larger error probabilities than the One Pixel Projection Test in this case. Indeed, the Sampled Pixels Projection Test performs better than the One Pixel Projection Test in other more noisy environments such as the one which was presented in Fig 7.8. An important conclusion that we can reach is that very simple projection tests such as the One Pixel Test can be more effective than sophisticated ones such as the Sampled Pixels Test, depending on the working environment. The scenario in which the projection test is going to be used will have to be carefully examined before choosing one or another approach.

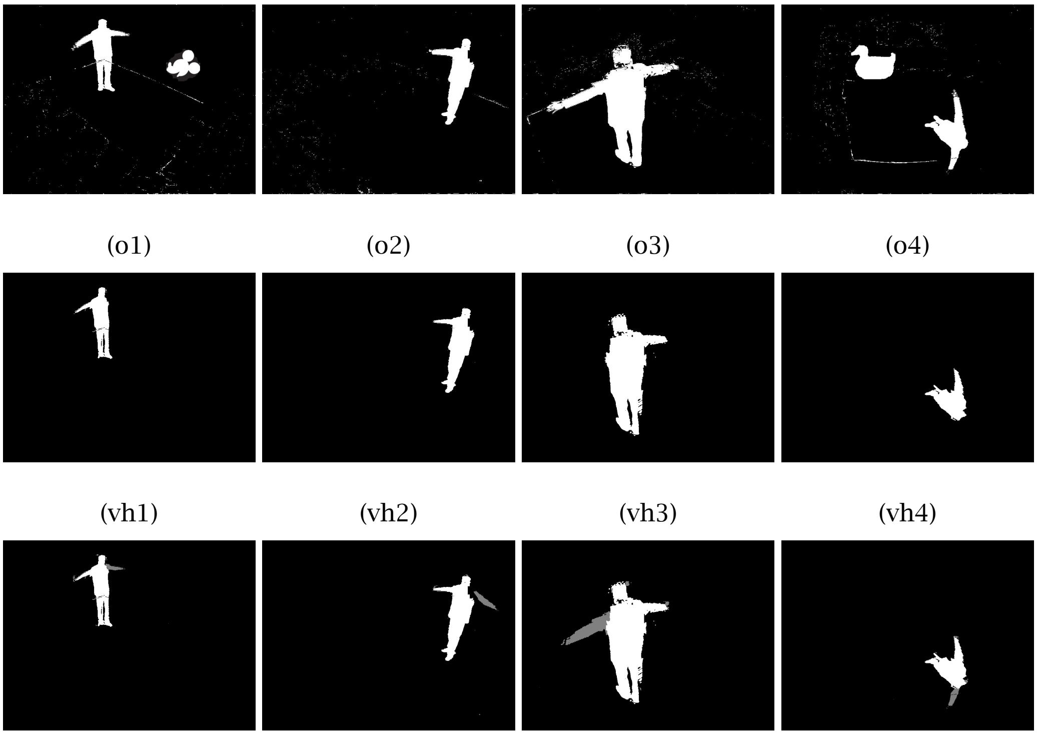
In any case, independently of the projection test, SfIS is always useful because it balances errors between false alarms and misses without increasing the error (F-measure).

7.6.2.2 Results with Real-World Images

In Figure 7.21, a real world scenario is shown. In this case, the foreground segmentation has been done using [SG00b], and we have added some additional false alarms in (o1) and (o4).

The experiment has been performed using a very high-resolution volume, employing voxels with edge size of less than 5 mm. The underlying idea is to guarantee that the projection of the splat of any voxel in the shape is comprised within a pixel in all the silhouettes. Therefore, the reconstruction is independent of the projection test and $P(FA_{2D})$ and $P(M_{2D})$ concur with the probabilities of FA and Miss of the background learning technique. In this case, we are assuming $P_{FA}(2D) = P_M(2D) = 0.1$, and P_B has been selected based on the percentage of voxel occupancy in the VH.

In (o2), the silhouette's left arm has not been detected due to the similar colour to its background counterpart. The second row of images shows the projection of the VH, reconstructed



(o1) (o2) (o3) (o4)

(vh1) (vh2) (vh3) (vh4)

(p1) (p2) (p3) (p4)

Figure 7.21: Silhouettes, projection of the VH and projection of the $VH \cup UH$, in first, second and third row, respectively.

using the standard voxel-based SfS algorithm. Note that the misdetection in (o2) has been propagated to the rest of silhouettes. The row of images on the bottom shows the projection of the $VH \cup UH$ in white and gray, respectively. Observe that the projection of the arm is recovered, even in (p2), while remaining unaffected to the artificial false alarms.

The experiment shows that SfIS can be used to recover some of the errors produced in the 2D foreground segmentation techniques by exploiting the redundancy present in a multi-camera setup. On the contrary, SfS does not only fail to recover these types of errors but it actually worsens all the silhouettes by propagating the 2D misses from one view to the rest of views.

To complete the experiments, we have considered appropriate to include here a set of tests comparing SfIS with other approaches using real world video sequences. In addition, since we are using video sequences, we have incorporated the Bayesian cooperative approach described in §7.5 to the set of evaluated sequences. This time, the experiment has been performed using a low-resolution volume, employing voxels with edge size 2.5 cm, therefore prioritizing fast 3D detections over a more accurate Shape.

Also, in this occasion, since we are using real-world images with imprecise calibration, we have opted to indirectly evaluate the performance of the reconstruction methods. To do so, we have compared the projection of the 3D volumes with a set of five manually classified silhouettes from images that have been randomly selected in a video sequence. These manually labeled silhouettes are the ground truth in this case.

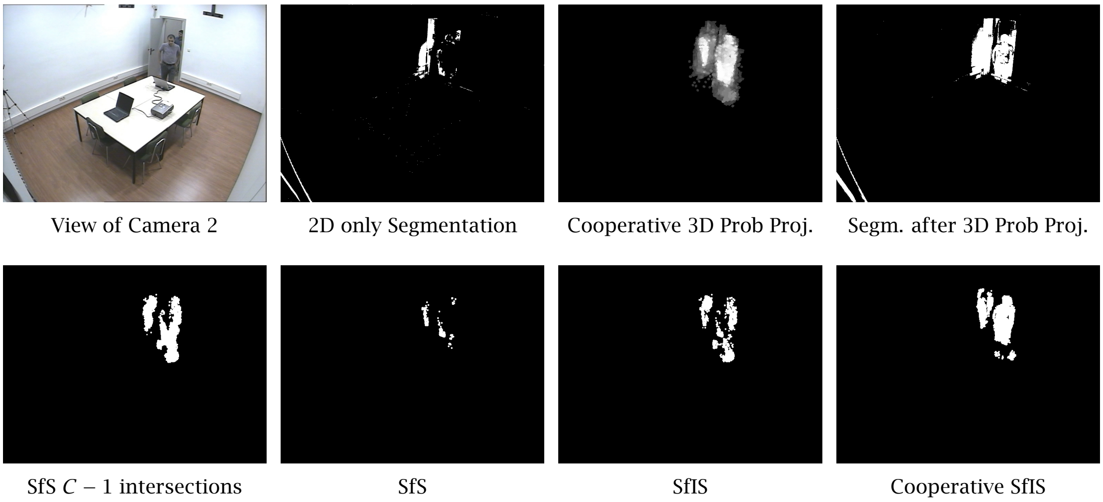
Four different techniques have been compared. The first technique is a version of SfS where a voxel is not classified as Shape if there are *more than one* views where its Projection Test fails. We identify this method as *SfS* $C - 1$ *intersections* in our experiments. The second evaluated technique is traditional SfS. Third and fourth tested methods are SfIS and cooperative SfIS, respectively. In this experiment, we have always employed the One Pixel Projection Test in all the methods for a fair comparison.

The pixel models employed for 2D classifications are a single Gaussian per pixel for the background and a uniform distribution for the foreground. 2D classifications are obtained using MAP and the models are updated using EM. The pixels models classification and update steps were described in chapter 4. In this experiment, the models adaptation speed corresponds to the probability after MAP of 2D models except in the cooperative SfIS approach that uses the projection of the 3D probabilistic representation described in §7.5.

For visual inspection purposes, we present two figures (Figure 7.22 and Figure 7.23) with results, corresponding to different times and camera views of a scene. As it can be observed, these tests were performed in the smart-room of the UPC. In the first row of images for all the camera groups, the original view and some intermediate results are presented and, in the second row, the projections of the Shapes obtained with the methods under evaluation are shown.

In Figure 7.22, the images corresponding to camera 2 in frame number 175 are shown. Note that the 2D only segmentation (2nd column, 1st row) -not using 3D redundancy information- has failed in this camera due to the similar colors of the person in the foreground and the clutter in the background. However, see that the projected voxel probabilities using the cooperative SfIS approach (3rd column, 1st row) do correct these errors and, therefore, 2D segmentations using the cooperative approach are more precise (4th column, 1st row).

Similar problems are observable in Figure 7.23. The figure corresponds to frame 650 and shows three out of the five camera views used in all the methods. Note that 2D misses in a view are transferred to the rest of views in the SfS approach. The SfS $C-1$ approach does not propagate 2D misses but incorporates many false alarms leading to larger Shapes and silhouettes' projections. As it can be observed from the images, SfIS is a good approach for not propagating 2D misses as well as for not incorporating many false alarms. The cooperative SfIS approach behaves even better than SfIS because it informs the 2D models when an error is made and, thus, the pixels models are updated with a more-informed strategy.


View of Camera 2

2D only Segmentation

Cooperative 3D Prob Proj.

Segm. after 3D Prob Proj.

SfS $C-1$ intersections

SfS

SfIS

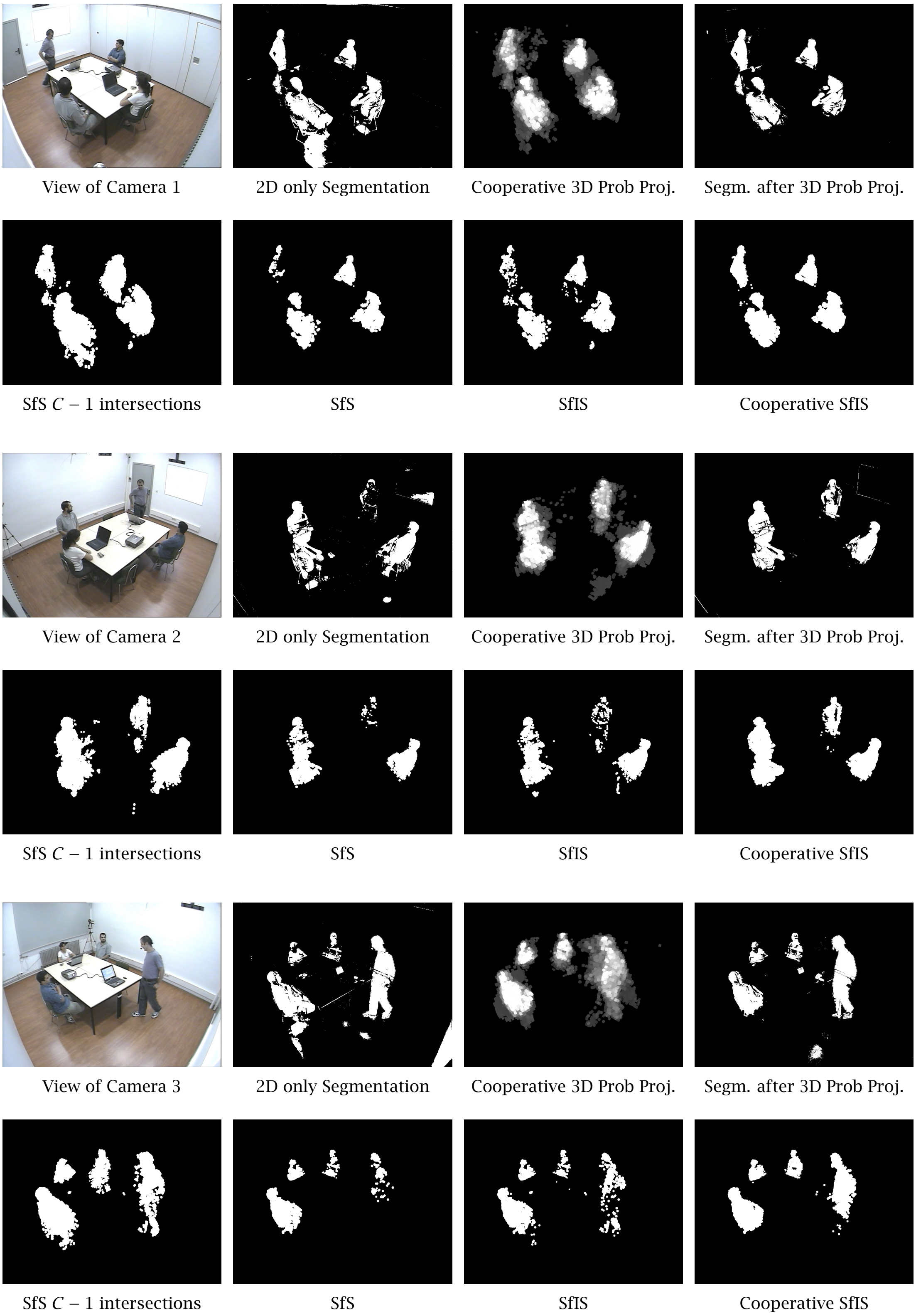
Cooperative SfIS

Figure 7.22: Silhouettes and 3D volumetric projections corresponding to frame 175 with different techniques using the One Pixel Projection Test.

Quantitative results of this experiment are presented in Table 7.4. These results have been obtained by averaging the number of 2D false alarms, 2D correct detections and 2D misses over a set of projected reconstructions. These projections correspond to the five views where the silhouettes were manually labeled to be the ground truth, as previously commented.

Some interesting conclusions can be extracted from the table.

Note that the SfS $C-1$ approach has the highest recall rate. Indeed, it also has a very large number of false alarms and, therefore, the poor precision rate, but it is a good method if we want to be sure to detect the foreground voxels when they exist.



View of Camera 1 | 2D only Segmentation | Cooperative 3D Prob Proj. | Segm. after 3D Prob Proj.

SfS $C-1$ intersections | SfS | SfIS | Cooperative SfIS

View of Camera 2 | 2D only Segmentation | Cooperative 3D Prob Proj. | Segm. after 3D Prob Proj.

SfS $C-1$ intersections | SfS | SfIS | Cooperative SfIS

View of Camera 3 | 2D only Segmentation | Cooperative 3D Prob Proj. | Segm. after 3D Prob Proj.

SfS $C-1$ intersections | SfS | SfIS | Cooperative SfIS

Figure 7.23: Silhouettes and 3D volumetric projections corresponding to frame 650 with different techniques using the One Pixel Projection Test.

Table 7.4: System Evaluation through the Projection of 3D Reconstructions in Video Sequences over 442368 pixels

	Ground truth	SfS ($C-1$ int.)	SfS	SfIS	Coop. SfIS
# Correct foreground det.	32279	**27471**	15023	20445	25061
# False alarms	0	29760	**5077**	7529	6872
# Misses	0	**4808**	17256	11834	7218
F.A. rate	0	0.07	**0.01**	0.02	0.02
Miss rate	0	**0.01**	0.04	0.03	0.02
Error rate	0	0.08	0.05	0.04	**0.03**
Recall	1	**0.85**	0.47	0.63	0.77
Precision	1	0.48	0.75	0.73	**0.78**
F-measure : $\frac{2\times\text{Recall}\times\text{Precision}}{\text{Recall}+\text{Precision}}$	1	0.62	0.57	0.68	**0.77**

In contrast, traditional SfS is very precise, even more than SfIS. This should not be a surprise as we previously mentioned in the results with synthetic images. Note that SfS detects fewer voxels but it is very good at asserting that those voxels form part of the Shape.

SfIS and cooperative SfIS are the most balanced methods. They have high precision and recall rates and their F-measures are the best. SfIS improves when combined with the cooperative Bayesian framework since the 3D information is continuously flowing to 2D in a probabilistic manner.

In conclusion, the cooperative SfIS approach definitely has the lowest error rate and best F-score of them, as simple visual inspection of the images confirms and it is the method that performs best when operating with video sequences[1].

7.7 Conclusion and Future Work

In this chapter we have presented a novel scheme for effective Shape from Silhouette using sets of inconsistent silhouettes as usually found in practical scenarios. The scheme exploits the consistency principle, and performs an error detection and correction procedure of the most probable consistent silhouettes according to the available data.

First, we have introduced a method to determine the IH, i.e., the volumetric zones leading to inconsistent regions in the silhouettes. Then, we have described a voxel-based technique -which works with any projection test- to enumerate the number of inconsistent cone intersections. We have also proposed a method to obtain the minimum number of inconsistent

[1] Due to the great number of images resulting from all the methods compared, the number of camera views and the large time interval evaluated, it is not possible to show here the complete set of resulting images. However, the video sequences with all the evaluated methods at all the cameras views can be obtained in `http://gps-tsc.upc.es/imatge/_jl/`

cone intersections $T^\star$ that have to be produced so that it can be determined that an object was wrongly missed by standard SfS. Threshold $T^\star$ is taken after minimization of the probability of voxel misclassification. In addition, we have stated the conditions in which a very fast implementation of SfIS is possible. Finally, we have presented the SfIS cooperative approach that obtains the best results in video sequences by giving feedback to the background learning techniques to make more trustworthy background models.

SfIS has proved to be an effective 3D reconstruction tool. We have given theoretical prove that SfIS misclassification probability is lower than the one using SfS. Experimental results have also been carried out showing that SfIS does not only reduce the number of errors but it is also successful in balancing errors between 3D false alarms and misses, contrary to conventional SfS that mainly introduces only 3D misses. Indeed, SfIS introduces false alarms to the Shape. However false alarms are introduced ensuring that the global error rate is reduced.

Finally, SfIS can be used in at least two different manners.

- (1) To extract better volumes by minimizing the effects that inconsistencies have over the reconstructed Shape.
- (2) To recover errors in the silhouettes from informed decisions made at the volume level, where individual detections at the 2D level are compared for consistency. Contrarily, conventional SfS does not only fail to recover errors in the silhouettes but it worsens the silhouettes by propagating 2D misses from one view to the others.

Certainly, there are some research topics in which we are interested to extend the work presented here. For instance, we have shown that SfIS increases the rate of false alarms as far as the global misclassification probability is reduced. Even so, we believe that many of these false alarms can be easily removed in a posterior stage by using morphological filters, leading to even better results. A geometric implementation of SfIS is another of the tasks we are interested for the future.

Chapter 8

Contributions and Future Work

8.1 Contributions

In this thesis we have proposed a unified framework for the planar and volumetric foreground detection tasks. The proposed framework operates in a collaborative manner by transferring more-informed 3D probabilistic information back to the planar detectors in each image that, in turn, also become more precise.

8.1.1 Contributions to planar activity detection

The cooperative planar-volumetric detector was achieved by first extending planar detection techniques to a Bayesian framework [LP06]. We set the bases for maximum a posteriori classification and introduced model update equations based on expectation maximization. The derivations of the equations for pixel classification and update can be found in chapter 4. These equations show that for proper Bayesian update, new background observations have to be incorporated to the background models proportionally to the probability of the background class. This makes a big difference with some of the current methods that update the models with heuristic rules without proper justification.

In addition, in chapter 5 we showed some of the possible applications of such planar activity detections, including two-dimensional trackers [LPX04] and a system for suspicious object detection [LPX05a]. We also explained how to detect and suppress shadows and specular reflections form original detections [LPX05b]. Some of these key technologies have recently been granted a patent [XL07a] (another one is pending [XL07b]) as well as appeared in journal [XLL04] and book chapter [SW05] publications. In addition, we have also applied foreground detection to other technologies not described in the text, including gestural user interfaces, 3D immersive video-communications applications [LMM$^+$07] and behavior modeling systems.

8.1.2 Contributions to volumetric activity detection

The planar-volumetric cooperative framework could not have been possible without first developing a Bayesian setting for the planar detection task. As it has been previously mentioned, the speed of adaptation of a pixel's background model is a function of the probability of the background class. This finding permitted us to employ the update scheme of the planar activity detectors to include higher-level probabilistic information obtained by other means. In this way, in chapter 6 we developed a new framework in which 3D probabilistic information is attained from 2D probabilistic maps and then projected back to each one of the original views to be used as the update speed of the two-dimensional background models [LP06].

In chapter 7, we reexamined the volumetric activity detection task. Following a different line of thought, we studied the coherence between planar and volumetric detectors and developed a novel technique, called Shape from Inconsistent Silhouette (SfIS) [LPC06, LP07a]. Basically, SfIS is able to reclassify some of the initial volumetric detections so that the misclassification error is minimized. SfIS can be used to extract better volumes by minimizing the effects that inconsistencies have over the reconstructed Shape. In addition, SfIS can also be used to recover errors in the silhouettes from more-informed decisions made at the volume level, where individual detections at the 2D level are compared for consistency. Finally, reassigned classifications can be introduced back into our Bayesian cooperative framework providing better long-term video activity detections in both the 2D and 3D domains.

We have also shown evidence of the usefulness of three-dimensional detectors, including 2D shape correction from multi-camera information, 3D trackers [LP05] as well as multi-modal 3D trackers [ACFS$^+$06], merging audio-visual information. Some of the technologies described have been used in real-time demonstrators for person localization and tracking in smart room environments.

Finally and to conclude, we have proposed an integrated, Bayesian, consistency-aware 2D/3D foreground detection system that operates in a collaborative manner by transferring more-informed 3D probabilistic information back to the planar detectors in each image that, in turn, also become more precise. Some of the key contributions of this dissertation have been abridged in [LP07b].

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 supported by the EU through the Integrated Project CHIL IST-2004-506909 (Computers in the Human Interaction Loop) and by the Networks of Excellence SCHEMA IST-2001-32795 (Content-Based Semantic Scene Analysis & Information Retrieval), SIMILAR IST-2002-507609 (Human-machine interfaces SIMILAR to human-human communication) and MUSCLE (Multimedia Understanding Through Semantics, Computation and Learning). In addition, this work has also been developed within the framework of the Spanish project TEC2004-01914 (Analysis, coding and semantic indexation in controlled environments).

8.2 Future work

Certainly, there are several research topics in which we are interested to extend the work presented here.

- In this thesis we have initially developed a Bayesian framework for the planar foreground classification task. The setting presented operates by using both foreground and background models of the pixel process. We have performed an in-depth study of the background modeling process. However, we believe that incorporating more sophisticated foreground models into our framework can bring better classification results.

  Foreground entities can be characterized by means of geometrical models that take into account the chromatic and structural characteristics of the entities. A tracker has then to be used to update the foreground models of each entity along the time. Then, in our per-pixel MAP setting, the model of each entity has to be mapped to each pixel before performing the foreground segmentation. The only condition that foreground models have to fulfill is that the mapping of the foreground models to each pixel must be expressed as a probability density function.

- The maximum a posteriori setting in the planar detection task permits detecting foreground and background pixels as an independent process for each pixel in the image. Besides, we have shown several post-processing techniques which make use of the global context of the image to improve the results of the classification.

  In order to take into account the similarity of the observations in neighboring pixels, it is possible to employ region-based models that assume statistical dependence between pixels in the same vicinity. These models can be incorporated into our cooperative framework as long as they are expressed in probabilistic terms. We believe that assuming pixel interdependency can boost classification performance at the expense of slower operation speed.

- In both planar and volumetric approaches, spatio-temporal continuity can also be exploited, giving higher priors to the classes in which pixels and voxels are usually detected. In this sense, predicting pixels and voxels by means of Kalman or particle filters can also give more precise detections. In general, any prior information can be incorporated into our scheme and, therefore, this makes it easy to integrate different types of technologies.

- We have presented SfIS as a method that constructs 3D volumes considering the effect of 2D noise and systematic errors on the accuracy of voxel-based SfS. However, apart from noise in the silhouettes, the shape estimation accuracy of SfS is also greatly affected by errors in the camera projection parameters, which mainly come from inaccurate camera calibration.

  An in-depth theoretical and quantitative analysis of how SfS is affected by camera projection errors would come in handy in deriving an optimal SfIS algorithm for both calibration and noise/systematic errors.

- Finally, we believe it is possible to port voxel-based SfIS to other types of Shape from Silhouette. For instance, there are a set of techniques called free-view video that make use of the SfS principle to generate new images from arbitrary point of views. These techniques do not explicitly reconstruct the 3D space, but use the silhouettes as the constraint for rendering novel views. Another different type of SfS is the geometrical-based SfS, described in § 2.2.4.1 on page 27.

  In order to take the idea of SfIS a bit further, one only needs to find the Inconsistent Hull for each one of the SfS approaches of interest and then reuse the scheme presented in chapter 7. In this respect, we are currently investigating SfIS for free-view video and have verified that the same concept is easily portable to non voxel-based approaches.

References

[ACFS$^+$06] Abad A., Canton-Ferrer C., Segura C., Landabaso J.L., Macho D., Casas J.R., Hernando J., Pardàs M., and Nadeu C. UPC audio, video and multimodal person tracking systems in the CLEAR evaluation campaign. In *Multimodal Technologies for Perception of Humans*, volume 4122 of *Lecture Notes in Computer Science*, pages 93–104. Springer, 2006. 90, 162

[Att99] Attias H. Inferring parameters and structure of latent variable models by variational Bayes. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 21–30. Morgan Kaufmann, San Francisco, CA, 1999. 56

[Aug03] August T.M. The 'summation hack' as an outlier model. Technical report, Department of Statistics, Carnegie Mellon University, 2003. URL `http://www.stat.cmu.edu/minka/papers/minka-summation.pdf`. 104

[Bau74] Baumgart B.G. *Geometric Modeling for Computer Vision.* Ph.D. thesis, CS Department, Stanford University, October 1974. AIM-249, STAN-CS-74-463. 99, 111

[BER02] Black J., Ellis T., and Rosin P. Multi view image surveillance and tracking. In *Proceedings of the Workshop on Motion and Video Computing.* IEEE Computer Society, 2002. 16

[Bev03] Bevilacqua A. Effective shadow detection in traffic monitoring applications. In *Proceedings of the International Conference in Central Europe on Computer Graphics, Visualization and Computer Vision.* 2003. 76

[Bil98] Bilmes J.A. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical Report TR-97-021, Department of Electrical Engineering and Computer Science, U.C. Berkeley, April 1998. 16, 62

[BL03] Bottino A. and Laurentini A. Introducing a new problem: Shape from Silhouette when the relative positions of the viewpoints is unknown. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(11):1484–1493, 2003. ISSN 0162-8828. 26, 112

[Bou] Bouguet J.Y. Camera calibration toolbox for Matlab. URL `http://vision.caltech.edu/bouguetj/calib_doc/`. 25

[BV04] Boyd S. and Vandenberghe L. *Convex Optimization*. Cambridge University Press, New York, NY, USA, 2004. ISBN 0521833787. 137

[CCSS01] Chuang Y.Y., Curless B., Salesin D.H., and Szeliski R. A Bayesian approach to digital matting. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 264–271. IEEE Computer Society, December 2001. 112

[CFCTP05] Canton-Ferrer C., Casas J.R., Tekalp M., and Pardàs M. Projective Kalman filter: Multiocular tracking of 3D locations towards scene understanding. In *Proceedings of Multimodal Interaction and Related Machine Learning Algorithms*, Lecture Notes in Computer Science. Springer, Edinburgh, UK, 2005. 22

[CGP$^+$04] Cucchiara R., Grana C., Prati A., Tardini G., and Vezzani R. Using computer vision techniques for dangerous situation detection in domotic applications. *IEE Intelligent Distributed Surveilliance Systems*, pages 1–5, February 2004. ISSN 0537-9989. 74

[CGPP03] Cucchiara R., Grana C., Piccardi M., and Prati A. Detecting moving objects, ghosts, and shadows in video streams. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(10):1337–1342, 2003. 10, 15, 76

[Che02] Chen B. Building a projection autostereoscopic display. Technical report, Stanford Computer Graphics Laboratory, 2002. URL `http://graphics.stanford.edu/~billyc/research/autostereo/autostereo.pdf`. 19

[CKBH00] Cheung K.M., Kanade T., Bouguet J.Y., and Holler M. A real time system for robust 3D voxel reconstruction of human motions. In *Proceedings of Computer Vision and Pattern Recognition*, volume 2, pages 714 – 720. IEEE Computer Society, June 2000. 30, 35, 112, 115, 122, 154

[CLRS01] Cormen T.H., Leiserson C.E., Rivest R.L., and Stein C. *Introduction to Algorithms, Second Edition*. The MIT Press, September 2001. ISBN 0262032937. 141

[Com] Computers in the Human Interaction Loop (CHIL) EU Project. URL `http://chil.server.de`. 125

[CSE05] Cavallaro A., Steiger O., and Ebrahimi T. Tracking video objects in cluttered background. *IEEE Transactions on Circuits and Systems for Video Technology*, 15(4):575–584, 2005. 5

[Day90] Day A.M. The implementation of an algorithm to find the Convex Hull of a set of three-dimensional points. *ACM Transactions on Graphics*, 9(1):105–132, 1990. ISSN 0730-0301. 25

[DBT03] Dockstader S.L., Berg M.J., and Tekalp A.M. Stochastic kinematic modeling and feature extraction for gait analysis. *IEEE Transactions on Image Processing*, 12(8):962–976, 2003. 22

[DLR77] Dempster A.P., Laird N.M., and Rubin D.B. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society*, 39(1):1–38, 1977. 16, 56, 61, 62

[Dye01] Dyer C.R. Volumetric scene reconstruction from multiple views. In *Foundations of Image Understanding*, pages 469–489. Kluwer, 2001. URL `ftp://ftp.cs.wisc.edu/computer-vision/repository/PDF/dyer.2001.fia.pdf`. 27, 112

[EDHD99] Elgammal A., Duraiswami R., Harwood D., and Davis L.S. Non-parametric model for background subtraction. In *Proceedings of International Conference on Computer Vision*. IEEE Computer Society, Sept 1999. 7, 8, 9, 10, 13, 15, 38, 42, 43, 44, 77, 112

[ES02] Esteban C.H. and Schmitt F. Multi-stereo 3D object reconstruction. In *Proceedings of International Symposium on 3D Data Processing Visualization and Transmission*, pages 159–167. June 2002. 25

[Fau93] Faugeras O. *Three-dimensional computer vision: a geometric viewpoint*. MIT Press, Cambridge, MA, USA, 1993. ISBN 0-262-06158-9. 22

[FB05] Franco J.S. and Boyer E. Fusion of multi-view silhouette cues using a space occupancy grid. In *Proceedings of International Conference on Computer Vision*. IEEE Computer Society, October 2005. URL `http://perception.inrialpes.fr/Publications/2005/FB05a`. 100

[FF95] Fitzgibbon A.W. and Fisher R.B. A buyer's guide to conic fitting. In *Proceedings of British Machine Vision Conference*, pages 513–522. BMVA Press, Surrey, UK, UK, 1995. ISBN 0-9521898-2-8. 91, 92

[FR97] Friedman N. and Russell S.J. Image segmentation in video sequences: A probabilistic approach. In *Proceedings of Conference on Uncertainty in Artificial Intelligence*, pages 175–181. 1997. 7, 8, 112

[FVB03] Forbes K., Voigt A., and Bodika N. Using silhouette consistency constraints to build 3D models. In *Proceedings of Fourteenth Annual South African Workshop on Pattern Recognition*. PRASA, 2003. 109, 117

[Gar04] Garcia O. *Mapping 2D images and 3D world objects in a multicamera system*. Master's thesis, Image Processing Department, Technical University of Catalunya, 2004. URL `http://gps-tsc.upc.es/imatge/_JosepRamon/PFC/reports/PFC_OscarGarcia_041008.pdf`. 22

[GHF86] Goldfeather J., Hultquist J.P.M., and Fuchs H. Fast constructive-solid geometry display in the pixel-powers graphics system. In *Proceedings of International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pages 107–116. ACM Press, New York, NY, USA, 1986. ISBN 0-89791-196-2. 27, 112

[GMBT04] Georis B., Maziere M., Bremond F., and Thonnat M. A video interpretation platform applied to bank agency monitoring. *IEE Intelligent Distributed Surveilliance Systems*, pages 46–50, February 2004. ISSN 0537-9989. 74

[Gra] Graphics Optics Vision group of Max Planck Institut fur Informatik. Kung Fu Girl dataset. URL `http://www.mpi-inf.mpg.de/departments/irg3/kungfu`. 28

[HF01] Haritaoglu I. and Flickner M. Detection and tracking of shopping groups in stores. In *Proceedings of Computer Vision and Pattern Recognition*, volume 1. IEEE Computer Society, 2001. ISSN 1063-6919. 74

[HHD99] Horpraset T., Harwood D., and Davis L. A statistical approach for real-time robust background subtraction and shadow detection. In *Proceedings of International Conference on Computer Vision*. IEEE Computer Society, 1999. 7, 42, 76, 79, 112

[HHD00] Haritaoglu, Harwood D., and Davis L. W4: Real time surveillance of people and their activities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, August 2000. 7, 8, 10, 13, 42, 44, 76, 77, 112

[HZ04] Hartley R.I. and Zisserman A. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004. 19, 22, 24, 102

[ISBG99] Ivanov Y.A., Stauffer C., Bobick A.F., and Grimson W. Video surveillance of interactions. In *Proceedings of Computer Vision and Pattern Recognition*. IEEE Computer Society, 1999. 7, 44

[JD03] Juszczak P. and Duin R.P.W. Uncertainty sampling methods for one-class classifiers. In *Proceedings of the ICML Workshop on Learning from Imbalanced Data Sets*. 2003. 12, 52

[JDWR00] Jabri S., Duric Z., Wechsler H., and Rosenfeld A. Detection and location of people in video images using adaptive fusion of color and edge information. In *Proceedings of International Conference on Pattern Recognition*. IEEE Computer Society, 2000. 7, 8, 112

[JS02] Javed O. and Shah M. Tracking and object classification for automated surveillance. In *Proceedings of European Conference on Computer Vision*, pages 343–357. 2002. 7, 76, 77, 112

[KB01] Kaewtrakulpong P. and Bowden R. An improved adaptive background mixture model for real-time tracking with shadow detection. In *Proceedings of 2nd European Workshop on Advanced Video Based Surveillance Systems*, volume 5308. 2001. 14, 65

[KCM83] Kirkpatrick S., Cd G., and Mp V. Optimization by simulated annealing. *Science*, 220(4598):671–680, 1983. 56

[KCM03] Kang J., Cohen I., and Medioni G. Soccer player tracking across uncalibrated camera streams. In *Proceedings of Performance Evaluation of Tracking and Surveillance*. 2003. 16

[KHDM98] Kittler J., Hatef M., Duin R.P.W., and Matas J. On combining classifiers. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(3):226–239, 1998. ISSN 0162-8828. 104, 105

[KML04] Kung S.Y., Mak M.W., and Lin S.H. *Biometric Authentication: A Machine Learning Approach*. Prentice Hall PTR, ISBN: 978-0131478244, first edition, 2004. 61

[KS00a] Khan S. and Shah M. Tracking people in presence of occlusion. In *Proceedings of Asian Conference on Computer Vision*. 2000. 7, 10, 13, 44, 112

[KS00b] Kutulakos K.N. and Seitz S.M. A theory of shape by space carving. *International Journal of Computer Vision*, 38(3):199–218, 2000. ISSN 0920-5691. 26

[KWH$^+$94] Koller D., Weber J., Huang T., Malik J., Ogasawara G., Rao B., and Russell S. Toward robust automatic traffic scene analyis in real-time. In *IEEE International Conference on Pattern Recognition*, pages 126–131. 1994. 15

[Lau91] Laurentini A. The Visual Hull: A new tool for contour-based image understanding. In *Proceedings of Seventh Scandinavian Comperence on Image Processing*, pages 993–1002. 1991. 26, 99, 112

[Lau94] Laurentini A. The Visual Hull concept for silhouette-based image understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. ISSN 0162-8828. 26, 99, 112, 114

[Lau95] Laurentini A. How far 3D shapes can be understood from 2D silhouettes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):188–195, 1995. ISSN 0162-8828. 26, 99, 112

[LFP98] Lipton A.J., Fujiyoshi H., and Patil R.S. Moving target classification and tracking from real-time video. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, pages 8–14. Princeton, NJ, USA, October 1998. 74

[LHGT02] Li L., Huang W., Gu I.Y.H., and Tian Q. Foreground object detection in changing background based on color co-occurrence statistics. In *Proceedings of IEEE Workshop on Applications of Computer Vision*, page 269. IEEE Computer Society, Washington, DC, USA, 2002. ISBN 0-7695-1858-3. 7, 112

[LHGT04] Li L., Huang W., Gu I.Y.H., and Tian Q. Statistical modeling of complex backgrounds for foreground object detection. *IEEE Transactions on Image Processing*, 13(11):1459–1472, 2004. 7, 10, 13, 44, 112

[LL02] Li L. and Leung M.K.H. Integrating intensity and texture differences for robust change detection. *IEEE Transactions on Image Processing*, 11(2):105–112, 2002. 7, 112

[LMM+07] Landabaso J.L., Mansi T., Molina C., Zangelin K., Enfedaque P., Canadas J., and Lizcano L. A 3D videoconferencing system with 2D backwards compatibility. In *Proceedings of IEEE 3DTV Conference*. IEEE Computer Society, Kos Island, Greece, 2007. 161

[LP05] Landabaso J.L. and Pardàs M. Foreground regions extraction and characterization towards real-time object tracking. In *Proceedings of Multimodal Interaction and Related Machine Learning Algorithms*, Lecture Notes in Computer Science. Springer, 2005. 30, 90, 96, 112, 117, 162

[LP06] Landabaso J.L. and Pardàs M. Cooperative background modelling using multiple cameras towards human detection in smart-rooms (invited paper). In *Proceedings of European Signal Processing Conference*. 2006. 30, 112, 161, 162

[LP07a] Landabaso J.L. and Pardàs M. Shape from Inconsistent Silhouette. *submitted to Journal of Computer Vision and Image Understanding*, 2007. 109, 162

[LP07b] Landabaso J.L. and Pardàs M. A unified framework for consistent 2D/3D foreground object detection. *submitted to IEEE Transactions on Circuits and Systems for Video Technology*, 2007. 162

[LPC06] Landabaso J.L., Pardàs M., and Casas J. Reconstruction of 3D shapes considering inconsistent 2D silhouettes. In *Proceedings of International Conference on Image Processing*. IEEE Computer Society, 2006. 109, 127, 162

[LPX04] Landabaso J.L., Pardàs M., and Xu L.Q. Robust tracking and object classification towards automated video surveillance. In *International Conference on Image Analysis and Recognition*, volume Part II of *Lecture Notes in Computer Science*, pages 463 – 470. Springer, Porto, Portugal, October 2004. 90, 161

[LPX05a] Landabaso J.L., Pardàs M., and Xu L.Q. Hierarchical representation of scenes using activity information. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*. IEEE Computer Society, Philadelphia, PA, USA, March 2005. 161

[LPX05b] Landabaso J.L., Pardàs M., and Xu L.Q. Shadow removal with blob-based morphological reconstruction for error correction. In *Proceedings of International Conference on Acoustics, Speech and Signal Processing*. IEEE Computer Society, Philadelphia, PA, USA, March 2005. 7, 79, 112, 161

[LV00] Lo B.P.L. and Velastin S.A. Automatic congestion detection system for platforms. In *International Symposium on Intelligent Multimedia, Video and Speech Processing*, pages 158–161. 2000. 10, 15

[MBM01] Matusik W., Buehler C., and McMillan L. Polyhedral Visual Hulls for real-time rendering. In *Proceedings of the 12th Eurographics Workshop on Rendering Techniques*, pages 115–126. Springer-Verlag, London, UK, 2001. ISBN 3-211-83709-4. 27, 30

[MBR+00] Matusik W., Buehler C., Raskar R., Gortler S.J., and McMillan L. Image-based visual hulls. In *Proceedings of International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pages 369–374. ACM Press, New York, NY, USA, 2000. ISBN 1-58113-208-5. 27, 30, 112

[MD02] Mittal A. and Davis L.S. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *Proceedings of European Conference on Computer Vision*, pages 18–36. Springer-Verlag, London, UK, 2002. ISBN 3-540-43745-2. 7, 10, 13, 44, 112

[MJD+00] McKenna S.J., Jabri S., Duric Z., Rosenfeld A., and Wechsler H. Tracking groups of people. *Computer Vision and Image Understanding*, 80(1):42–56, 2000. 7, 10, 13, 42, 44, 76, 77, 112

[MK97] Mclachlan G.J. and Krishnan T. *The EM Algorithm and Extensions.* John Wiley and Sons, 1997. ISBN 978-0-471-12358-3. 62

[MKKJ96] Moezzi S., Katkere A., Kuramura D.Y., and Jain R. Reality modeling and visualization from multiple video sequences. *IEEE Computer Graphics and Applications*, 16(6):58–63, 1996. ISSN 0272-1716. 30, 112

[ML98] Marques F. and Llach J. Tracking of generic objects for video object generation. In *Proceedings of International Conference on Image Processing.* IEEE Computer Society, 1998. 5

[MP00] Mclachlan G. and Peel D. *Finite Mixture Models (Wiley Series in Probability and Statistics).* Wiley-Interscience, October 2000. ISBN 0471006262. URL `http://www.amazon.co.uk/exec/obidos/ASIN/0471006262/citeulike-21`. 62

[MRG99] McKenna S.J., Raja Y., and Gong S. Tracking colour objects using adaptive mixture models. *Image and Vision Computing*, 17(3-4):225–231, 1999. 13, 44, 65

[MTG97] Moezzi S., Tai L.C., and Gerard P. Virtual view generation for 3D digital video. *IEEE MultiMedia*, 4(1):18–26, 1997. ISSN 1070-986X. 30, 112

[Mur99] Murota K. Discrete convex analysis — exposition on conjugacy and duality. *Graph Theory and Combinatorial Biology. Bolyai Mathematical Society*, 7:253–278, 1999. 137

[NH98] Neal R. and Hinton G. A view of the EM algorithm that justifies incremental, sparse, and other variants. In M.I. Jordan, editor, *Learning in Graphical Models.* Kluwer, 1998. 65

[Now91] Nowlan S.J. *Soft competitive adaptation: neural network learning algorithms based on fitting statistical mixtures.* Ph.D. thesis, CS Department, Carnegie Mellon University, Pittsburgh, PA, USA, 1991. 65

[ORP00] Oliver N.M., Rosario B., and Pentland A.P. A Bayesian computer vision system for modeling human interactions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):831–843, 2000. ISSN 0162-8828. 10

[PGV$^+$04] Pollefeys M., Gool L.V., Vergauwen M., Verbiest F., Cornelis K., Tops J., and Koch R. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004. ISSN 0920-5691. 35

[PH77] 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. ISSN 0001-0782. 25

[Pic04] Piccardi M. Background subtraction techniques: a review. In *Proceedings of IEEE SMC International Conference on Systems, Man and Cybernetics*, volume 4, pages 3099–3104. The Hague, The Netherlands, Oct 2004. 7

[PLP02] Perrott A., Lindsay A.T., and Parkes A.P. Real-time multimedia tagging and content-based retrieval for cctv surveillance systems. In *Proceedings of SPIE Internet Multimedia Management Systems III*, pages 40–49. Boston, USA, July 2002. 74

[Pot87] Potmesil M. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40(1):1–29, 1987. ISSN 0734-189X. 30, 112

[PS02] Power W.P. and Schoonees J.A. Understanding background mixture models for foreground segmentation. In *Proceedings of Image and Vision Computing New Zealand*. 2002. 16

[RS97] Rappoport A. and Spitz S. Interactive boolean operations for conceptual design of 3-d solids. In *Proceedings of International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pages 269–278. ACM Press, New York, NY, USA, 1997. ISBN 0-89791-896-7. 27, 112

[SB96] Smith A.R. and Blinn J.F. Blue screen matting. In *Proceedings of International Conference and Exhibition on Computer Graphics and Interactive Techniques*, pages 259–268. ACM Press, New York, NY, USA, 1996. ISBN 0-89791-746-4. 112

[SCMS01] Slabaugh G., Culbertson B., Malzbender T., and Shafer R. A survey of methods for volumetric scene reconstruction from photographs. In *International Workshop on Volume Graphics*. Stony Brook, New York, June 2001. 27, 112

[Sen02] Senior A. Tracking people with probabilistic appearance models. In *Proceedings of Performance Evaluation of Tracking and Surveillance*. Copenhagen, Denmark, May 2002. 74

[SG00a] Salembier P. and Garrido L. Binary partition tree as an efficient representation for image processing, segmentation, and information retrieval. *IEEE Transactions on Image Processing*, 9(4):561–576, 2000. 81, 85, 86

[SG00b] Stauffer C. and Grimson W.E.L. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):747–757, 2000. xxi, 7, 8, 10, 11, 13, 14, 38, 42, 44, 76, 85, 112, 126, 154

[Sta03] Stauffer C. Estimating tracking sources and sinks. In *Proceedings of Computer Vision and Pattern Recognition*, volume 04, page 35. IEEE Computer Society, Los Alamitos, CA, USA, 2003. ISSN 1063-6919. 77, 94

[Ste99] Stein G.P. Tracking from multiple view points: Self-calibration of space and time. In *Proceedings of Computer Vision and Pattern Recognition*, pages 1521–1527. IEEE Computer Society, 1999. 16

[SVZ00] Snow D., Viola P., and Zabih R. Exact voxel occupancy with graph cuts. In *Proceedings of Computer Vision and Pattern Recognition*, pages 345–353. IEEE Computer Society, 2000. 30, 100, 112, 117

[SW05] Steventon A. and Wright S., editors. *Intelligent Spaces - The application of Pervasive ICT*, chapter 14. Springer, 2005. ISBN 1846280028. 90, 161

[SWFS03] Seki M., Wada T., Fujiwara H., and Sumi K. Background subtraction based on cooccurrence of image variations. In *Proceedings of Computer Vision and Pattern Recognition*, pages 65–72. IEEE Computer Society, 2003. 10

[SY99] Schmitt F. and Yemez Y. 3D color object reconstruction from 2D image sequences. In *Proceedings of International Conference on Image Processing*, pages 65–69. IEEE Computer Society, 1999. 25

[Sze93] Szeliski R. Rapid octree construction from image sequences. *Computer Vision, Graphics and Image Processing*, 58(1):23–32, 1993. ISSN 1049-9660. 30, 35, 112

[TA02] Tsaig Y. and Averbuch A. Automatic segmentation of moving objects in video sequences: a region labeling approach. *IEEE Transactions on Circuits and Systems for Video Technology*, 12(7):597–612, 2002. 5

[TD01] Tax D.M.J. and Duin R.P.W. Combining one-class classifiers. In *Proceedings of the second international workshop Multiple Combining Systems*, volume 2096, pages 299–308. 2001. 12

[Tra91] Traven H.G.C. A neural network approach to statistical pattern classification by'semiparametric' estimation of probability density functions. *IEEE Transactions on Neural Networks*, 2(3):366–377, May 1991. ISSN 1045-9227. 65

[WADP97] Wren C.R., Azarbayejani A., Darrell T., and Pentland A.P. Pfinder: Real-time tracking of the human body. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):780–785, 1997. ISSN 0162-8828. 7, 8, 10, 11, 13, 38, 42, 112

[WB03] Welch G. and Bishop G. An introduction to the Kalman filter. Technical Report TR 95-041, Department of Computer Science, University of North Carolina at Chapel Hill, 2003. 92

[Wix00] Wixson L. Detecting salient motion by accumulating directionally-consistent flow. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(8):774–780, 2000. ISSN 0162-8828. 7, 112

[Won01] Wong K.Y.K. *Structure and Motion from Silhouettes*. Ph.D. thesis, Department of Engineering, University of Cambridge, 2001. 109, 117

[XE02] Xu M. and Ellis T. Partial observation vs blind tracking through occlusion. In *Proceedings of British Machine Vision Conference*. 2002. 77, 94

[XL07a] Xu L.Q. and Landabaso J.L. Object detection in images. *European Patent Number EP1683105 B. United States Patent Applcation 20070036432*, February 2007. URL http://www.freepatentsonline.com/20070036432.html. 90, 161

[XL07b] Xu L.Q. and Landabaso J.L. Object tracking within video images. *United States Patent Application 20070092110*, April 2007. URL http://www.freepatentsonline.com/20070092110.html. 90, 161

[XLL04] Xu L.Q., Landabaso J.L., and Lei B. Segmentation and tracking of multiple moving objects for intelligent video analysis. *BT Technology Journal*, 22(3):140–150, July 2004. ISSN 1358-3948 (Print) 1573-1995 (Online). 7, 90, 112, 161

[ZA01] Zhou Q. and Aggarwal J.K. Tracking and classifying moving objects from video. In *Proceedings of Performance Evaluation of Tracking and Surveillance*. Hawaii, 2001. 76, 77, 91, 92