Multi-view Object Segmentation

Abdelaziz Djelouah

▶ **To cite this version:**

Abdelaziz Djelouah. Multi-view Object Segmentation. Other [cs.OH]. Université Grenoble Alpes, 2015. English. NNT : 2015GREAM004 . tel-01148203

HAL Id: tel-01148203
https://theses.hal.science/tel-01148203

Submitted on 4 May 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UNIVERSITÉ DE GRENOBLE

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ DE GRENOBLE

Spécialité : **Mathématiques et informatique**

Arrêté ministérial : 7 août 2006

Présentée par

Abdelaziz Djelouah

Thèse dirigée par **Edmond Boyer**
et codirigée par **Jean-Sébastien Franco**

préparée au sein **INRIA Grenoble, Laboratoire Jean Kuntzmann**
et de **l'école doctorale Mathématiques, Sciences et Technologies de l'Information, Informatique**

Segmentation multi-vues d'objet

Multi-view object segmentation

Thèse soutenue publiquement le **17 Mars 2015**,
devant le jury composé de :

Cremers, Daniel
Professeur, Université technique de Munich, Rapporteur
Hilton, Adrian
Professeur, Université de Surrey, Rapporteur
Drettakis, George
Directeur de Recherche, INRIA, Examinateur
Pérez, Patrick
Distinguished Scientist, Technicolor, Examinateur
Boyer, Edmond
Directeur de Recherche, INRIA, Directeur de thèse
Franco, Jean-Sébastien
Maître de Conférences, Grenoble INP, Co-Directeur de thèse
Le Clerc, François
Senior Researcher, Technicolor, Invité

Résumé

L'utilisation de systèmes multi-caméras est de plus en plus populaire et il y a un intérêt croissant à résoudre les problèmes de vision par ordinateur dans ce contexte particulier. L'objectif étant de ne pas se limiter à l'application des méthodes monoculaires mais de proposer de nouvelles approches intrinsèquement orientées vers les systèmes multi-caméras. Le travail de cette thèse a pour objectif une meilleure compréhension du problème de segmentation multi-vues, pour proposer une nouvelle approche qui tire meilleur parti de la redondance d'information inhérente à l'utilisation de plusieurs points de vue.

La segmentation multi-vues est l'identification de l'objet observé simultanément dans plusieurs caméras et sa séparation de l'arrière-plan. Les approches monoculaires classiques raisonnent sur chaque image de manière indépendante et ne bénéficient pas de la présence de plusieurs points de vue. Une question clé de la segmentation multi-vues réside dans la propagation d'information sur la segmentation entres les images tout en minimisant la complexité et le coût en calcul. Dans ce travail, nous investiguons en premier lieu l'utilisation d'un ensemble épars d'échantillons de points 3D. L'algorithme proposé classe chaque point comme "vide" s'il se projette sur une région du fond et "occupé" s'il se projette sur une région avant-plan dans toutes les vues. Un modèle probabiliste est proposé pour estimer les modèles de couleur de l'avant-plan et de l'arrière-plan, que nous testons sur plusieurs jeux de données de l'état de l'art.

Deux extensions du modèle sont proposées. Dans la première, nous montrons la flexibilité de la méthode proposée en intégrant les mélanges de Gaussiennes comme modèles d'apparence. Cette intégration est possible grâce à l'utilisation de l'inférence variationelle. Dans la seconde, nous montrons que le modèle bayésien basé sur les échantillons 3D peut aussi être utilisé si des mesures de profondeur sont présentes. Les résultats de l'évaluation montrent que les problèmes de robustesse, typiquement causés par les ambigüités couleurs entre fond et forme, peuvent être au moins partiellement résolus en utilisant cette information de profondeur. A noter aussi qu'une approche multi-vues reste meilleure qu'une méthode monoculaire utilisant l'information de profondeur.

Les différents tests montrent aussi les limitations de la méthode basée sur un échantillonnage éparse. Cela a montré la nécessité de proposer un modèle reposant sur une description plus riche de l'apparence dans les images, en particulier en utilisant les superpixels. L'une des contributions de ce travail est une meilleure modélisation des contraintes grâce à un schéma par coupure de graphes liant les régions d'images aux échantillons 3D. Dans le cas statique, les résultats obtenus rivalisent avec ceux de l'état de l'art mais sont obtenus avec beaucoup moins de points de vue. Les résultats dans le cas dynamique montrent l'intérêt de la propagation de l'information de segmentation à travers la géométrie et le mouvement.

Enfin, la dernière partie de cette thèse explore la possibilité d'améliorer le suivi dans les systèmes multi-caméras non calibrés. Un état de l'art sur le suivi monoculaire et multi-caméras est présenté et nous explorons l'utilisation des matrices d'autosimilarité comme moyen de décrire le mouvement et de le comparer entre plusieurs caméras.

Abstract

There has been a growing interest for multi-camera systems and many interesting works have tried to tackle computer vision problems in this particular configuration. The general objective is to propose new multi-view oriented methods instead of applying limited monocular approaches independently for each viewpoint. The work in this thesis is an attempt to have a better understanding of the multi-view object segmentation problem and to propose an alternative approach making maximum use of the available information from different viewpoints.

Multiple view segmentation consists in segmenting objects simultaneously in several views. Classic monocular segmentation approaches reason on a single image and do not benefit from the presence of several viewpoints. A key issue in that respect is to ensure propagation of segmentation information between views while minimizing complexity and computational cost. In this work, we first investigate the idea that examining measurements at the projections of a sparse set of 3D points is sufficient to achieve this goal. The proposed algorithm softly assigns each of these 3D samples to the scene background if it projects on the background region in at least one view, or to the foreground if it projects on foreground region in all views. A complete probabilistic framework is proposed to estimate foreground/background color models and the method is tested on various datasets from state of the art.

Two different extensions of the sparse 3D sampling segmentation framework are proposed in two scenarios. In the first, we show the flexibility of the sparse sampling framework, by using variational inference to integrate Gaussian mixture models as appearance models. In the second scenario, we propose a study of how to incorporate depth measurements in multi-view segmentation. We present a quantitative evaluation, showing that typical color-based segmentation robustness issues due to color-space ambiguity between foreground and background, can be at least partially mitigated by using depth, and that multi-view color depth segmentation also improves over monocular color depth segmentation strategies.

The various tests also showed the limitations of the proposed 3D sparse sampling approach which was the motivation to propose a new method based on a richer description of image regions using superpixels. This model, that expresses more subtle relationships of the problem trough a graph construction linking superpixels and 3D samples, is one of the contributions of this work. In this new framework, time related information is also integrated. With static views, results compete with state of the art methods but they are achieved with significantly fewer viewpoints. Results on videos demonstrate the benefit of segmentation propagation through geometric and temporal cues.

Finally, the last part of the thesis explores the possibilities of tracking in uncalibrated multi-view scenarios. A summary of existing methods in this field is presented, in both mono-camera and multi-camera scenarios. We investigate the potential of using self-similarity matrices to describe and compare motion in the context of multi-view tracking.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Image and video segmentation are among the most studied subjects in computer vision. They represent the first step of many algorithms such as scene analysis, matting, compositing for post-production, image indexing, compression and 3D reconstruction.

During the course of this thesis, we will be mainly interested in foreground/background segmentation, which consists in separating image and video pixels into two layers: the foreground layer, that represents the object of interest - a notion that we will more precisely define in this thesis - and the background layer that is often replaced in compositing.

From original image segmentation works [4, 5] to more recent video segmentation tools [6] used in the image and video industry, a wide variety of methods have been proposed, constantly improving segmentation results by addressing more and more challenging situations.

In this thesis we will study the particular situation of multi-camera segmentation, a situation that frequently arises nowadays. Thanks to the advances in multi-camera calibration [7, 8], it is now possible to obtain good information on camera positions with off-the-shelf tools. The work presented here will primarily investigate the usage of this information to improve segmentation results and make user interaction less necessary.

1.1 Motivation

It is interesting to see that despite the important work done in this domain, green screen based segmentation is still widely used, even in professional setups as shown in Fig. 1.1, with an example from Technicolor post-production services. Not only are green screens still largely used, but user guidance is still heavily present. This situation is often encountered in in-painting problems [9, 10] where the proposed algorithms rely on a precise segmentation often obtained with user-assisted tools. It is true that automatically identifying and segmenting interesting layers for image and video processing can be hard to automate, especially when it is more a matter of artistic consideration, such as applying visual effects to specific image regions. However, for a task as simple as identifying the main characters and segmenting them from the background

layer, one can expect this problem to be efficiently addressed by existing algorithms. It appears that this is not the case even in such simple scenarios. Deciding what is the object of interest remains a hard problem and it is only recently that some methods addressing automatic object segmentation have achieved interesting results [11].

FIGURE 1.1: Compositing example from Technicolor: green screen is still widely used for segmentation.

For the purpose of scene editing, the same scene is often captured from different viewpoints. Thus, using multiple cameras in a studio corresponds to a regular situation. It is also becoming a more frequent situation in every day life, with the lower cost of camera devices. This naturally raises numerous questions about how computer vision algorithms adapt to multiple camera setups. Such questions have already been addressed in various domains, such as tracking [12, 13], shape and motion analysis [14] and action recognition [15–17].

For motion and shape analysis, state of the art algorithms rely on binary silhouettes of the foreground object to perform reconstruction. Integrating the obtained models into virtual scenes to achieve photo-realistic rendering is among the main objectives. These new approaches represent an interesting alternative to classic reconstruction and compositing techniques, and the documentary *The destiny of Rome* [1] is a good example of their potential in real productions. In this case, the working datasets often consist of videos captured using a large number of cameras. Relying on user guided tools to segment foreground characters in the context of multi-view capture quickly becomes infeasible: manual preprocessing time is too important in this case. Another consideration is the necessity of a quick rendering of the captured scene, allowing result checking and onset decision of new captures with possible new configurations. Currently state of the art methods require well controlled environments with known background models

[1] http://www.imdb.com/title/tt1722310/

(mostly green and blue screens). An example of images from multi-camera capture platform and reconstruction results is shown in Fig. 1.2.

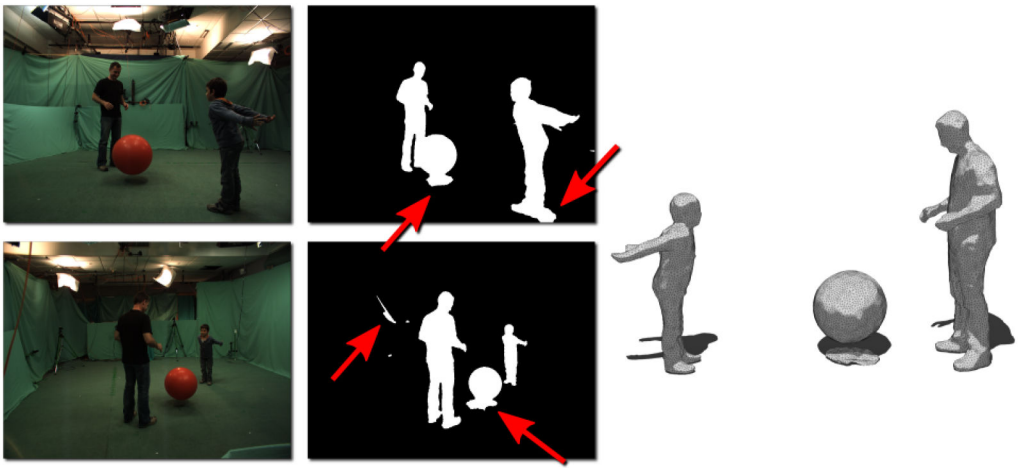


FIGURE 1.2: In this reconstruction example from [14], we can see the segmentation errors with background subtraction techniques. Currently these errors are handled through the 3D reconstruction and mesh tracking [18].

Currently, 3D reconstruction from silhouettes relies on the redundancy of information in multi-camera setups to limit the impact of segmentation errors on the reconstructed visual hull. This is the case for all segmentation errors in Fig. 1.2 except the shadow of the ball. Here, the shadow is consistently observed in all the views, impacting the 3D reconstruction. Using a mesh tracking approach [18], it is possible to correctly recover from this error. Still, low segmentation quality will necessarily affect the reconstruction and if the results obtained in a controlled environment are not perfect, segmentation in general environments is still a challenge to address. Figure. 1.3 shows, for instance, a scenario taken from the work of Guillemaut and Hilton [19] that seems simple at first but turns out to be very challenging, where none of he standard automatic monocular segmentation techniques give satisfactory results. The tested methods are based on mixture of color Gaussians (first result), difference keying with a background image (second result) and finally segmentation using background cut [20], using a combination of global and local appearance models.

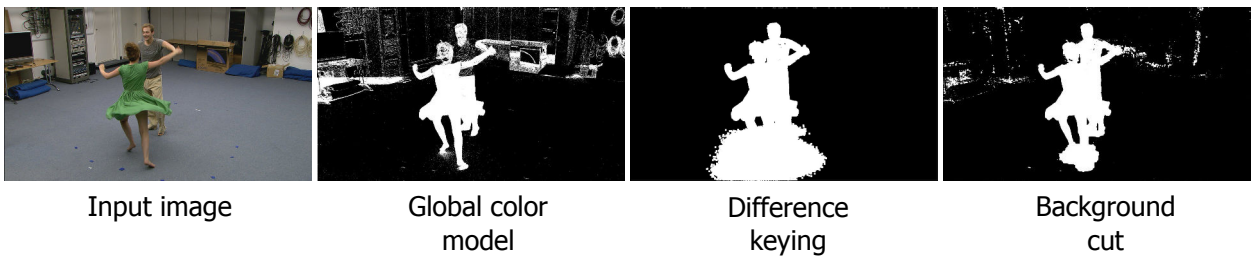


FIGURE 1.3: With this dataset, Guillemaut and Hilton [19] show that automatic monocular segmentation methods are not able to achieve good segmentation results, even in situations that may seem simple.

If 3D reconstruction is an important objective of multi-view setups, it is not the only one. As

previously mentioned, segmentation is a preliminary step for many image post-processing applications like inpainting and compositing. In this case, it is not clear that 3D reconstruction is necessary or even helpful. Especially when dealing with a limited number of viewpoints (3 or 4).

Multi-view setups have their own specificities of which segmentation methods should take advantage. In these configurations, cameras are synchronized and calibration information is available. Some works have targeted multi-view object segmentation either indirectly through 3D reconstruction or directly, trying to express the geometric constraints with epipolar geometry and depth estimation.

As we will show in the state of the art chapter, an exhaustive review of the existing methods reveals that despite the advances made and their interesting results, all the algorithms struggle to express the constraints associated with the multi-view aspect of the problem. Using 3D reconstruction is the most common approach. By solving the dense shape estimation, segmentation in the images is obtained as a byproduct of the main task by re-projecting the 3D model. Another "classic" approach is to resort to epipolar geometry to express the links between the views. In this case there is no dense reconstruction, but to the cost of introducing a point to line relationship that does not simplify the problem. Finally, a last category of approaches assumes a large number of cameras and relies on estimated depth maps from stereo pairs to link the segmentations. Good results can be achieved following this direction, but we are not anymore solving the general case.

The work in this thesis is an attempt to have a better understanding of the multi-view object segmentation problem: what is the foreground object? How the information from different viewpoints can be used? Is a multi-view approach more efficient than monocular segmentation methods? The main objective was to propose an alternative approach to the problem of multi-view segmentation. This new approach should avoid dense 3D reconstruction and complex constraints from epipolar geometry. It should address the general case and achieve results as good as possible even with a limited number of viewpoints. The work in this thesis has naturally led to the possibility of using movement to address video segmentation in the uncalibrated case. In the last part of this thesis, we started investigating this direction.

1.2 Thesis outline

The structure of the dissertation is illustrated in Fig. 1.4. First, a review of state of the art in segmentation is presented (chapter 2). This reviews includes work in image segmentation, co-segmentation and multi-view segmentation. This chapter is important to understand the motivation behind the work in this thesis. In chapter 3, we present a new method to extract multiple segmentations of an object viewed by multiple cameras. It is based on a sparse sampling of the 3D space and a complete probabilistic framework is proposed to estimate foreground and background color models. Chapter 4 presents an extension of the proposed solution to handle different color models and other modalities such as depth information. The results obtained in this chapter are the main motivation behind the work of chapter 5. Indeed, not all the constraints

of the multi-view segmentation problem are expressed in the sparse sampling framework. To address this issue we propose a multi-view graph cut method that is able to express more subtle relationships of the problem. The proposed solution naturally extends to segmentation of multi-view video sequences. With the extension to videos, the question of using motion as a supplementary source of information naturally arises. In chapter 6, we explore this direction as a way to identify and match different tracking hypotheses in a multi-view scenario.

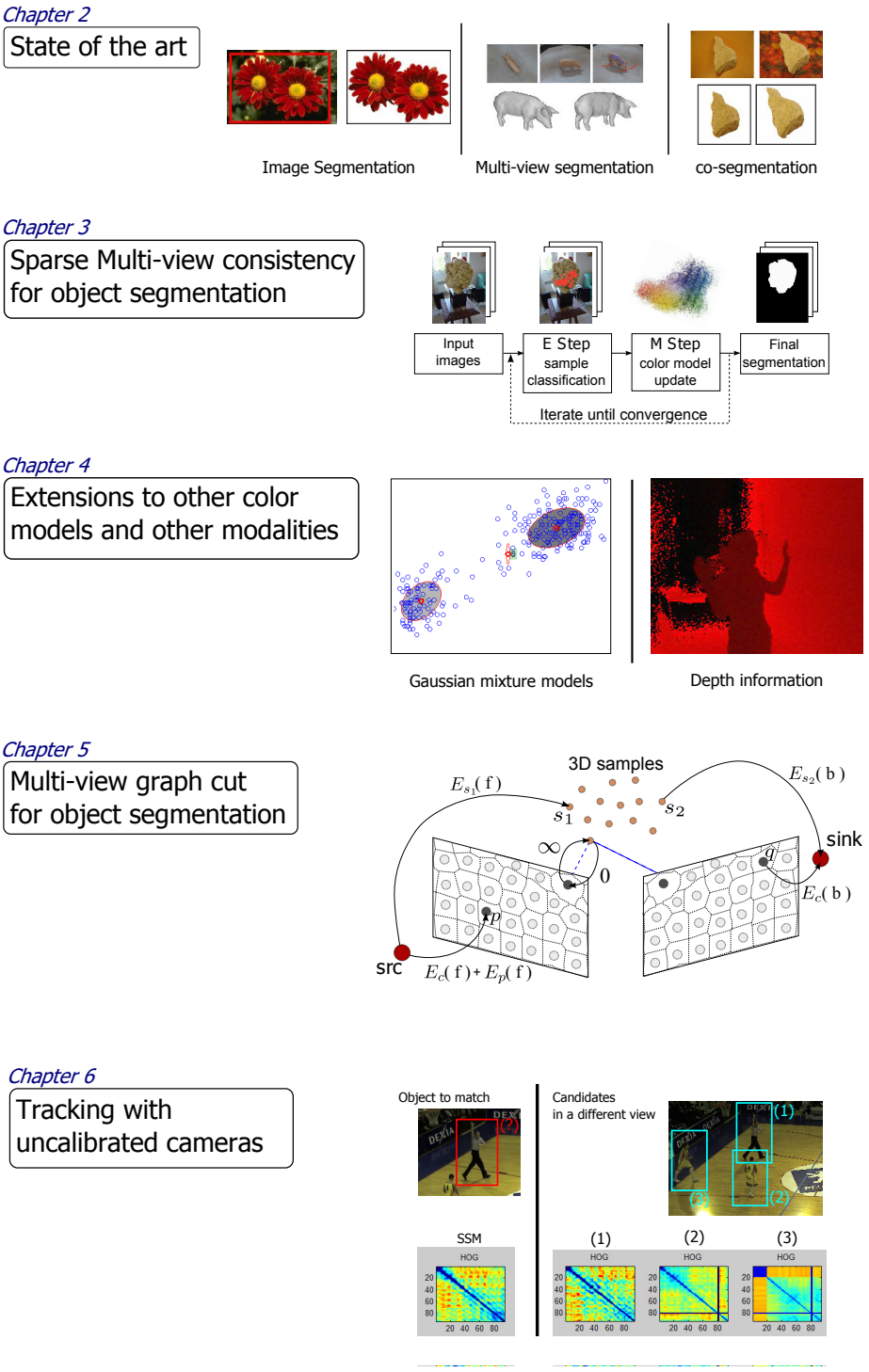


FIGURE 1.4: Thesis outline.

A more detailed description of the chapters follows

- In Chapter 2, a review of state of the art methods in image segmentation is presented. Given the subject of this thesis, we focus on binary segmentation methods, but multi-class labeling algorithms are also discussed, especially in the monocular situation. In this case, the review is fairly short compared to the amount of existing works, presenting only the main directions for binary and multi-class segmentation. Segmentation using multiple images is then addressed, with cosegmentation methods where the main objective is to identify the object commonly present in two or more images. Similarity of appearance is the key assumption for cosegmentation approaches. Finally, a detailed review of existing work in multi-view object segmentation is presented. Using constraints from the multi-view geometry is the driving idea and the existing methods generally follow two main trends: segmentation from dense 3D reconstruction and segmentation using epipolar geometry. This chapter presents the main drawbacks of existing methods that we wish to address in this thesis.

- Chapter 3 presents a new method to extract multiple segmentations of an object viewed by multiple cameras. The main drawbacks of previous methods are avoided. There is no dense 3D reconstruction of the object and no complex constraints from epipolar geometry. The method is designed to address the most general case and it is not limited to short-baseline scenarios. A key difficulty in designing a multi-view segmentation algorithm is in how to enforce this inter-view consistency without sacrificing the simplicity of the approach. To this end, we propose to use sparse 3D sampling of the space as a way to enforce geometric consistency between segmentations in different views. A complete probabilistic framework is proposed to estimate foreground/background color models. The method is tested on various datasets from state of the art.

- In Chapter 4, we explore the potential of the sparse 3D sampling segmentation framework. The idea here is to have a better understanding of the limitations to address. We want to find out if the multi-view segmentation framework is limited by the quantity of information it is provided or if the model proposed so far is missing some aspects of the problems. Two different extensions of the sparse 3D sampling segmentation framework are proposed in two scenarios. In the first, we show how, using variational inference, Gaussian mixture models can be integrated in the framework. In the second scenario, we propose a study of how to incorporate depth measurements in multi-view segmentation. We present a quantitative evaluation, showing that typical color-based segmentation robustness issues due to color-space ambiguity between foreground and background, can be at least partially mitigated by using depth, and that multi-view color depth segmentation also improves over monocular color depth segmentation strategies.

- Chapter 5 presents a new approach that propagates segmentation coherence information in both space and time. The work in this chapter is motivated by the results obtained using different color models and modalities. Results of the sparse sampling segmentation method suggest that using simple color only appearance models is not sufficient to

distinguish between foreground and background in complex scenarios. Despite the geometric consistency expressed in the n-tuple model, the method is still not fully taking advantage of the multi-view setup. In particular, not all of the multi-view segmentation constraints are expressed with the sparse sampling framework. To address this issue, we propose a method based on a richer description of image regions using superpixels and more subtle relationships of the problem are expressed with the 3D samples. In the new framework, time related information is also integrated. With static views, results compete with state of the art methods but they are achieved with significantly fewer viewpoints. Since the proposed method also addresses video segmentation, a review of existing methods is proposed and in the experimental section, results on videos demonstrate the benefit of segmentation propagation through temporal cues.

- Chapter 6 explores the possibilities of tracking in uncalibrated multi-view scenarios. The results obtained in multi-view segmentation suggests that it is also possible to achieve interesting results in segmentation and tracking using only information from movement. A summary of existing methods in this field is presented, in both mono-camera and multi-camera scenarios. This chapter explores the potential of self-similarity matrices to solve this problem in the context of multi-view tracking. An energy minimization framework is proposed to match and validate tracking hypotheses. Preliminary tests on multi-view datasets show good results, indicating that a frequency analysis of the movement is a promising direction in the case of uncalibrated multi-view tracking.

1.3 Main contributions

The contributions of this thesis are consistent with the logical flow of the chapters. The first contribution is the sparse sampling framework for multi-view object segmentation. Its objective is to address the segmentation problem while avoiding the drawbacks of standard approaches. This framework can be applied in different contexts (different appearance models and modalities). The results we obtained motivated the second contribution where an energy minimization approach is used to model more complex relationships of the multi-view segmentation problem. Finally, because of the natural extension of the graph cut framework to videos, we investigate the possibility of using motion as a supplementary source of information in multi-view video sequences.

A more detailed description of the contributions is presented here:

- A framework for multi-view segmentation is proposed based on sparse 3D sampling of the space. The main contribution at this level is to show the possibility to solve the multi-view segmentation problem without using dense 3D reconstruction or epipolar geometry. The segmentation algorithm relies on a generative model for n-tuples. A color n-tuple is a set of pixel colors associated to the n projections of a 3D point (a 3D sample). Multi-view consistency is implicit in this color association. A generative model for the n-tuple is proposed to associate their state, ”empty” or ”occupied”, with their color information using

foreground/background color models. From a technical point of view, model parameters are estimated using a MAP approach, promoting background models explaining known background pixels. The estimation algorithm is derived in the most general terms, then solved for different appearance models and extended to handle other modalities.

- Thanks to the various tests on the sparse 3D sampling framework, a better understanding of the multi-view segmentation problem is achieved. As a result, a more complex model is proposed where 3D samples play a key role. In particular, classic constraints from image segmentation are now included as well as more subtle constraints at the geometric level. At the image level, neighborhood and texture similarity play now an important role in the iterative process. At the geometric level, the lightweight cost associated with sparse 3D samples allows to use them in modeling subtle relationships of the multi-view segmentation problem. Following these ideas, an energy minimization approach is proposed where constraints are expressed using a variety of unary and binary energy terms. The resulting energy is still sub-modular and an s-t graph can be build, where the min-cut will provide the solution to energy minimization problem. The framework naturally extends to temporal domain demonstrating the advantages of using a multi-view approach for video segmentation.

- A preliminary step towards multi-view tracking is proposed, through the usage of self-similarity matrices as a way to match different tracking hypotheses in different views. Contribution at this level is to show the interest and the possibility to use only motion for tracking, making it possible to address challenging situations where camera calibration is not present. Originally proposed for frequency analysis, then used in multi-view video synchronization and action recognition, self-similarity matrices are used in this thesis in an energy minimization approach to address hypotheses validation in uncalibrated tracking scenarios.

1.4 Publications

- **Sparse Multi-View Consistency for Object Segmentation** Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, François Le Clerc, Patrick Pérez. Accepted at IEEE Pattern Analysis and Machine Intelligence.

- **Segmentation Multi-vues par Coupure de Graphes** Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, François Le Clerc, Patrick Pérez. RFIA 2014 - Reconnaissance de Formes et Intelligence Artificielle, 2014.

- **Multi-View Object Segmentation in Space and Time** Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, Patrick Pérez. ICCV 2013 - IEEE International Conference on Computer Vision, 2013.

- **Modélisation Probabiliste pour la Segmentation Multi-vues** Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, Patrick Pérez. ORASIS 2013 - Congrès des jeunes chercheurs en vision par ordinateur, 2013.

- **N-Tuple Color Segmentation for Multi-View Silhouette Extraction** Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, François Le Clerc, Patrick Pérez. ECCV 2012 - 12th European Conference on Computer Vision, 2012.

Chapter 2

State of the art - Multi-view segmentation

In recent years, configurations including multiple cameras have gained interest and many research works have targeted reconstruction [21], tracking [22], motion analysis [23] and action recognition [24]. For all these applications, the identification and extraction of foreground objects is a key element. Usually, standard monocular methods are used, with manual interaction or a well-controlled capture environment. In this chapter we present a review of various segmentation techniques, from seminal monocular methods to more recent research work addressing the multiple views or multiple images segmentation. The work presented in this thesis is mainly concerned with the two-class foreground/background segmentation. Consequently, we focus more in this chapter on methods addressing binary segmentation in the images. State of the art in multi-class segmentation is also mentioned, as it is closely related to binary segmentation where it can be used as a preprocessing step [25].

In monocular scenarios, the object to segment is not clearly defined and monocular segmentation methods rely on a user interaction [3, 26, 27] to have some prior information. The objective is usually to separate the image into foreground/background regions using active contours, that will align with image strong edges [26, 28], or using a Markov Random field [3, 27]. Other approaches try to identify similar regions in the image [29–32] with the idea of obtaining a coherent over-segmentation into different objects or object parts.

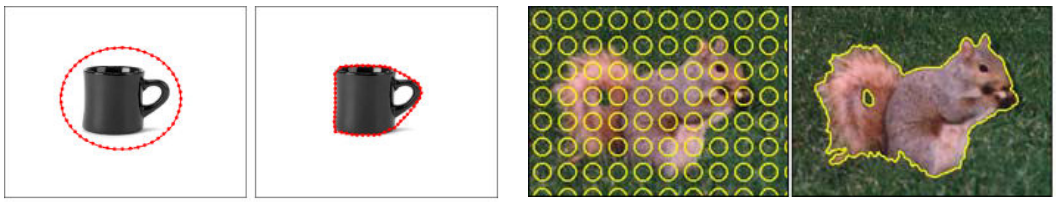
In multi-camera setups, monocular segmentation techniques are still widely used. To avoid the burden of manual segmentation on multi-view video sequences (it is not rare to see several minutes sequences with more than 16 cameras), research teams resort to automatic background subtraction techniques. However these methods are generally limited to well controlled environments where segmentation results can be of satisfying quality. Given this particular context, it is clear that the problem of multi-view segmentation offers great potential to make the best use of its particular constraints. Zeng *et al.* [33] first proposed a method with the clear objective of multi-view object segmentation. Many works [1, 34–37] since followed this first attempt to solve the problem, with the common objective of expressing segmentation constraints from the

multi-view geometry. This state of the art in multi-view segmentation is reviewed in detail in section 2.3.

More recently, the idea of using appearance cues from multiple images to perform segmentation has emerged with the seminal work of Rother *et al.* [38]. This category of works is driven by the idea that the presence of the same object in different images should be helpful in the segmentation task. Originally based on a strong assumption of color similarity, this subject has seen many developments [39–41] tackling more complex scenarios with a wide variety of approaches, which are detailed in section 2.2. Still this category is intrinsically restricted to appearance-based cues or very limited shape similarity constraints that do not correspond to the multi-view segmentation problem, as we show through the comparative studies in this thesis.

2.1 Monocular segmentation

Many approaches exist to monocular object segmentation, which is one of the most studied problems of computer vision. With the earliest works dating back to 1970 [4, 5, 42], the state of the art in this field is very rich. Consequently, drawing a complete review of monocular segmentation methods is beyond the scope of this section. Instead we will describe the main directions and some of the most recent developments.



(a) Segmentation using Snakes [26]

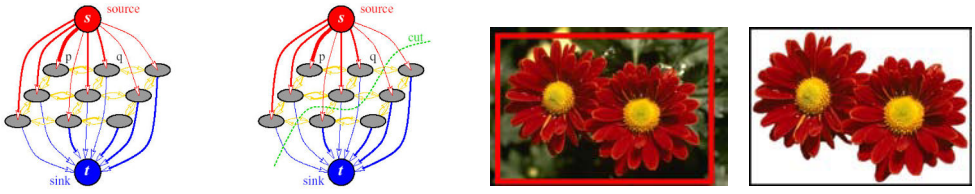
(b) Segmentation using level sets[43]

FIGURE 2.1: Example of active contour segmentation methods: (a) Contour is initialized outside the object to segment. We can see the difficulty for *Snakes* methods [26] to handle topological changes. (b) On the other hand, level set [28, 43] achieve better results with the implicit representation of the curve.

The first category of work, active contours, relies on the location of boundaries in the image to identify the object to segment. The *Snakes* method approach proposed by Kass *et al.* [26] is an energy-minimizing spline curve that evolves towards strong edges. It usually relies on a discrete energy that consists in two terms, called internal spline energy and image-based energy. The internal spline energy consists in two terms. The first term enforces membrane like behavior, increasing the total energy as the contour stretches. The second makes the snake behave like a thin-plate, increasing internal energy as the contour develops more curves. The image-based energy term of snakes, generally based on gradient, attracts the segmentation toward image contours with high image gradient. Because of the internal energy terms, regular snakes are biased toward shrinking and thus, the user input contour is roughly drawn outside the object to segment. It is also very difficult for snakes to adapt to topological changes, as introducing more curves increases the total energy. Figure 2.1(a) shows an example of user initialization

(outside the object to segment) and the resulting segmentation, where we see the difficulty to adapt topological changes. Finally, snakes are sensitive to he local minimum problem because of the image-based term based on gradient that can attract segmentation to a strong edge that is not one of the objects. *Intelligent scissors* methods [44] try to take maximum advantage of this user interaction by learning an edge profile from user input and promote similar boundaries in the following steps. As for snakes, the main limitation of *intelligent scissors* is the difficulty to handle topology changes as the curve evolves [45], which may require reparametrization with drastic shape changes. In the JetStream segmentation method [46], a probabilistic framework is proposed to take into account constraints based on curvature, corners and parallelism. Based on sequential Monte-Carlo techniques, this method is able to handle topological changes by detecting corners in the image and accepting gradient direction changes at these points. Another approach is to use level set [28] where the zero crossings of a characteristic function define the curve. Taking an Eulerian approach eliminates the need to parametrize the tracked curve and makes it very easy to follow topological changes (Fig. 2.1(b)). To avoid local minima, the method can use energy terms related to image region statistics [43] (e.g., color, texture), promoting a segmentation that results in statistically consistent inner and outer regions.

Originally optimized with gradient descent techniques, energy minimization problems in segmentation are essentially solved using Markov Random Field (MRF) optimization algorithms. Boykov and Jolly [47] use a polynomial time algorithm [48] to estimate binary segmentation in the image (Fig. 2.2(a)). The problem is formulated as the estimation of the max-flow/min-cut in a source/sink graph (S-T graph). the capacities of the edges depend on color models for edges linking with terminal nodes and appearance similarity between neighbor pixel nodes. One of the most famous extension of this work is the *GrabCut* [3] algorithm where user interaction is used to specify a bounding box for the object (see Fig. 2.2(b)), then the method iterates between estimating binary segmentation and updating color models. Many priors can be expressed in such graph constructions like connectivity [49] or shape [50].



(a) Energy minimization through graphCut [47] (b) Example of Grabcut result [3]

FIGURE 2.2: GraphCut based segmentation: (a) Boykov and Jolly [47] apply the graphcut method for energy optimization in image segmentation. The estimated cut will label pixels according to the side they belong to (source/sink). (b) In GrabCut [3], the method is then extended with an iterative scheme alternating between image segmentation and color model estimation.

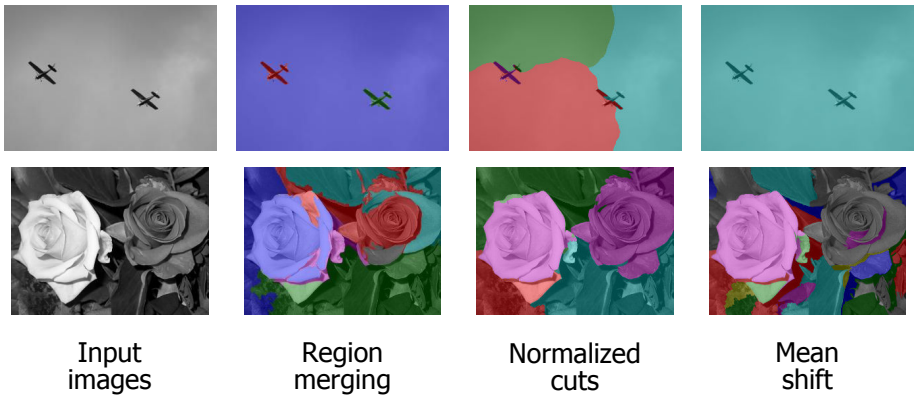
Using quadratic energy terms, segmentation can be expressed as an energy minimization problem over continuous random fields $[0, 1]$ and solved using linear solvers. Works in this direction include [51], [52] and [53].

Except some special cases which can be reduced to a multi-terminal min-cut problem, multiple label energies optimization is not as straightforward as in the binary case. In the $\alpha\beta$-swap

algorithm [54], the idea is to perform a local search on multi-label energies by searching over all the 2^k possible pairs of labels (with k the number of labels). For each pair of possibilities, the energy is optimized using a graphcut. The α-expansion [55] algorithm is another method of similar nature, where at each stage, a node is offered the possibility to stay with the current label or to switch to a new label α. It is also interesting to point to more recent works making use of convex-relaxations to obtain figure/ground segmentation [56] or image region multi-labeling [57].

Another category of monocular segmentation methods is based on split and merge operations to obtain coherent regions segmentation. Vicent and Soille [32] present a method to compute watershed in gray scale images. In [31] Arbelaez *et al.* use maximum oriented energy as a basis for their oriented watershed transform. Felzenszwalb *et al.* [58] develop a graph-based merging algorithm using *relative dissimilarity* to decide which regions should be merged and Alpert *et al.* [30] present a probabilistic merging algorithm based on gray-level and texture similarity. Some results of different clustering algorithm are presented in Fig. 2.3.

(a) Watershed based segmentation [31]



(b) Examples of image mode finding approaches from [30]

FIGURE 2.3: Splitting and merging segmentation techniques: (a) The original work [32] was designed for gray scale images, but this result is from a more recent work [31] which combines contour detection and region segmentation based on oriented watershed. (b) This figure from [30] shows results of their approach based on region merging with comparisons with state of the art normalized cuts and mean shift.

Some methods propose to directly reason on pixel features (e.g., color, texture) to identify the clusters of the distribution, expecting that each cluster represents a meaningful segment of the image. Image distributions are usually described using k-mean or a mixture of Gaussians model

[59]. In [60] a segmentation method for multivariate mixed data is proposed and applied to image segmentation. Instead of explicitly estimating the data distribution, mean shift [61] methods find the peaks in the high dimensionality data distribution. The function is approximated from a sparse set of samples by convolving it with a kernel and gradient ascent is used to find the maxima. With multiple random initializations of the process, one can expect to find all the maxima of the distribution. A review of mean shift implementations and applications can be found in [62].

With the idea of splitting the image into coherent groups or regions, Shi and Malik [29] propose a method based on spectral clustering. The image can be thought as a graph where pixels are nodes linked by edges according to the image neighborhood. The edge weights are proportional to pixel similarity and segmentation is obtained by finding the minimal cut that separates pixels into distinctive subgraphs. The normalized cut is proposed to avoid cuts isolating single superpixels, by looking for collection of edges that are relatively weak to all edges inside a subgraph. Finding the optimal normalized cut involves solving a large eigenvalue problem which can be quite slow and Sharon *et al.* [63] adopt a coarse to fine approach to accelerate the computations.

All the methods presented so far rely on user interaction to identify the object of interest or limit themselves to clustering pixels into "coherent" image regions, but some works have also been undertaken to overcome this limitation by taking into account *object saliency* [64]. Different methods exist to estimate this visual saliency based on color, contrast [65] or patch analysis [66]. The main idea is that salient objects consist of coherent regions that usually have an appearance different from their surrounding regions. These methods provide interesting results; however, due to the possible complex and varying lighting condition in real-world sequences, they may not succeed in producing the correct object proposals.

Instead of identifying the visually salient object, Endres *et al.* [67] try to propose a larger variety of object proposals and rank them according to their diversity. As one may expect, this represents a valuable input for a segmentation algorithm, as it is shown in [68].

One last aspect in monocular segmentation that has not been discussed yet is the use of temporal information when working with video sequences. We draw a short list of such methods, but this point is discussed in more details in chapter 5.

Among monocular segmentation techniques used in controlled multi-camera setups, background subtraction is one of the most important. These methods reason at a per-pixel level, assuming a fixed or constant-color background has been observed with little corruption by foreground objects [69]. A number of such techniques also account for temporal changes of the background [70, 71]. The main advantage of these methods is computational efficiency, however the associated assumptions about background are often too strong to deal with general environments.

The huge amount of work done in monocular segmentation represents a solid base on which multiple images and multi-view segmentation methods build their particular specificities and we show in next sections how similarity of appearance and geometric constraints are used to address more specific situations.

2.2 Cosegmentation

It was first coined in the seminal work of Rother *et al.* [38] as the simultaneous binary segmentation of the common parts in an image pair. It is a more recent subject than multi-view segmentation and it corresponds to a different set of hypotheses. The idea here is to leverage the assumed appearance similarity of some of the object parts between images (Fig.2.4 for an example) to improve segmentation results and overcome the typical errors of automatic monocular methods [72]. To this end, a generative model is proposed yielding an MRF energy function that includes a histogram matching cost for the shared foreground objects. However, this term depends on global properties of the segmentation, making the optimization problem not tractable and only an approximate solution is found through iterated graph-cut schemes, namely, a submodular-supermodular procedure [73] and Trust Region Graph Cuts [38]. In the same spirit, Mukherjee *et al.* [74] seek to match the histograms of the matched regions with an objective function comprising MRF terms and a penalty term on the sum of squared differences of foreground region histograms. After linearization and adjustments, the relaxed linear problem is optimally solved with a solution consisting of only $\{0, 1/2, 1\}$ values.

FIGURE 2.4: Example of cosegmentation result from [38]

Instead of penalizing the variation of appearance models between the common objects, Hochbaum and Singh [75] propose to reward the consistency of the two foreground histograms. The objective is to simplify the underlying optimization problem by using an MRF energy where it is more gainful to label two similar pixels as foreground. Similarity is expressed as being part of the same histogram bin and the minimization problem can be solved efficiently by the construction of an s,t graph and solving the max-flow min-cut problem.

Other methods also compare color histograms in different ways to achieve cosegmentation [76] and use additional features such as SIFT and texture descriptors [40]. Chang *et al.* [77] use visual saliency [65] to identify objects segments, defining the notion of co-saliency. Alternatively, discriminative clustering techniques can also be used, as in [78].

Cosegmentation has clear applications in interactive graphics as has been investigated by Batra *et al.* [79], with their interactive segmentation method for related images. The main argument is that unsupervised cosegmentation methods rely on high variability of background regions between the different images, which does not correspond to many situations that naturally arise in practice (Fig. 2.5). In this work, the user is asked to define the foreground, and the common appearance information is used to improve the segmentation. A key element of the proposed approach is the recommendation system for user interaction, highlighting image regions

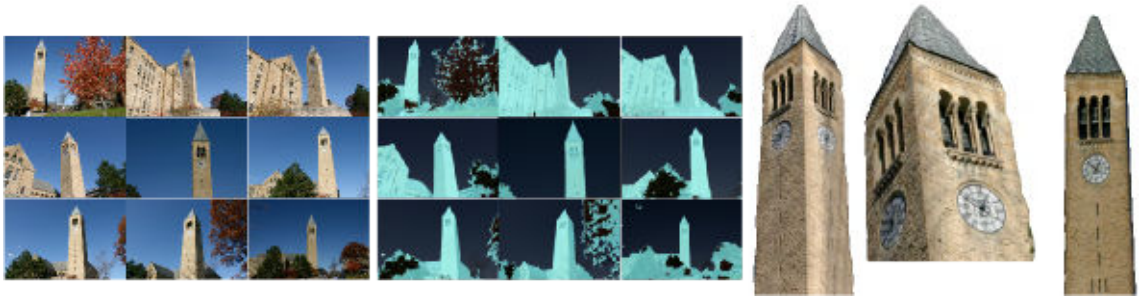


FIGURE 2.5: Cosegmentation and reconstruction example with user interaction [82]

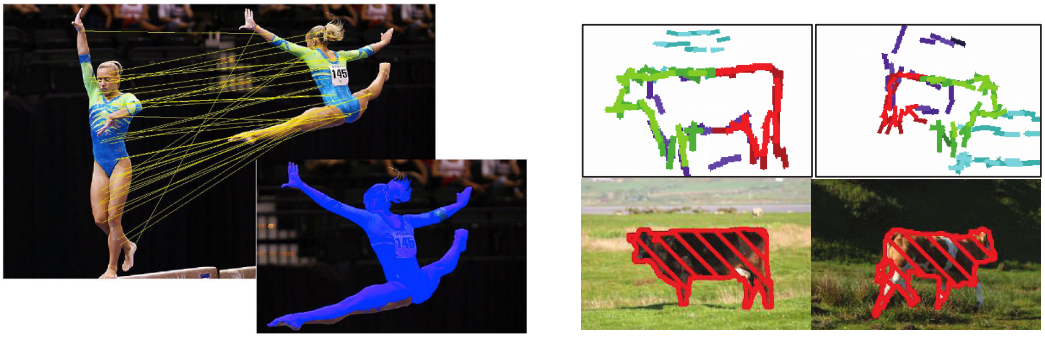
where information input would be most profitable. Other examples of cosegmentation applications include the reconstruction of 3D models of individuals [80] (with user interaction) and the recognition of people wearing the same clothes [81].

As noted by Vicente *et al.* [39], cosegmentation increasingly refers to a diverse set of assumptions and application scenarios, such as user-guided segmentation of large sets [79] or segmentation of object classes rather than a particular instance [83]. In their work [39], Vicente *et al.* focus on object cosegmentation, i.e "things" (such as car) as opposed to "stuff"(such as grass). To achieve a certain level of versatility, the proposed method is based on training a random forest classifier on the appropriate dataset. Reasoning is not anymore performed on pixels or image regions, but on segmentation candidates obtained from [84], and the problem is formalized as a labelling problem in a complete graph. Each image is a node that can have as many labels as segmentation candidates. Pairwise terms encode how similar and how close to ground-truth the to proposals are and an exact A^*-search algorithm is used for inference.

A key assumption of this family of methods is the observation of a common foreground region or object sharing appearance properties, versus a background with higher variability across images. To circumvent this problem, Rubio *et al.* [41] rely on graph matching between different images (see Fig. 2.6(a)) to propagate consistent labelling: the problem is defined in terms of an energy minimization on a two-layered Markov Random Field composed of region and pixel nodes. The proposed energy includes terms enforcing coherent segmentation between different scales (pixel and regions) and between different images through terms defined by matching region graphs for pairs of images. The method iterates between segmentation and appearance model update, where "objectness" [85] plays a key role in the initialization step. Reasoning at different image segmentation layers is also used in [86] where spectral clustering is used to identify shared foreground regions (cosegmentation) and background regions. The proposed graph includes intra- and inter-image affinity edges based on region descriptors.

In a recent work, Faktor *et al.* [88] try to address more complex situations where there seems to be no simple model common to the objects: different colors, poses and shapes. To this end, the authors propose a framework where the foreground segments (image regions) are the ones "well composed" from other images and rely on [89] and [90] to identify non-trivial image parts occurring in different images and infer statistically meaningful affinities.

Another category of methods relies on shape similarity, assuming that segmentations only differ by a rigid body transformation [91]. Similarly, Dai *et al.* [87] propose an unsupervised learning



(a) Cosegmentation with region matching[41] (b) cosegmenation using shape similarity[87]

FIGURE 2.6: Example of different segmentation approaches from state of the art: (a) In [41] region matching is used to perform segmentation. (b) In [87] cosegmentation is performed by the identification of a codebook of deformable shape template shared by the images.

framework for cosegmentation and "cosketch". Here, cosketch refers to the identification of a codebook of deformable shape templates shared by the images (Fig. 2.6(b)). Cosketch and cosegmentation assist each other to achieve better results.

Despite their somewhat different goal, namely image sequence segmentation, Nieuwenhuis *et al.* [92] relate their work to cosegmentation. The conservation of the relative size of object parts is used in a Bayesian framework for multi-region cosegmentation based on convex relaxation techniques from multi-label segmentation.

To sum up, research work in cosegmentation consists in a wide variety of methods that try to leverage cues from different images of the same object to perform segmentation. Essentially based on appearance cues [38, 40, 76–78], the main idea is to identify the visually similar object in the different views. Some of them make also use of geometric cues like [87, 91] where the object shape is matched across images up to a simple transformation. All these methods provide a very interesting view on the segmentation of an object from multiple images based on appearance similarity hypotheses. It means that co-segmentation methods have different application scenarios and they are not designed to handle the multi-view segmentation problem, where drastic variations in the shape and appearance can be observed. In this case, 3D geometry consistency is the main constraint for the segmentation. In the next section we will review in detail multi-view segmentation methods.

2.3 Multi-view segmentation

Multi-view segmentation is the identification and the segmentation of the object simultaneously observed by a set of calibrated cameras. Unlike monocular approaches that essentially rely on user input and cosegmentation methods that use the similarity of appearance between the images, works in multi-view segmentation mainly use geometric constraints.

The first category of works in this topic has as primary objective the 3D reconstruction. Consistent silhouette segmentation is obtained as a byproduct, generally by reprojecting the estimated

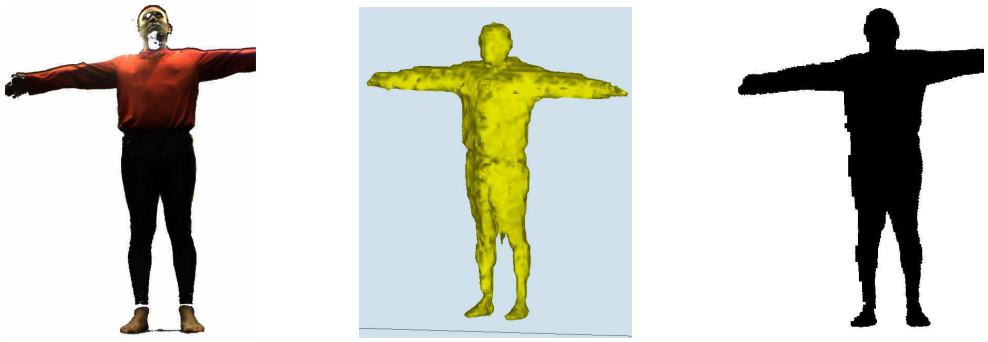


FIGURE 2.7: With an occupancy grid for multi-view fusion [93] some silhouettes errors can be corrected. We note however the aliasing effect of such methods (see text for more details).

3D reconstruction. These approaches rely either on the silhouette consistency of the visual hull [93, 94] or on the photo-consistency of the photo-hull [95]. They assume some initial knowledge about foreground or background appearance, formulating the problem as a purely geometric extraction.

Snow *et al.* [94] try to do a voxel-based 3D reconstruction of the object. To identify foreground voxels (occupied voxels) an energy minimization framework is proposed based on silhouette consistency and solved using graph-cut. In [93] the 3D object is described using an occupancy grid and Franco and Boyer provide a forward sensor model expressing the relationship between the causes, i.e. grid occupancy, and the observations, i.e. silhouettes in the images. The proposed Bayesian framework takes into account voxel states, silhouettes, color information and possible errors on calibration or mis-detections. The joint probability of all problem variables is computed, decomposed and then simplified. Using Bayes' rule, the probability distribution of voxel occupancy state is inferred. Interestingly, the resulting estimation relates directly pixels' color information to grid voxel occupancy. Once the occupancy grid is estimated , segmentation in the images can be corrected by projecting back the 3D model in the images (see Fig. 2.10).

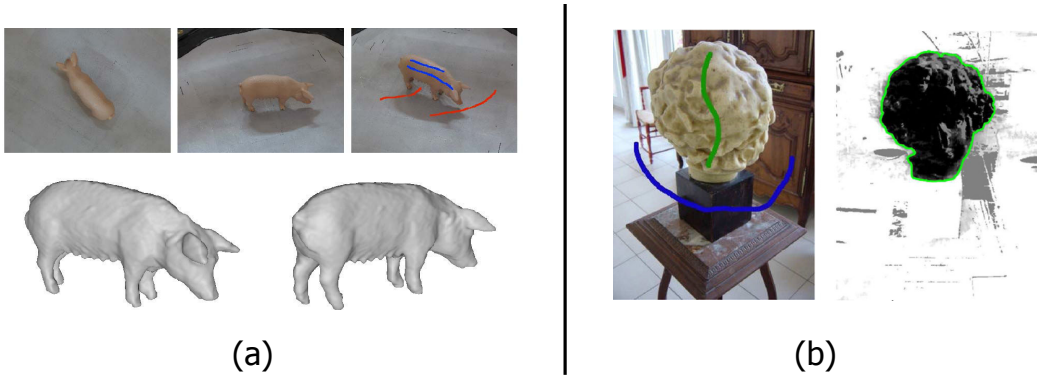


FIGURE 2.8: Variational approaches to the multi-view segmentation problem: (a) shows different input images (with user interaction) and the resulting 3D shape from [2]. (b) User interaction is also used in this example from [96], which extends the previous work addressing more specifically the multi-view segmentation problem.

In the same spirit of multi-view information fusion, Kolev *et al.* [2] propose to reason on a voxel grid to estimate the 3D shape of the objects from user interaction in one view. User scribbles on

foreground and background regions allow to model appearance as a multivariate Gaussian. The probability of occupancy for each grid voxel is given as follows

$$\begin{aligned} P_{obj}(x) &= \prod_{i=1}^{n} P(I_i(x)|x \in obj) \\ P_{back}(x) &= 1 - \prod_{i=1}^{n} (1 - P(I_i(x)|x \in back)) \end{aligned} \tag{2.1}$$

where the probability of a voxel x to be part of the object $P_{obj}(x)$, is estimated as the joint probability of color values $I_i(x)$ (i.e. the colors at the projection of the voxel x in each view i) to be part of the foreground multivariate Gaussian distribution. However, the probability to be background is estimated by reverting foreground evidence with respect to background model, to avoid explicit estimation of voxel visibility. The surface of the object is estimated on this grid by convex relaxation techniques with established algorithmic and convergence properties when color models are known. The segmentations are obtained by projecting back the surface in the images (Fig. 2.8).

Some methods choose to build an explicit dense shape estimate, additionally re-estimating the parameters of color distributions of foreground and background regions, usually leading to complex and computationally intensive pipelines [97–99]. Contrary to the previously mentioned methods, they address both segmentation and reconstruction but still with the idea of obtaining the segmentation by projecting the 3D model in the images.

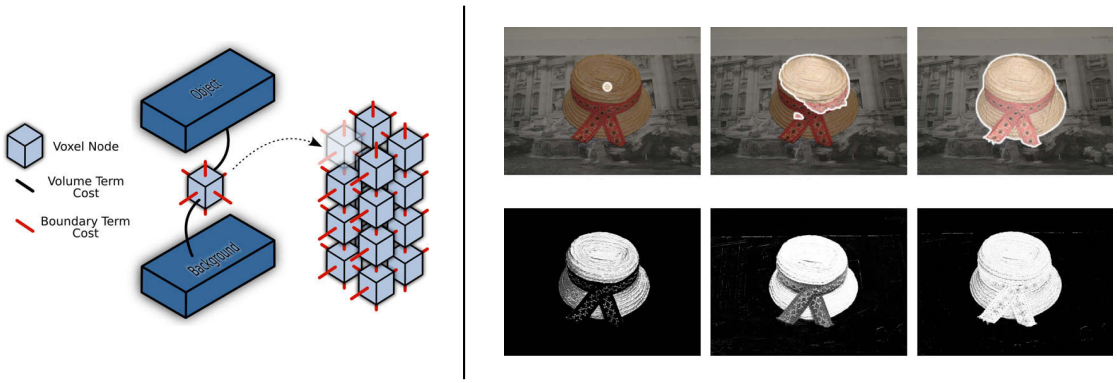


FIGURE 2.9: Graph structure proposed by Campbell *et al.* [97] with results on multi-view dataset. We can see the initialization using the fixation condition, the evolution of segmentation in the first row and object probability map in the second row.

Campbell *et al.* [97] rely on a volumetric graphcut and fixation condition (Fig. 2.9). The main assumption is that the object is entirely seen by each camera and is more or less in the center of the images. This assumption is used to have an initial estimate of foreground color model. Multi-view information fusion is done at voxel level. An energy minimization framework is proposed to estimate object/non-object voxels:

$$\mathrm{E}_{total} = \lambda E_{vol} + (1 - \lambda) E_{boundary} \tag{2.2}$$

The boundary energy term $E_{boundary}$ introduces a higher cost for labelling neighboring voxels differently if they project in image regions of similar appearance. The volume term E_{vol} is the energy term that relates color models to the probability of labelling a voxel x as foreground (part of the object):

$$P_{obj}(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathcal{N}_{obj}(I_i(x))}{\mathcal{N}_{obj}(I_i(x)) + \mathcal{N}_{back}(I_i(x))} \tag{2.3}$$

which is a view normalized "classification probability" that depends on foreground/background appearance models $\mathcal{N}_{obj}$ and $\mathcal{N}_{back}$. This doesn't correspond to a mathematically sound probabilistic model, but rather to a classification cost based on per view color models. The algorithm alternates between this energy minimization step and the update of color models. The same iterative process is proposed in [98] but a background model has to be learned beforehand and fusion is performed following the model from [93]. Finally, Gallego *et al.* [99] use both voxel grids and surface estimation to get the reconstruction and the segmentation. Similar to [36], a graphcut is used to minimize a labelling energy on the voxels where a voxel is more likely to be occupied if it projects on foreground region in "many" views. The object surface is estimated on this grid and silhouettes are obtained by projection in the images.

FIGURE 2.10: Results for multi-view segmentation method [33]: The different views are successively considered and at each stage image regions not corresponding to the current visual hull estimate are segmented as background.

The problem of multi-view foreground segmentation is increasingly addressed as a stand-alone topic and several methods have been proposed to segment an object seen in multiple views. Zeng *et al.* [33] first proposed a method based on classifying superpixel regions. Object silhouettes are identified as the union of a set of superpixel patches. To get the desired set of foreground patches, the silhouette extraction procedure iterates between an estimation of 3D shape as the intersection of visibility cones of the current segmentations and the elimination of patches not coherent with the resulting visual hull (Fig. 2.10). While original, the proposed solution makes deterministic, hard decisions on patch labels and may diverge in case of any classification error.

Reinbacher *et al.* [96] raise interesting questions about 3D dense approaches for multi-view segmentation. In particular, the fact that to segment accurately an object that projects in a region of size $n \times n$ one would require a dense grid of n^3 voxels, which becomes rapidly infeasible for $n \gg 500$. Despite these interesting remarks, they still rely on the same dense framework proposed in [2] and extend it to address more specifically the multi-view segmentation problem. The reasoning on voxel labelling is the same (Eq. (2.1)). The main difference appears at the segmentation step where silhouettes are not estimated using the object surface but by projecting voxel probabilities in the images an performing segmentation independently for each view (Fig. 2.8).

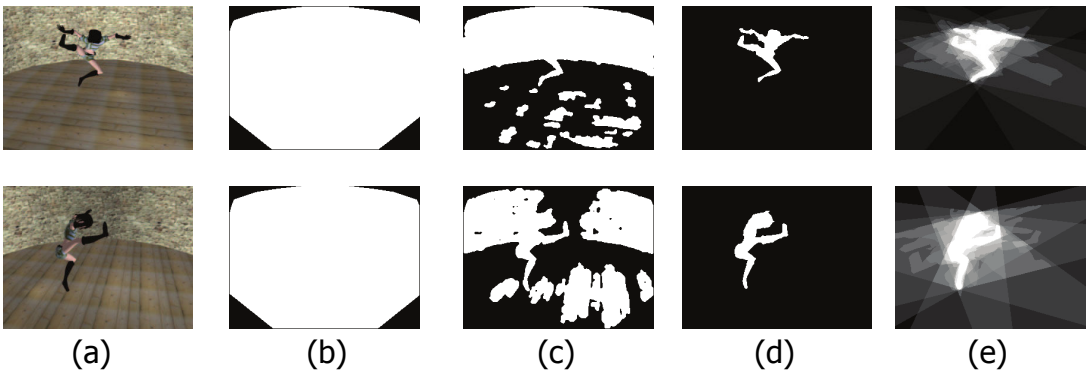


FIGURE 2.11: Lee *et al.* [34, 35] (a) use the calibrated input image to estimate (b) the common visibility volume that is used as initialization for the background regions. The method iterates between color model update and (c) segmentation using appearance and geometric constraints. At convergence (d) Segmentation is refined by including foreground appearance model. (e) We can also see foreground probability related to the silhouette calibration ratio [100].

Still very close to 3D reconstruction, Lee *et al.* [34, 35] focus on probabilistic occupancy along viewing lines. Iteratively, all the images are considered and for each image an MRF energy accounting for geometric and photometric constraints is minimized. The per view MRF energy includes a classical Ising term between pixels but more importantly a data term that depends on the probability of pixels to be foreground or background. Background probability is estimated with color models whereas foreground probability computation is based on the silhouette calibration ratio [100]. It is a 3D consistency measure based on the intersections between the viewing ray at each pixel and the current silhouette viewing cones. The resulting foreground probability is shown in Fig. 2.11. This method does not make use of any dense representation but it is not computationally efficient, due to the calibration ratio which requires to do the segmentation in the views sequentially. The authors mention a running time of several minutes.

Some multi-view segmentation techniques do not rely on a dense reconstruction of the object and try to propagate coherent geometric constraint by other means. In a scenario similar to GrabCut [3], Sorman *et al.* [101] formulate the problem in a graph cut framework where user interaction and segmentation is propagated between adjacent views as 2D shape coherence constraints. However, this supposes a short baseline and smooth transition of the foreground object between the views.

Another approach to the multi-view segmentation problem is to consider epipolar geometry as a way to link segmentation in different views. In [102] the objective is to propagate a user defined trimap to the different views. The trimap consists of background, foreground and unknown regions and the alpha matte is obtained using Levin *et al.* [103] method. To propagate the trimap labels, both foreground and background models are used but the key element is epipolar geometry. To label a given pixel in a new view $i+1$, the method will base its estimation on the most similar pixels, among other things, in the corresponding epipolar band in the already processed view i.

In [37], Campbell *et al.* use both epipolar geometry and depth information as constraints for the problem. The images are first over-segmented into superpixels. Segmentation is formulated

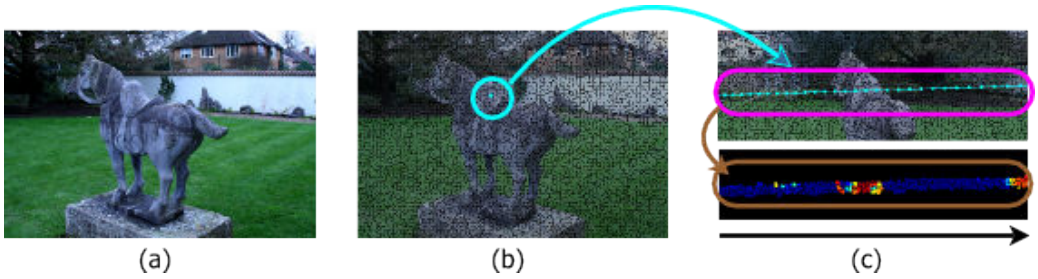


FIGURE 2.12: One way to address the multi-view segmentation problem is to use epipolar geometry. In their work, Campbell *et al.* [37] (a) first segment the image into superpixels then (b) link each superpixel from a given view with the ones on its epipolar line in another view. (c) The number of superpixels to link with is reduced using depth and color information.

as finding superpixel labels that minimize a classic MRF segmentation energy. The data term for this energy depends on the estimated foreground/background appearance models. More interestingly, the pairwise term links superpixels from different views using epipolar geometry and depth estimation. The strength of these links varies according to the similarity between the considered superpixels and allows to reduce some of the combinatorial aspects induced by the point to line match of epipolar geometry (Fig. 2.12). The method relies on the fixation condition to bootstrap the color model of the object and requires reliable stereo correspondences for good results.

Multi-view segmentation is also important in the context of free view point videos [19, 104, 105]. In a first work [104], Guillemaut *et al.* assume the presence of a trimap for each view with a known background model, and propose to jointly solve for matting and reconstruction with application to novel view synthesis. In more recent works [19, 105], they address more challenging situations with an outdoor environment and rely on depth estimation and interest point matching between views to propagate a coherent segmentation.

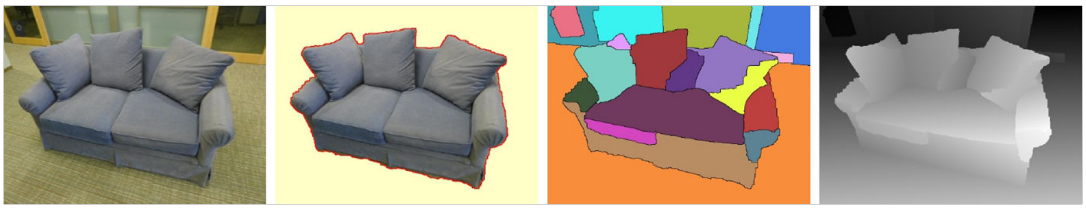


FIGURE 2.13: Results from [1] using planar piecewise reconstruction of the scene. Only planar regions seen by all the cameras are kept and identified as the foreground object.

In a more constrained situation, Kowdle *et al.* [1] present a method based on stereo and piecewise planar reconstruction assuming short baseline viewpoints. A multi-label MRF optimization is proposed to associate pixels to the set of planes. Small planes are coherently grouped together to form bigger planar regions. A binary labelling problem is defined on both the regions and the pixels, where the data term measure the 3D coherence. Since the object of interest is defined as the object simultaneously seen by all the cameras, coherence is defined as the number of views that see a given pixel.

Throughout this review of state of the art in multi-view segmentation, it clearly appears that the main challenge is to find the best way to express geometric constraints to link in a consistent

way segmentation between the different views. Most of the approaches [36, 96, 98] model these geometric constraints by building an explicit, dense shape estimate of the object, mainly through voxel or occupancy grids. The segmentation is obtained by projecting this dense shape in the images. The work of Reinbacher *et al.* [96], in parallel to the one in this thesis, already points the limits of such an approach where any precise image segmentation would require an unreasonable number of voxels, but this idea is not investigated any further. Other methods make use of depth information, or epipolar geometry, with clear limitations. Estimating depth is not reliable and even not possible in many scenarios with a small number of views. Epipolar geometry induces a point to line relationship which is only manageable in scenarios where image segmentation is known in certain views [104] or by limiting the number of links to create by using color or depth information [37]. Again, this implies a similarity of appearance across the views and the possibility to compute depth information. In a different direction, Lee *et al.* [35] address the problem using the silhouette calibration ratio. However, this limits the approach to process views sequentially.

2.4 Conclusion

In this chapter various state of the art methods in image segmentation were presented. The related work is classified into three categories according to the initial assumption. The first category is monocular segmentation where only a single image is available and no particular assumption exists. The corresponding section presented the necessary background with the main ideas and notions that should be kept in mind while working on image segmentation. These methods mainly deal with object segmentation, which is the primary concern of the work presented in this thesis; however, methods addressing pixel clustering or multi-class segmentation were also mentioned. They represent an important part of segmentation and are often used as a preprocessing step in many object segmentation algorithms.

The second category, co-segmentation, assumes that several images of an object exist and reasons on appearance similarity to identify the shared object. Despite the various strategies, cosegmentation methods do not appear to be fully adapted to the multi-view segmentation problem where neither object appearance nor shape similarity between the images can be assumed.

Finally, multi-view segmentation concerns the segmentation of an object simultaneously captured by a set of cameras. Multi-view segmentation methods try to address this issue through approaches based on geometric constraints. However, the detailed review of these segmentation methods revealed the clear struggle to express constraints of the problem. In previous work, this issue was addressed through dense 3D shape representation, complex multi-image relationship using epipolar geometry or depth information. These approaches represent interesting solutions and produce good results in their particular context. However, they may not fit the general case where the color model can be rich and the number of cameras limited. Addressing theses limitations was one of the main ideas that drove the work presented in this thesis.

Chapter 3

Sparse Multi-View Consistency for Object Segmentation

3.1 Introduction

The multi-view segmentation problem can be defined as identifying and segmenting the object(s) simultaneously seen by multiple calibrated cameras. This problem has gained interest in recent years with the objective to leverage multiple views cues toward automatic segmentation. A key issue in that respect and compared to monocular settings is to ensure propagation of segmentation information between views while minimizing complexity and computational cost. In previous work, this problem has essentially been addressed through: dense reconstruction of the object where segmentation is obtained as a byproduct in [2] and [96]; information transfer between the views using either epipolar geometry [19], stereo information [1] or both [37].

In this chapter, we first discuss the notion of foreground object in multi-view segmentation. In the state of the art, the existing methods can rely on a fixation condition [37, 97] or user interaction [2] to define the foreground. Other methods propose less restrictive assumptions [33, 35] that we adopt in this work. These hypotheses lead to a clear definition of the object of interest, and correspond to a limited set of constraints from appearance and geometry that are directly used in our framework.

Let's clarify this point before proceeding to the presentation of the methods developed in this thesis: Multi-view foreground/background segmentation is to identify and segment the object(s) entirely seen by all the views in a multi-camera setup. But how to identify this object? let's consider the scene in figure 3.1. According to our definition, the foreground object is the plant. Humans are able to perform this task because they identify all the objects of the scene and only keep the ones entirely seen in all the images. This reasoning eliminates the table from the foreground regions, despite the fact that some parts of it are seen in all the views.

In multi-view segmentation, the idea is to take advantage of camera calibration information to replicate this reasoning, while keeping the complexity of the solution as low as possible.

This means that we do not want to identify all the objects constituting the scene to produce segmentation. The fixation condition and user interaction can be used but they constitute a strong *a priori* on the scene that we would like to avoid (for example, the fixation condition doesn't correspond to the situation illustrated in Fig. 3.1).

To address this issue, we adopt a similar approach to some state of the art methods in multi-view segmentation [33, 35] that consists in using appearance to propagate information about regions inside a single view. Concretely, if we know the foreground/background state of some elements of the scene, then the objective is to propagate this segmentation to other parts of the scene (in the same image) assuming that similar regions are likely to be parts of the same object.

FIGURE 3.1: The multi-view segmentation problem: Given a set of input images of a scene, we want to identify and segment the object(s) entirely seen by all the cameras. In this case, it is the plant.

These constraints at image level are very similar to those in image segmentation. However, in multi-view segmentation there is another dimension that must be taken into account which is the geometric coherence. The idea is to use calibration information to propagate in a coherent way segmentation information between the views. The primary objective in this chapter is the expression of geometric constraints for the segmentation problem. Instead of using dense reconstruction or epipolar geometry, we propose to use a sparse set of 3D samples to propagate consistent information on foreground and background color models in the different views. We describe a new approach, based on a complete probabilistic framework, that allows us to cast the multi-view segmentation problem as Maximum A Posteriori(MAP) over this sparse set of 3D samples. First presented in general terms in section §3.2, then formally detailed in §3.3, this method avoids altogether a complete dense representation while encoding the specificities of the multi-view segmentation problem. A complete evaluation is performed in section §3.6, including comparative results with state of the art monocular and multi-view methods.

3.2 Proposed approach

To present the proposed approach, we rely in this section on the toy example of Fig. 3.2. Here the scene is captured by 3 cameras. The teddy bear is entirely seen by all the cameras at the same time. If we consider the right-most view, we can see that some parts of the checkerboard are not present in the other cameras (especially the parts far from the teddy bear that don't appear in the closer middle view). Our objective is then to propagate this information, to produce the segmentation that only keeps the object (teddy bear) with an appearance different from that of the background (checkerboard).

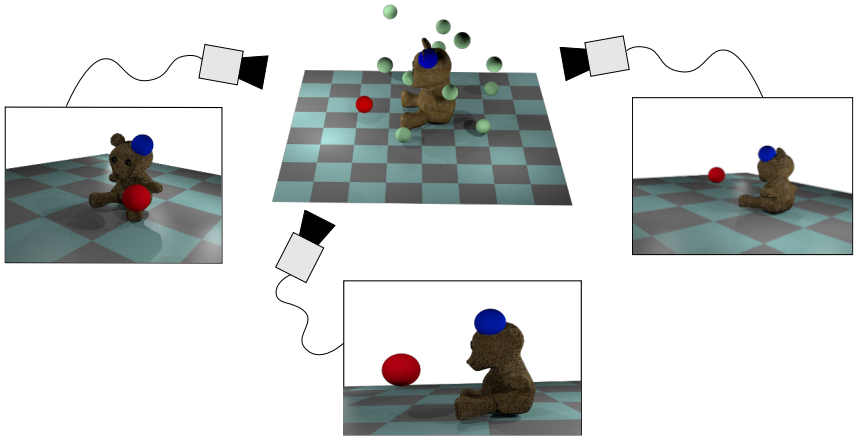


FIGURE 3.2: Principle of multi-view object segmentation using sparse 3D samples: In this synthetic scene, the teddy bear satisfies our definition of *foreground object* and should be the result of our segmentation method. To identify the foreground region, samples (depicted by the spheres) are created in the common visibility volume. A sample is labelled as foreground (blue sphere), if it projects on foreground regions in all the views. In contrast, it is enough for a sample to project to background in one of the views to be labelled as background. This is the case here for the red sphere, classified as background as it projects to background in the middle and rightmost views, even though it projects to foreground in the leftmost view.

In the proposed approach, to avoid using the same constraints as in state of the art methods (e.g. epipolar geometry, dense reconstruction), we focus on a set of sparse 3D samples of the space commonly viewed by the n calibrated cameras and considering only the set of colors at the projections of each sample. The n colors present at pixel projections of a given sample define a color n-tuple, which is the basic unit of information processed by the method. The spatial consistency of the foreground across views is expressed using these n-tuples. Since none of the 3D position related information is used (e.g. visibility, neighborhood), reasoning directly on these n-tuples allows a simpler and clearer framing of the problem. From now on, the terms *sample* and *n-tuple* will be indistinctly used to designate a 3D sample and its corresponding n-tuple, respectively. A generative model for sample labels is defined from the following intuition (Fig. 3.2): If a sample is from the foreground object, then all corresponding tuple colors should simultaneously be predicted from the foreground color model in their respective images. This sample may not be visible in all the views but it will always project on the foreground region in the images. Conversely, if the sample is not from the foreground object, then there exists at least one image where the corresponding sample color should be predicted from the background

color model in that image. This sample, that projects in a background region in one view, may project on background or foreground regions.

Let's consider a sample labelled as background by a view i. It has its corresponding color predicted from the background distribution in this view i. But the other color components from the other views can be background or foreground. This means that the color of this sample for a view j ($j \neq i$) can not be predicted by background or foreground distributions only. However, whether it projects onto foreground or background regions in the other views, this sample will still project inside the image[1]. This means that this color information from view j is not randomly sampled in the color space but follows the statistics of the color distribution in the image. To model this, we introduce the general image distribution for each view that is used to predict color for a sample when it is classified as background by some other view. This generative model relates, in a probabilistically sound way, the state of 3D samples with color models in the images. This a key contribution of this work, as state of the art methods generally use approximations that are not valid in the Bayesian sense. We note that this is in principle equivalent to deciding whether the 3D sample belongs to the visual hull of the object [35].

3.3 Probabilistic Model

Let $\mathcal{S}$ be the selected 3D sample set. The color n-tuple associated to the sample $s \in \mathcal{S}$ is $I^s_{1:n} = (I^s_1, \cdots, I^s_n)$. Following the intuition presented earlier, n-tuple colors should be predicted according to the sample state (foreground or background) and the appearance models. The color n-tuple of a foreground sample (labeled f) is predicted from the shared foreground appearance model Θ^F. On the other hand, a single view i is sufficient to label a sample as background (label b_i). In this case, the corresponding n-tuple color I^s_i is predicted according to the view specific color model Θ^B_i.

This reasoning is illustrated by the graphical model of Fig. 3.3, where each sample's color n-tuple is predicted according to its classification label k_s, and to the parameters Θ of the appearance models. The classification label k_s is in state space $\mathcal{K} = \{\mathrm{f}, \mathrm{b}_1, \cdots, \mathrm{b}_n\}$. The parameters π_k's are the mixing coefficients representing the proportion of samples explained by each hypothesis in $\mathcal{K}$. They act as prior on the classification labels k_s. The proposed model can be viewed as a mixture of foreground-background models on the n-tuples where we try to estimate sample membership and model parameters.

If we note by

- $I = \{I^s_{1:n}\}_{s\in\mathcal{S}}$ the set of image observations,
- $K = \{k_s\}_{s\in\mathcal{S}}$ the sample labels,
- Θ the appearance models,

[1] If this was not the case then this sample is not foreground according to our assumptions and should not be considered anyway

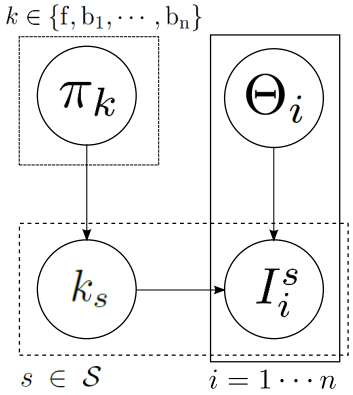


FIGURE 3.3: Graphical model: I_i^s, the color of the projection in the image i of the sample s, relates color models Θ_i according to its labelling k_s. Parameter π_k is the mixture coefficient (label prior).

- $\pi = \{\pi_k\}_{k,\in\mathcal{K}}$ the set of mixing coefficients,

our goal is to find the set of parameters $\Phi = (\Theta, \pi)$ that maximizes the *a posteriori* density given the observations:

$$\Phi = \arg\max_{(\Theta,\pi)} \quad \mathcal{L}(\Theta, \pi | I, K) p(\Theta, \pi), \tag{3.1}$$

where $\mathcal{L}(\Theta, \pi | I, K)$ denotes parameter likelihood. The MAP estimation of the parameters using only the observation is intractable. Therefore, as in other mixture fitting problems, unknown assignment labels K will be marginalized out (through EM) rather than explicitly estimated along with parameters.

Likelihood function. Given variable dependencies defined in our generative model, the likelihood function can be rewritten as follows:

$$\mathcal{L}(\Theta, \pi | I, K) = p(I, K | \Theta, \pi) = \prod_{s \in \mathcal{S}} p(k_s, I_{1:n}^s | \Theta, \pi), \tag{3.2}$$

where for a given sample s, assuming conditional independence of the observations in each view, we have:

$$p(k_s, I_{1:n}^s | \Theta, \pi) = p(k_s | \pi) \prod_{i=1}^{n} p(I_i^s | \Theta_i, k_s). \tag{3.3}$$

Before further developing the proposed probabilistic model, we introduce the notion of region of interest. Given our assumption of common visibility for the foreground, we limit sample classification to the 3D volume seen by all the views. This volume can be automatically computed from the common field of views of the cameras [106]. Its projection in each view, the image region noted R_i^{Int} (see Fig. 3.4), contains all foreground parts. It is the region where the segmentation is to be estimated.

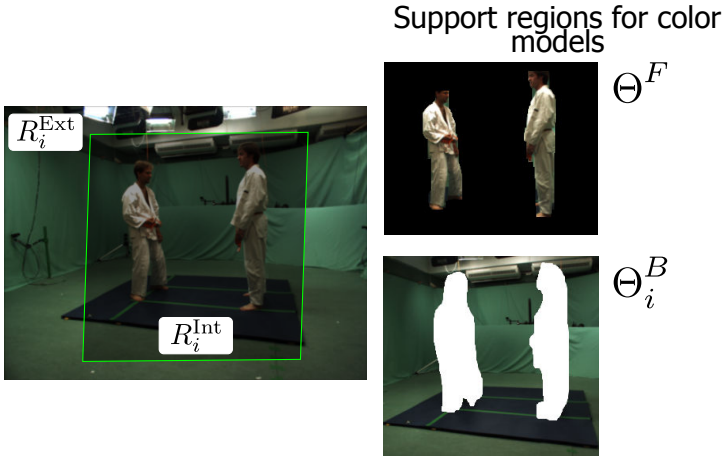


FIGURE 3.4: Support regions for the color models using the assumption that foreground objects are seen in all images. R_i^{Int} is the projection of the common field of view that includes all foreground pixels in image i. Pixels in R_i^{Ext} are then known background pixels. Color model Θ_i^B is to be learned for background pixels inside R_i^{Int} and Θ^F is to be learned for foreground pixels (shared between the views).

We can now exploit the definition of the foreground introduced earlier to develop the probabilistic model of samples classification labels expressed by (3.3).

(A) *A foreground sample projects on foreground regions in all the views.* Using the shared foreground color model Θ^F, this translates to

$$\forall i \in [\![1, n]\!], \quad p(I_i^s|\Theta_i, k_s = \text{f}) = p(I_i^s|\Theta^F). \tag{3.4}$$

The foreground model is defined in general terms. It is a global appearance model for all the views. This does not imply an assumption of inter-view shared appearance. The idea here is to take advantage of shared appearance when it exists, without making it a primary assumption. This point is discussed more in details in the experimental section, including comparisons with results using a per-view model.

(B) *One view is enough to label a sample as background*. For a sample s classified as background for view i (label b_i), the i-th color of the n-tuple should be predicted from the background color model in image i. In the other views j (with $j \neq i$), this sample can project in foreground or background regions. If visibility information was available for the sample s, the color value I_j^s would be predicted using the appropriate model (foreground or background). However, computing visibility of samples would require a dense reconstruction of the scene, which we want to avoid in the first place. In the absence of visibility information, one can still predict an indiscriminate but correct sample color likelihood for all other views, by assuming it is drawn uniformly from the color distribution of the region of interest R^{Int}. This encompasses both background and foreground pixels and can be seen as discounting the exact influence of the membership status of the sample in all other views than i, as long as view i's color provides

the most decisive evidence that the sample belongs to the background:

$$\forall i \in [\![1, n]\!], \begin{cases} p(I_i^s|\Theta_i, k_s = \mathrm{b}_i) = p(I_i^s|\Theta_i^B), \\ p(I_j^s|\Theta_j, k_s = \mathrm{b}_i) = p(I_j^s|\Theta_j^{\text{Int}}), \forall j \neq i. \end{cases} \tag{3.5}$$

Finally, the term $p(k_s|\pi)$ in Eq.3.3 represents the mixture proportion prior:

$$p(k_s|\pi) = \pi_{k_s}. \tag{3.6}$$

Prior from known background pixels. If the regions of interest R_i^{Int} contain all foreground parts, then the complementary regions (noted R_i^{Ext}) are background regions. Thus, an estimate of background appearance distribution is more likely if it also predicts pixels from outer background image regions R_i^{Ext}.

To use this prior in the MAP approach, and enforce similarity between the distribution of background pixels and colors in regions R_i^{Ext}, with respect to our generative model, we would need to create 3D samples that project in this region and thus, have a background label. This defines the following prior over Θ:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{s \in \mathcal{S}_i} p(I_i^s|\Theta_i^B), \tag{3.7}$$

where $\mathcal{S}_i \subset \mathcal{S}$ is the set of such 3D samples relative to view i, making appearance model $\{\Theta_i^B\}$ more likely if it explains known background samples.

We can express the constraint in terms of pixels, instead of 3D samples:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{p \in R_i^{\text{Ext}}} (p(I_i^p|\Theta_i^B))^{t_p}, \tag{3.8}$$

where I_i^p is the color of pixel p in image i and t_p is the number of 3D samples projecting onto this pixel.

Since we do not want to create samples outside the common field of view, we approximate the value of t_p with λ_i, the mean number of samples projecting on a single pixel in R_i^{Int}. The prior on the background color distribution is then the following:

$$p(\Theta) = \prod_{i=1}^{n} \prod_{p \in R_i^{\text{Ext}}} (p(I_i^p|\Theta_i^B))^{\lambda_i}. \tag{3.9}$$

3.4 Estimation Algorithm

The unknown parameters $\Phi = (\Theta, \pi)$ are obtained through MAP estimation:

$$\begin{aligned}\hat{\Phi} &= \arg\max_{(\Theta,\pi)} \mathcal{L}(\Theta, \pi | I, K) p(\Theta, \pi) \\ &= \arg\max_{(\Theta,\pi)} \prod_{s\in\mathcal{S}} \left[\prod_{i=1}^{n} p(I_i^s | \Theta_i, k_s) \right] \pi_{k_s} \cdot \prod_{i=1}^{n} \prod_{p\in R_i^{\text{Ext}}} (p(I_i^p | \Theta_i^B))^{\lambda_i},\end{aligned} \tag{3.10}$$

where the classification labels k_s are treated as latent variables. We use an Expectation-Maximization algorithm that alternates between:

1. E-step: Computing the expectation of the posterior over the classification variables k_s, given the current parameter estimate $\Phi^g = (\Theta, \pi)^g$;
2. M-step: Estimating the new set of parameters $\Phi = (\Theta, \pi)$ maximizing the expected log-posterior.

We build the E- and M-steps using the expectation of the complete-data log likelihood Q, with established convergence properties [59]:

$$Q(\Phi, \Phi^g) = \sum_{K\in\mathcal{K}^n} p(K | I, \Phi^g) \log \mathcal{L}(\Phi | I, K) + \log p(\Phi). \tag{3.11}$$

Expanding each term of the sum, we get:

$$\begin{aligned}Q(\Phi, \Phi^g) = &\sum_{K\in\mathcal{K}^n} \left[\log\Big(\prod_{s\in\mathcal{S}} p(k_s, I_{1:n}^s | \Phi)\Big) \prod_{s'\in\mathcal{S}} p(k_{s'} | I_{1:n}^{s'}, \Phi^g) \right] \\ &+ \sum_{i=1}^{n} \lambda_i \Big(\sum_{p\in R_i^{\text{Ext}}} \log p(I_i^p | \Theta_i^B) \Big).\end{aligned} \tag{3.12}$$

The term

$$A = \sum_{K\in\mathcal{K}^n} \log\Big(\prod_{s\in\mathcal{S}} p(k_s, I_{1:n}^s | \Phi)\Big) \prod_{s'\in\mathcal{S}} p(k_{s'} | I_{1:n}^{s'}, \Phi^g) \tag{3.13}$$

can be simplified following [107]. If we note $N_{\mathcal{S}}$ the number of 3D samples, then the sum $\sum_{K\in\mathcal{K}^n}$ over all possible labellings can be written as the some over label values for samples indexed from 1 to $N_{\mathcal{S}}$:

$$\begin{aligned}A &= \sum_{K\in\mathcal{K}^n} \sum_{s\in\mathcal{S}} \log(p(k_s, I_{1:n}^s | \Phi)) \prod_{s'\in\mathcal{S}} p(k_{s'} | I_{1:n}^{s'}, \Phi^g) \\ &= \sum_{k_1=1}^{n+1} \sum_{k_2=1}^{n+1} \cdots \sum_{k_{N_{\mathcal{S}}}=1}^{n+1} \sum_{s\in\mathcal{S}} \log(p(k_s, I_{1:n}^s | \Phi)) \prod_{s'\in\mathcal{S}} p(k_{s'} | I_{1:n}^{s'}, \Phi^g)\end{aligned} \tag{3.14}$$

In the subsequent developments (equations (3.15) and (3.16)), the objective is to simplify the expression A to recover the independence between sample labels in the expression of the expectation of the complete-data log likelihood (Eq. (3.12)):

$$\begin{aligned} A &= \sum_{k_1=1}^{n+1}\sum_{k_2=1}^{n+1}\dots\sum_{k_{N_\mathcal{S}}=1}^{n+1}\sum_{s\in\mathcal{S}}\sum_{k=1}^{n+1}\delta_{k,k_s}\log(p(k_s=k,I_{1:n}^s|\Phi))\prod_{s'\in\mathcal{S}}p(k_{s'}|I_{1:n}^{s'},\Phi^g) \\ &= \sum_{k=1}^{n+1}\sum_{s\in\mathcal{S}}\log(p(k_s=k,I_{1:n}^s|\Phi))\sum_{k_1=1}^{n+1}\sum_{k_2=1}^{n+1}\dots\sum_{k_{N_\mathcal{S}}=1}^{n+1}\delta_{k,k_s}\prod_{s'\in\mathcal{S}}p(k_{s'}|I_{1:n}^{s'},\Phi^g) \end{aligned} \tag{3.15}$$

This expression can be further simplified if we observe that

$$\begin{aligned} B &= \sum_{k_1=1}^{n+1}\sum_{k_2=1}^{n+1}\dots\sum_{k_{N_\mathcal{S}}=1}^{n+1}\delta_{k,k_s}\prod_{s'\in\mathcal{S}}p(k_{s'}|I_{1:n}^{s'},\Phi^g) \\ &= \left(\sum_{k_1=1}^{n+1}\sum_{k_2=1}^{n+1}\dots\sum_{k_{s-1}=1}^{n+1}\sum_{k_{s+1}=1}^{n+1}\dots\sum_{k_{N_\mathcal{S}}=1}^{n+1}\prod_{\substack{s'\in\mathcal{S}\\s'\neq s}}p(k_{s'}|I_{1:n}^{s'},\Phi^g)\right)p(k_s=k|I_{1:n}^s,\Phi^g) \\ &= \prod_{\substack{s'\in\mathcal{S}\\s'\neq s}}\left(\sum_{k_{s'}=1}^{n+1}p(k_{s'}|I_{1:n}^{s'},\Phi^g)\right)p(k_s=k|I_{1:n}^s,\Phi^g) \\ &= p(k_s=k|I_{1:n}^s,\Phi^g) \end{aligned} \tag{3.16}$$

and from A and B we get

$$\begin{aligned} Q(\Phi,\Phi^g) = &\sum_{s\in\mathcal{S}}\sum_{k\in\mathcal{K}}p(k_s=k|I_{1:n}^s,\Phi^g)\log p(k_s=k,I_{1:n}^s|\Phi) \\ &+\sum_{i=1}^{n}\lambda_i\left(\sum_{p\in R_i^{\text{Ext}}}\log p(I_i^p|\Theta_i^B)\right), \end{aligned} \tag{3.17}$$

and the new set of parameters is

$$\hat{\Phi} = \arg\max_{\Phi} Q(\Phi,\Phi^g). \tag{3.18}$$

Expectation Step. In the Expectation step, we compute for each sample $s\in\mathcal{S}$ the probability of its classification hypothesis k_s in the EM Q-expectation (Eq. 3.17):

$$\forall k\in\mathcal{K},\qquad p_s^k := p(k_s=k|I_{1:n}^s,\Phi^g) = \frac{\pi_k^g\prod_{i=1}^{n}p(I_i^s|\Theta_i^g,k_s=k)}{\sum_{\ell\in\mathcal{K}}\pi_\ell^g\prod_{i=1}^{n}p(I_i^s|\Theta_i^g,k_s=\ell)}. \tag{3.19}$$

Maximization Step. In this step, we find the new set of parameters $\hat{\Phi}$ that maximizes the Q-expectation, which can be written as the sum of independent terms

$$\begin{aligned} Q(\Phi, \Phi^g) = & \sum_i \left[\sum_s p_s^{\mathrm{b}_i} \log p(I_i^s|\Theta_i, k_s = \mathrm{b}_i) + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} \log(p(I_i^p|\Theta_i^B)) \right] \\ & + \sum_{i,s} p_s^{\mathrm{f}} \log p(I_i^s|\Theta_i, k_s = \mathrm{f}) \\ & + \sum_{s,k} p_s^k \log \pi_k \\ & + \text{constant}, \end{aligned} \tag{3.20}$$

where the constant term holds the contributions of labels b_j for $j \neq i$, which do not depend on parameters Φ. Each of the other terms can be maximized independently.

The appearance models have been defined in very general terms, and the equations derived so far are independent of the considered appearance models. We show in the following the derivation of color model update equations for color histograms. The extension of the model to other color models is discussed in the next chapter.

Estimation algorithm with color histograms. In this case we use one color histogram in each view for the background and one shared color model for the foreground. The background histogram for view i is noted H_i. The shared foreground histogram is noted H^F. The region of interest R_i^{Int} is described by its histogram noted H_i^{Int}. All color models are thus fully parametrized by $\Theta = \{H^F, H_i, H_i^{\mathrm{Int}} | i \in \{1, \cdots, n\}\}$. In this case $\Theta_i = \{H^F, H_i, H_i^{\mathrm{Int}}\}$ and sample labelling Equations 3.4 and 3.5 become

$$p(I_i^s|\Theta_i, k_s) = \begin{cases} H_i(I_i^s) & \text{if } k_s = \mathrm{b}_i, \\ H^F(I_i^s) & \text{if } k_s = \mathrm{f}, \\ H_i^{\mathrm{Int}}(I_i^s) & \text{if } k_s = \mathrm{b}_j \text{ and } i \neq j. \end{cases} \tag{3.21}$$

We also note $b \in [\![1, B]\!]$ a histogram bin and H_b the value of b for the histogram H. The histograms are normalized which means that all bin values sum up to one. In this case, the previous equation can be written

$$\text{If } I_i^s \in b,\ p(I_i^s|\Theta_i, k_s) = \begin{cases} H_{i,b} & \text{if } k_s = \mathrm{b}_i, \\ H_b^F & \text{if } k_s = \mathrm{f}, \\ H_{i,b}^{\mathrm{Int}} & \text{if } k_s = \mathrm{b}_j \text{ and } i \neq j. \end{cases} \tag{3.22}$$

with the constraints

$$\begin{cases} \sum_{b=1}^{B} H_{i,b} = 1, \\ \sum_{b=1}^{B} H_b^F = 1, \\ \sum_{b=1}^{B} H_{i,b}^{\mathrm{Int}} = 1. \end{cases} \tag{3.23}$$

Replacing the new sample labelling equations in the Q-expectation expression defined in Eq. (3.20) we obtain:

$$\begin{aligned} Q(\Phi, \Phi^g) = & \sum_{i,b} \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{b}_i} log(H_{i,b}) + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} \log(H_{i,b}) \\ & + \sum_{b,i} \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{f}} log(H_b^F) \\ & + \sum_{s,k} p_s^k \log \pi_k \\ & + \text{constant}. \end{aligned} \tag{3.24}$$

If we note $N_{i,b}^{\mathrm{Ext}}$ the number of pixels from R_i^{Ext} inside histogram bin b for view i, then

$$\begin{aligned} Q(\Phi, \Phi^g) = & \sum_{i,b} \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{b}_i} log(H_{i,b}) + \lambda_i N_{i,b}^{\mathrm{Ext}} log(H_{i,b}) \\ & + \sum_{b,i} \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{f}} log(H_b^F). \\ & + \sum_{s,k} p_s^k \log \pi_k \\ & + \text{constant}. \end{aligned} \tag{3.25}$$

Using the Lagrange multiplier for terms related to π_k with the constraint $\sum_k \pi_k = 1$, we obtain the new values for the mixture coefficients (See Appendix A for details):

$$\pi_k = \frac{1}{N_{\mathcal{S}}} \sum_{s \in \mathcal{S}} p_s^k \qquad (N_{\mathcal{S}}\text{: number of samples}). \tag{3.26}$$

Similarly, using the constraints defined in (3.23), solving $\Phi = \arg\max_{\Phi} Q(\Phi, \Phi^g)$ can be shown to come down to updating bin values as follows for the background:

$$H_{i,b} = \frac{\sum_{s \in \mathcal{S}: I_i^s \in b} p_s^{\mathrm{b}_i} + \lambda_i N_{i,b}^{\mathrm{Ext}}}{\sum_{b'=1}^{B} \sum_{s \in \mathcal{S}: I_i^s \in b'} p_s^{\mathrm{b}_i} + \lambda_i N_{i,b'}^{\mathrm{Ext}}}. \tag{3.27}$$

Likewise, the update equation for the foreground histogram reads

$$H_b^F = \frac{\sum_{i=1}^{n} \sum_{s \in \mathcal{S}: I_i^s \in b} p_s^{\mathrm{f}}}{\sum_{b'=1}^{B} \sum_{i=1}^{n} \sum_{s \in \mathcal{S}: I_i^s \in b'} p_s^{\mathrm{f}}}. \tag{3.28}$$

In each case, bin histograms are updated with a participation proportional to background and foreground label probabilities. We note the important bias added by known background pixels.

The estimation algorithm iterates between the expectation step where label probabilities are estimated according to equation (3.19) and the maximization step where model parameters are updated using equations (3.26), (3.27) and (3.28). The convergence is reached when the changes

in appearance histograms are negligible:

$$\forall i, \quad \sum_b |H_{i,b} - H^g_{i,b}| \leq \epsilon \qquad \text{and} \qquad \sum_b |H^F_b - H^{F,g}_b| \leq \epsilon \tag{3.29}$$

with superscript g indicating the estimated values at the previous iteration, and ϵ being the convergence threshold for histogram bin update.

3.5 Final segmentation

The EM scheme described in the previous sections will converge to an estimate of the color models for each view and a classification probability table for each sample. The samples would only yield a sparse image segmentation if their classifications were crudely reprojected. This is why we use the obtained estimates to build a final dense 2D segmentation of each image, combining results of sample classifications and color models. Note that this is only required after convergence in our approach, as opposed to being mandatory in the iterative process with existing approaches [35, 96]. Segmentation of view i then amounts to assigning to each pixel p of this image a binary label $l^p_i \in \{\mathrm{f}, \mathrm{b}\}$ (foreground or background) according to the current estimate of the color models Θ_i and to the set Ξ of projection positions and label posterior probabilities of all the 3D samples (see Fig. 3.5).

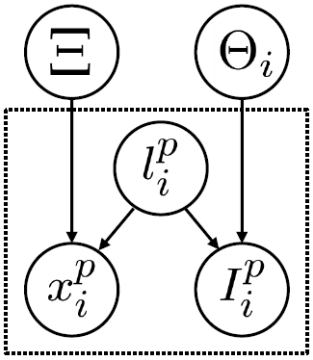


FIGURE 3.5: Relation between variables in the final segmentation problem: l^p_i, x^p_i and I^p_i are respectively the binary label, the position and the color value of pixel p in image i. Variable Ξ stands for the 3D sample positions and associated posterior label probabilities. Variable Θ_i represents the foreground/background color model.

While various strategies could be used, we follow [35] and finalize segmentation using a simple graph cut scheme similar to [27], minimizing a discrete energy in view i:

$$E = \sum_p E_d(l^p_i | \Xi, \Theta_i, x^p_i, I^p_i) + \sum_{\{p,q\} \in N_i} \alpha E_s(l^p_i, l^q_i). \tag{3.30}$$

The data related term E_d at pixel p depends first, on how likely its color is under color models obtained for image i. It also depends on how its spatial position x_p relates to projections in the

image of the set of softly classified 3D samples:

$$E_d(l_i^p|\Xi,\Theta_i,x_i^p,I_i^p) = -\log(p(x_i^p|\Xi,l_i^p)p(I_i^p|\Theta_i,l_i^p)), \tag{3.31}$$

where $p(x_i^p|\Xi,l_i^p)$ acts as prior from 3D samples. To this end, we define $\mu_{\mathcal{S}}$ the mean distance between neighbor samples in 3D. This distance (Fig. 3.6) is used to define the radius of influence $\mu_{\mathcal{S}}^i$ of each sample in view i.

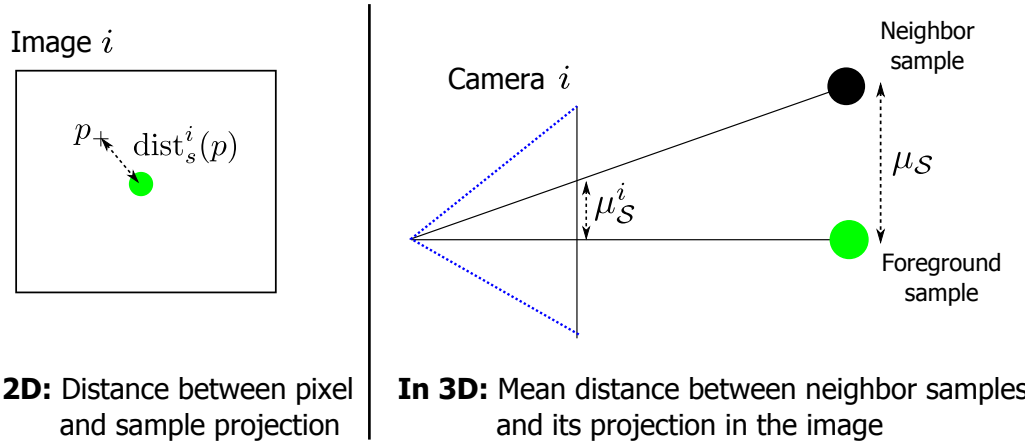


FIGURE 3.6: Sample projection in the final segmentation.

If we note $\text{dist}_s^i(p)$ the distance between a pixel p in view i and the projection of a sample s, then

$$p(x_i^p|\Xi,l_i^p) = \begin{cases} \max\limits_{\substack{s\in\mathcal{S} \\ \text{dist}_\text{s}^\text{i}(\text{p})\leq 2\mu_{\mathcal{S}}^i}} \dfrac{2\mu_{\mathcal{S}}^i - \text{dist}_\text{s}^\text{i}(\text{p})}{2\mu_{\mathcal{S}}^i}\, p_s^\text{f} & \text{if} \quad l_i^p = \text{f} \\ \text{cst} & \text{if} \quad l_i^p = \text{b}_i \end{cases} \tag{3.32}$$

- In the case $l_i^p = \text{f}$, it is inversely proportional to the distance to the closest projection of a foreground sample. This allows a smooth projection of inferred foreground information.
- In the case $l_i^p = \text{b}$, this probability is constant (0.8 in our case, to avoid penalizing foreground labelling, the probability value of which is rarely equal to 1).

The second term $p(I_i^p|\Theta_i,l_i^p)$ is based on the foreground or background histograms previously obtained:

$$p(I_i^p|\Theta_i,l_i^p) = \begin{cases} H_i(I_i^p) & \text{if} \quad l_i^p = \text{b}, \\ H^F(I_i^p) & \text{if} \quad l_i^p = \text{f}. \end{cases} \tag{3.33}$$

The second energy term E_s in (3.30) enforces the smoothness over the set N_i of neighbor pixels. It can be any energy that favors consistent labelling in homogeneous regions. In our implementation we use a simple inverse distance between neighbor pixels. A final remark is the possible inter-view inconsistencies in segmentation details due to the view-independence of segmentations in this final step.

Dataset	max. view number	User interact.	Comparison with GrabCut	Comparison with multi-view
Arts [108] *Martiaux*	16		✓	
Bear[1]	15			✓
Bike[1]	35			✓
Bust[109]	26			✓
Couch[1]	11		✓	✓
Car[1]	44			✓
Pig[2]	27	✓		✓
Rabbit[2]	27	✓		✓

TABLE 3.1: The different calibrated multi-view datasets used. For each one, we indicate the original number of views, the comparison we performed and if user interaction was needed. Multi-view segmentation methods we compare with are [1, 2, 35, 96].

3.6 Experimental results

In this section we present segmentation results of the proposed method on various calibrated multi-view datasets summarized in Table 3.1. These datasets are used by state of the art methods in multi-view segmentation. Some of them correspond to captures in constrained environments such as *Arts Martiaux*, *Pig* and *Rabbit*. We will see that despite the relative simplicity of these datasets, monocular methods do not produce good results. *Bear* and *Couch* correspond to more realistic scenarios but are still relatively simple. The color information is not very rich and there is a limited ambiguity between foreground and background objects. The rest of the datasets are more challenging and already correspond to complex scenarios. The plane based reconstruction method [1] achieves good results, but many points of view are necessary to estimate depth information.

The objective of the experiments presented here is twofold: first, validate the n-tuple approach and, second, perform a comparative study with both monocular and multi-view segmentation approaches to show the improvements over the state of art. We also design some experiments to further investigate the sensitivity of the n-tuple model to the number of samples and to the number of viewpoints.

We use joint HSV color histograms, with $B = 32^3$ bins. Samples in $\mathcal{S}$ are drawn from the common visibility domain of all cameras. This defines a bounding volume that is used to define regions R_i^{Int} in each image i and to find a first set of background pixels, but the method is also entirely compatible with user inputs. For our initial experiments, we used a regular 3D sampling with about $N = 50^3$ samples. All the labels are set to the same probability for all the samples and we start the iterative process by a maximization step. We run our algorithm on a 2.5 GHz Intel Xeon PC with 12GB RAM, using a sequential C++ implementation.

3.6.1 Case study

To have a better understanding of the proposed algorithm prior to the evaluation, we propose a detailed study on the *Bust* dataset. This dataset consists in 26 views around a statue in a generic indoor environment. Results (Fig. 3.7) on this dataset raise interesting questions whose responses clarify some aspects of the proposed model.

Using calibration information, a bounding box of the common visibility volume is computed. Projected in the images, this common visibility volume defines the regions: R^{Int} containing the foreground object and R^{Ext} defining the prior on background models. Samples are then created inside the common visibility with labels set to foreground with probability $p_s^{\text{f}} = 0.8$.

The algorithm starts with a maximization step and Fig. 3.7 shows the iterative process that leads to the segmentation. Convergence is reached in 5 iterations. For each step we show samples labelled as foreground with a high probability $p_s^{\text{f}} > 0.8$ along with pixel level segmentation. The pixel level segmentation is only necessary at convergence, but it is estimated here to give an insight on the expected results.

One of the advantages of the proposed estimation algorithm is the fact that no hard decision is taken at any time. This means that samples once labelled as background with high probability, can be relabelled foreground during convergence if this is consistent in all the views.

What is foreground? Given the input images (Fig. 3.7) one would expect to see the table and the black box segmented as foreground since they are seen by all the cameras. However, we should recall the foreground definition used in this work: *The foreground object is the object seen by all the cameras with appearance different from background*. Thus, the numerous black elements in the background prevent the black pedestal from being segmented as foreground. Likewise for the table that matches the cabinet (Fig. 3.8) and which is not entirely seen by all the cameras. Finally, using a shared color model for the foreground doesn't imply an assumption of inter-view shared appearance. The idea here is to take advantage of foreground shared appearance when it is verified, without making it a primary assumption of the approach. Assuming a shared appearance for the object of interest is a much stronger assumption and is unlikely to work in the multi-view context. The head is the first example of this: it has very limited shared appearance across the views (due to illumination differences) but it is still identified as a foreground object. The second example is the black base which has more similar appearance across the views, but it is nonetheless labelled as background.

Necessity of the final segmentation step. The result of the proposed iterative process is an estimate of sample labelling and foreground/background color models. However, the obtained appearance models can be ambiguous (Fig. 3.8). This can be explained by a more precise analysis of the background update equation (3.27). Models for view i are updated according to the label probability $p_s^{\text{b}i}$ for the sample. Depending on the number of views and the discriminative power of each of them, this probability can be very small and the participation to the corresponding histogram bins very limited. In Fig. 3.8 we show foreground samples in green and samples labelled by the current view as background in purple ($p_s^{\text{b}i}$ is the highest). These samples have a

	Iteration 2		Iteration 3		Convergence	
Input Images	Samples	Segmentation	Samples	Segmentation	Samples	Segmentation

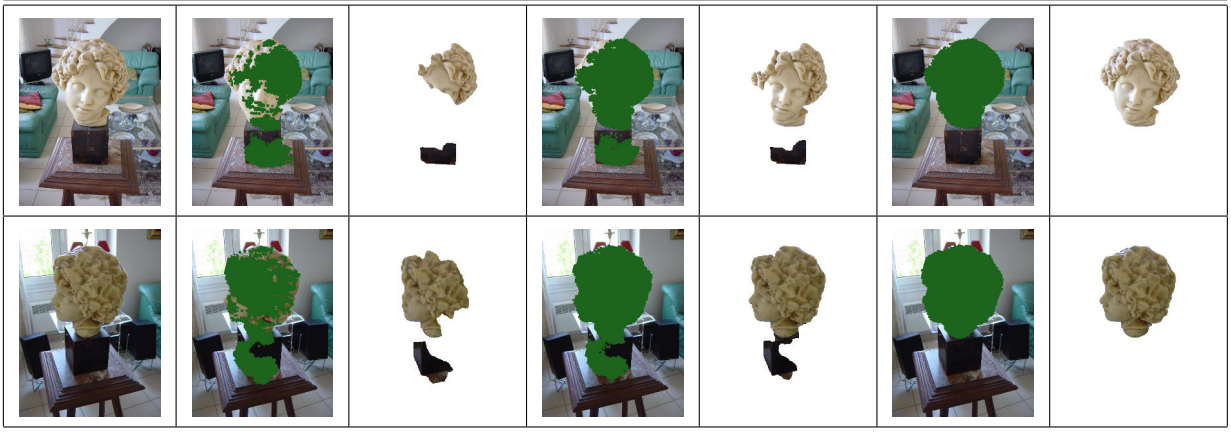
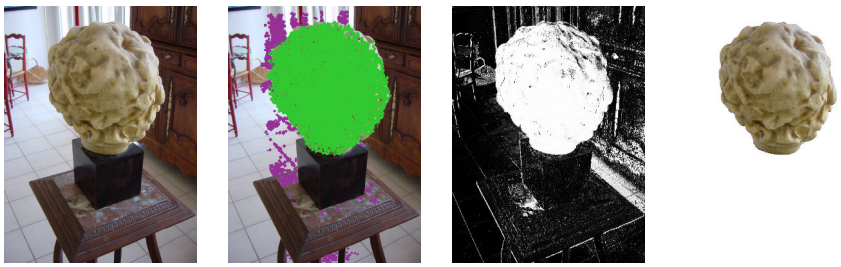


FIGURE 3.7: Intermediate results of the algorithm on the *Bust* dataset ($n = 13$ views) with $N_{\mathcal{S}} = 50000$ samples. Green dots indicate the projection of the 3D samples from set $\mathcal{S}$ with high foreground probability ($p_s^{\mathrm{f}} > 0.8$). Segmentation at each iteration is performed using the method described in §3.5. These intermediate segmentation results are used to study algorithm convergence (Fig. 3.13).



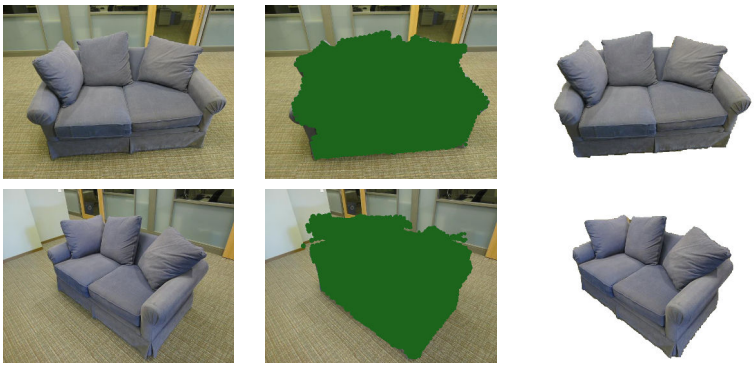
Input image | Samples at convergence | Foreground color model | Final segmentation

FIGURE 3.8: Illustration of the final segmentation step (§3.5) on the *Bust* dataset. In the second column we show foreground samples (in green) and samples labeled as background by the current view (in purple). Despite the correct labelling of foreground 3D samples, the obtained color models are still ambiguous (third column): white pixels indicate pixels most likely to be foreground. This demonstrates the necessity of the final segmentation step leveraging 3D sample positions and appearance models.

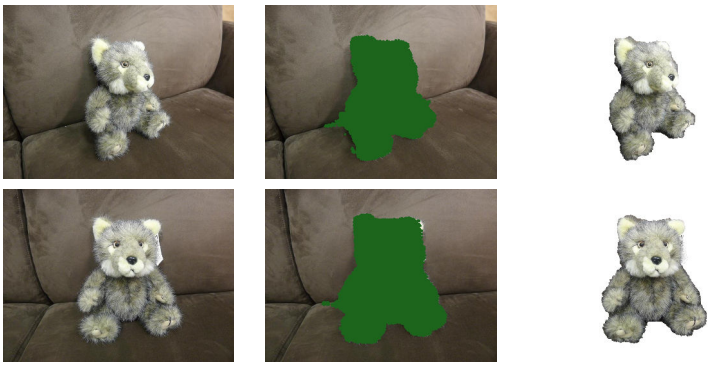
clear participation in the appearance models of the view, but it is less obvious for the other samples (labelled as background by the other views but not shown here). This explains the obtained color models at convergence and the necessity to perform a final segmentation step, leveraging sample positions and color models.

3.6.2 Qualitative validation

For all the datasets, we show the foreground samples at convergence and the segmentation results. The method performs almost perfectly on the *Bust*, *Couch* and *Bear* datasets (Figs. 3.7, 3.9(a) and 3.9(b)). It can handle multiple foreground objects as in the *Arts Martiaux* dataset.

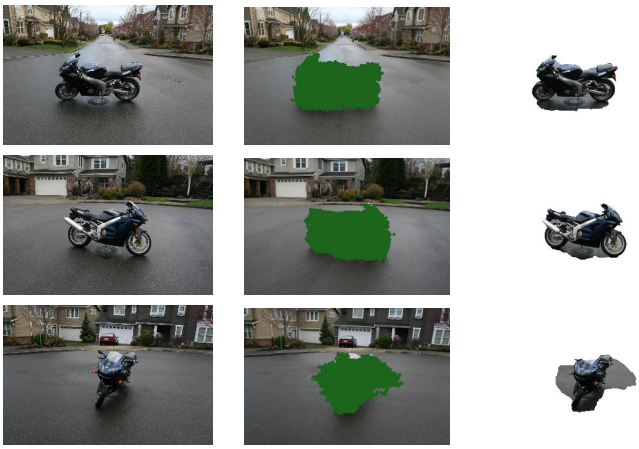


(a) *Couch* dataset: 10 views - Runtime $\approx 15s$



(b) *Bear* dataset: 10 views - Runtime $\approx 15s$

(c) *Car* dataset: 14 views - Runtime $\approx 30s$



(d) *Bike* dataset: 11 views - Runtime $\approx 20s$

FIGURE 3.9: Results on datasets from [1]. We show (green dots) the projection of samples with high foreground probability ($p_s^{\mathrm{f}} > 0.8$) at convergence, and final segmentation.

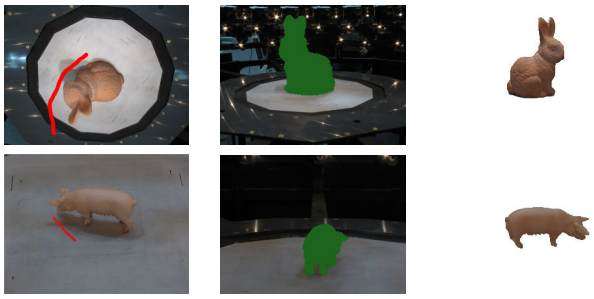


FIGURE 3.10: Results on *Pig* and *Rabbit* [2]: The user indicates the background region in one view (red stroke). Green dots indicate the projections of samples with high foreground probability ($p_s^{\mathrm{f}} > 0.8$) at convergence. The last column is the final segmentation.

On *Car* and *Bike* (Fig. 3.9(c) and Fig. 3.9(d)), with significantly fewer points of view than state of art approaches, we achieve results of comparable quality. However in some views, the foreground/background color models are more ambiguous and this affects the segmentation results. This point is discussed in more detail in §3.8.

Although our method proposes an automatic initialization, we can also incorporate user interaction. For example, on the *Rabbit* and *Pig* datasets the shadow on the ground is also seen by all the views and falls within our definition of the foreground. As in [2], user interaction is needed to resolve ambiguities. Typically with our method, one stroke in a single view is sufficient to propagate information to other views (Fig. 3.10).

To demonstrate the advantages of using a multi-view approach, we compare our approach with the *OpenCV* [110] implementation of GrabCut [3]. The GrabCut algorithm is initialized with a region of size equivalent or smaller than the region of interest used by our method. The results (Fig. 3.11) show that in a monocular approach, it is hard to eliminate background colors that are not present outside the bounding box. In contrast, our approach benefits from the information of the other views and provides a correct segmentation.

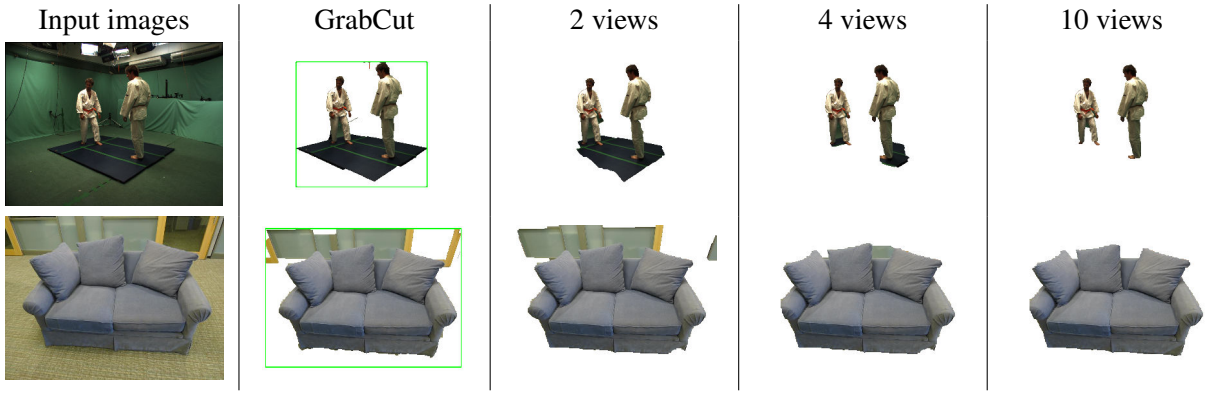


FIGURE 3.11: Comparison with GrabCut [3] monocular approach and influence of the number of views on our segmentation results on *Art martiaux* and *Couch* datasets. The green box is used only with GrabCut. This box is always smaller than the one obtained from the common visibility volume bounding box used by our method.

Dataset	Mean Error (%)	Hit Rate (%)	False Alarms (%)
Bust	0.2 ± 0.1	99.4 ± 0.01	0.7 ± 0.3
Arts Martiaux	0.5 ± 0.2	97.5 ± 0.3	2.7 ± 1.4
Couch	1.2 ± 0.8	97.0 ± 2.8	0.1 ± 0.1
Bear	2.7 ± 1.5	94.5 ± 3.0	7.0 ± 9.0
Car	2.8 ± 0.8	98.8 ± 0.8	16.7 ± 8.8
Bike	2.4 ± 1.1	96.7 ± 2.1	25.0 ± 13.3

TABLE 3.2: Full evaluation of the proposed approach on the different datasets.

3.6.3 Quantitative evaluations

In this section we propose a quantitative evaluation of the proposed method, based on three performance metrics [35]: *mean error*, which gives a global measure of segmentation errors; *hit rate*, which indicates the proportion of well segmented foreground; and *false alarm rate*, which indicates the proportion of background segmented as foreground. Denoting W_b^a the subset of pixels from the set $a \in \{F, B\}$ (foreground or background) in the ground truth that are labelled as $b \in \{F, B\}$ by our segmentation, and $N(W_b^a)$ its cardinal, the performance metrics are defined as follows:

$$\textit{Mean Error} = \frac{N(W_F^B) + N(W_B^F)}{\textit{Number of pixels}}, \tag{3.34}$$

$$\textit{Hit Rate} = \frac{N(W_F^F)}{N(W_F^F) + N(W_B^F)}, \tag{3.35}$$

$$\textit{False Alarms} = \frac{N(W_F^B)}{N(W_F^B) + N(W_B^B)}. \tag{3.36}$$

We also define

$$\textit{Accuracy} = 1 - \textit{Mean Error}, \tag{3.37}$$

$$\textit{Missed Rate} = 1 - \textit{Hit Rate}. \tag{3.38}$$

A full quantitative evaluation of the method is proposed in Table 3.2. The mean and standard deviation are computed on segmentation results for all the views.

Shared vs. per-view foreground model. In the original version of this work [108], we adopted a reasoning similar to the segmentation method proposed by Pham *et al.* [111] where foreground and background color histograms were considered as view dependent and complementary. Actually, this choice proved to be less efficient because this assumption of color dissimilarity is not always verified, especially when working with outdoor datasets with richer colors (see §3.7). Using a shared appearance model for the foreground object biases density

	Our Method		Djelouah	Kowdle	Vicente
	(a)	(b)	[108]	[1]	[39]
Couch	7 98.7 ± 0.9	10 98.8 ± 0.8	10 98.8 ± 0.8	10 99.6 ± 0.1	not available
Bear	5 97.3 ± 1.3	15 97.3 ± 1.5	15 98.8 ± 0.4	15 98.8 ± 0.4	not available
Car	11 97.4 ± 0.8	44 97.2 ± 0.8	44 $*$	44 98.0 ± 0.7	44 91.4 ± 4.3
Bike	11 97.4 ± 1.5	35 97.6 ± 1.1	35 $*$	35 99.4 ± 0.4	35 88.9 ± 6.3

TABLE 3.3: *Comparative* results, using the proportion of correctly labelled pixels in the image (*Accuracy* in %). For each dataset, the number of views used is indicated. Our method is tested with various number of views (see text for details on the experimental protocol). Column (a) indicates the minimum number of views needed for our method to give reasonable results. Column (b) indicates the results obtained using all the views. The symbol $*$ indicates that the method failed to segment the object).

estimation toward the object if the considered colors are present in multiple images. This is verified by the quantitative evaluation of table 3.3.

Number of views. We also study the influence of the number of views used (Table 3.3 and Fig. 3.12) on four different datasets: *Bear, Couch, Bike* and *Car*. For a given number of views n, we randomly select n widespread views among those available in the dataset and compute the segmentation. This test is performed 10 times for each number of views and the mean *Accuracy* is estimated. Using more views improves the segmentation results. However, contrary to [1], using one third of dataset views is enough to produce good segmentation results (Table 3.3). We would like to emphasize the challenging nature of color ambiguities between foreground and background in *Car* and *Bike* datasets.

Convergence of EM. As a fully derived EM, our model converges to a local minimum. Our experiments show that silhouette-consistent objects (whose appearance is strongly different from the initial partial background) are strong easily reached minima (Fig. 3.13). The algorithm converges in 6 iterations for the considered datasets and even in 2 iterations for configurations such as *Couch* and *Bear*.

Complexity and number of samples. Each iteration complexity is linear in the number of samples and views, with running time of a few seconds (see Fig. 3.9 for details), whereas [1] indicates 2 minutes processing time on a single image due to the piece-wise planar depth map estimation and [35] indicates several minutes.

Grid based methods [2] and [96] show the same complexity properties but, as discussed in [96], the number of voxels is a key factor in the quality of segmentation: for an image resolution $M \times M$ a grid of $N = M^3$ voxels must be used to achieve pixel level precision. This becomes quickly unfeasible and, to circumvent the problem, they propose to perform segmentation at image level instead of directly taking grid projection. Still, the two methods rely on the same

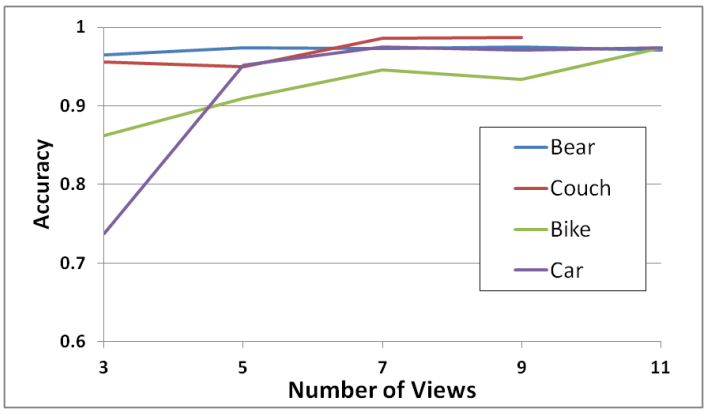


FIGURE 3.12: Evolution of segmentation results (using *Accuracy*) according to the number of views used.

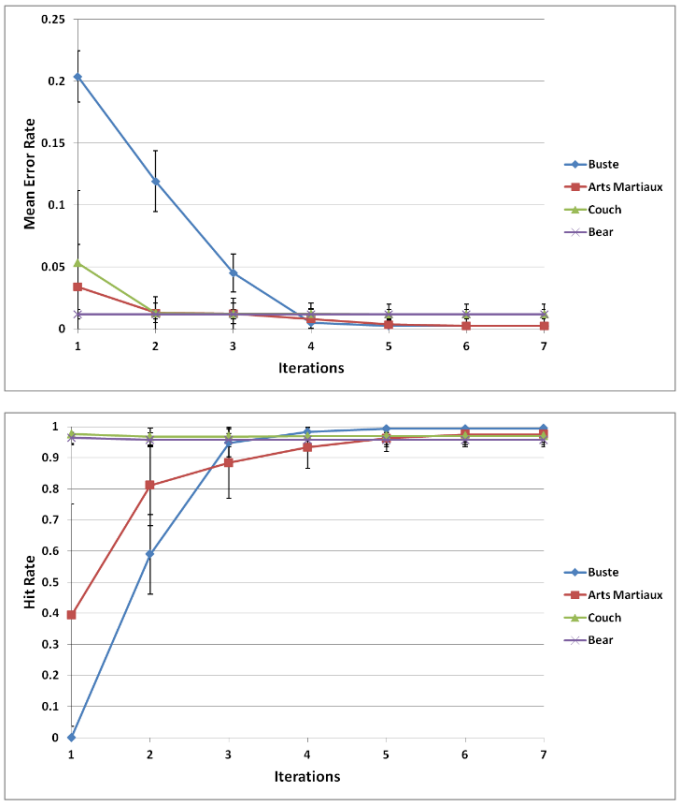


FIGURE 3.13: Convergence study: Mean error rate and hit rate (with confidence intervals) on *Bust, Arts Martiaux, Couch* and *Bear*. For all the datasets, convergence is reached in 6 iterations. Only 2 iterations are needed for simpler scenarios like *Bear* and *Couch*.

framework, using approximately 300^3 voxels and achieving reasonable runtime only through a GPU implementation. Thus, our method based on sparse sampling of the space, presents a key improvement over voxel based methods.

Fig. 3.14 shows *Missed rate* values for different numbers of samples on three datasets. Such a random draw in the limited range of $20^3 \sim 50^3$ samples was enough to converge to a correct estimation of color models. This reflects directly on the processing time. For example, on the *Bust*

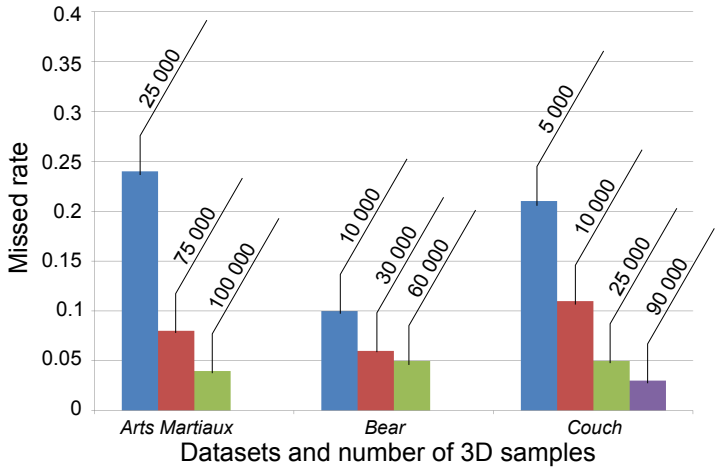


FIGURE 3.14: Results with different numbers of 3D samples. The missed rate is used as the error measure (lower is better).

dataset our non-optimized C++ implementation performs the segmentation in 10s, while [96] report 5s with a highly optimized GPU implementation. We also note that our method is entirely compatible with a GPU implementation (highly parallel E- and M-steps). This would drastically reduce processing time and would be extremely beneficial when extending the method to video sequences.

3.7 Discussion

The proposed model is built on the assumption that foreground and background objects have a different color appearance. This explains the results on *Bust* (Fig. 3.7) where the black objects in the background prevent the black base from being segmented as foreground. This assumption also implies that the foreground and background color models must be discriminative enough. This condition is hardly met when working on outdoor datasets. For example the *Car* and *Bike* datasets from [1] are really challenging due to the color ambiguity between foreground and background. The approach of [1] benefits from the short baseline between the views, which is used to estimate depths and more precisely a plane based reconstruction (Fig. 3.15).

This situation is even more clearly illustrated on the *Plant* dataset where only the green leaves are segmented as foreground. Indeed, the blue elements in the background induces a strong bias to label the blue pot as background despite the face that it is seen in all the views.

Another point to discuss is the final segmentation. As explained in section §3.5, various strategies can be used. In some cases, using a graph cut is not the best choice, especially when the object has thin parts. This shrinking bias is a known issue of graph-cut based methods [49] and Fig. 3.17 illustrates such a situation, where the samples are correctly labelled but the graph cut segments the thin parts as background. This leads to view inconsistent segmentations in the final step. An approach to solve this problem, using the same final graph-cut based segmentation, would consist in definitely setting foreground label for pixels at the projection of the most probable foreground samples.

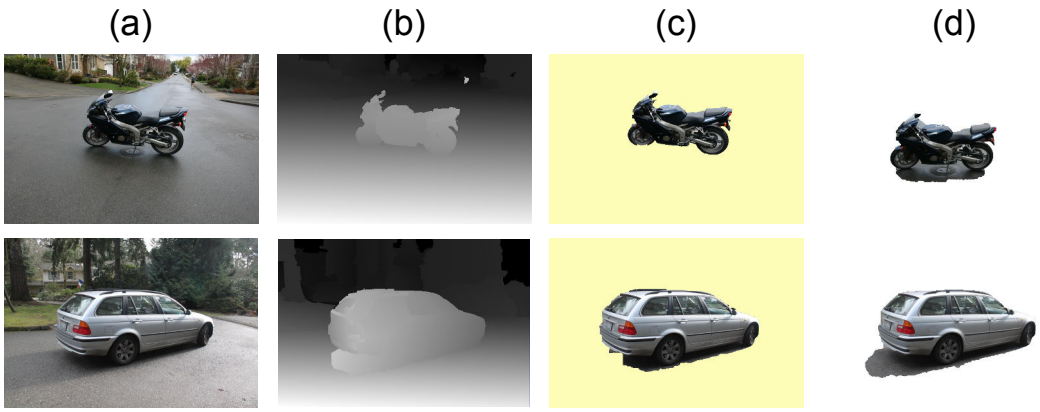


FIGURE 3.15: On the *Car* and *Bike* datasets, we can see the advantages of using the depth map (column b) in the method proposed by Kowdle *et al.* [1] (column c). Our method relies only on color information (column d).

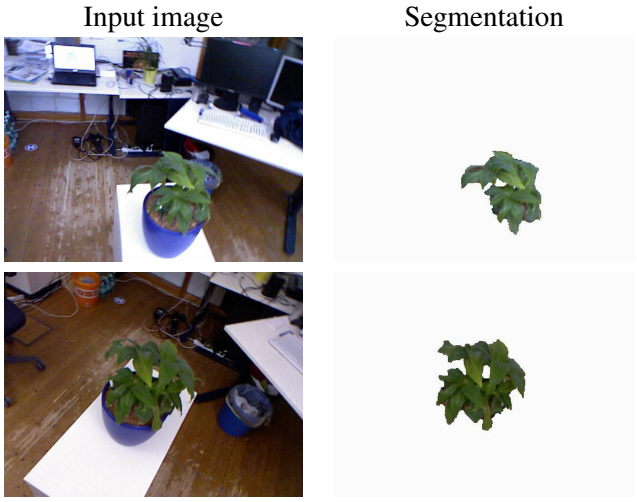


FIGURE 3.16: Segmentation results on *Plant* dataset: because of the blue elements in the background, only the green leaves are segmented as foreground.

3.8 Conclusion

In this chapter the sparse 3D sampling framework for multi-view segmentation was presented. We argue that the approach presented here avoids the typical shortcomings of previous state of the art methods; in particular, it successfully embeds the multi-view consistency in the proposed

FIGURE 3.17: Results on the *Chair*[1] dataset (using 4 views). Green points indicate projection of samples with high foreground probability. The last column is the final segmentation, with thin legs being lost while they were correctly labelled at the sparse samples level.

n-tuple formulation. The generative model is probabilistically sound and allows coherent probability estimation for foreground and background labels from appearance models. This generative model was used in an effective expectation maximization scheme, with good results on state of the art datasets, notably using fewer viewpoints than state of art methods.

Despite its major strengths, the method still exhibits difficulties in challenging situations where the foreground/background dissimilarity assumption is not met. To further investigate the potential of the n-tuple model, we will show in the following chapters how different color models or modalities such as depth can be naturally integrated.

Chapter 4

Extension to other color models and others modalities

4.1 Introduction

In the previous chapter, a multi-view segmentation method was presented based on a sparse sampling of the 3D space. The interest of the method was shown both at the modelling level, through the sound probabilistic model that coherently labels 3D samples using estimated appearance models, and at the computation level, through the expectation maximization estimation algorithm that efficiently solves for foreground/background model estimation and sparse 3D samples labeling. It was also clearly shown that a sparse sampling approach allows us to express multi-view consistency constraints at minimum cost, while achieving better or equivalent results than state of the art multi-view segmentation methods.

In this chapter, the advantages of the proposed sparse 3D sampling scheme for multi-view segmentation are further explored. We show the flexibility of the proposed generative model by exploring both a different appearance model for foreground and background based on a Gaussian Mixture Model (GMM), and the usage of depth information when available. First, we demonstrate that Gaussian mixture models can be seamlessly used in our generative model. Using such models allow for better propagation of appearance models. The results obtained show, however, that despite their better representative power, the obtained segmentation quality doesn't justify the increased computation time required to process the GMM update. In the case of depth information, we adapt the proposed generative model to account for the presence of this extra information. We discuss the possible ways of using this extra information and how it improves segmentation results.

Exploring these two directions permits a better understanding of the advantages of the current approach but also reveals the limitations that should be addressed and provides directions for improvement that are further explored in the course of this thesis.

4.2 Multi-view Segmentation with GMMs

In image segmentation research work, the two most widely used approaches to model appearance are histograms [38, 75, 112] and GMMs [3, 113]. Histograms are usually used with the objective of fast and easy representation of appearance statistics and as an approximation of the probability density function for image regions colors. An interesting direction was proposed in [112], where the assumed complementary nature of the foreground and background was used as a driving constraint for segmentation and foreground/background color model updates. This complementary nature of foreground and background appearances has inspired the first model we proposed during this thesis [108] (see the discussion in the previous chapter for more details).

Gaussian mixture models are able to represent arbitrarily complex probability density functions [114]. GMMs are also useful for representing patch-wise shapes of images and clusters. However, some problems usually arise when selecting the number of components: with too many components, the mixture may over-fit the data and with too few components, the mixture may not be flexible enough to approximate the true underlying model. Some very important work in segmentation [3, 113] relies on this tool with good results. The Gaussian mixture model hypothesis is also used in multi-view segmentation [2, 96]. Usually, estimating a Gaussian mixture model is done using the expectation maximization algorithm as way to conduct Maximum Likelihood (ML) estimation. In the context of segmenting a single image, a fixed low number of mixture components is usually enough to achieve reasonable results. With a limited number of components, the GMM will not properly fit the actual underlying density distribution. It is not a crucial issue in the monocular case since we are more interested in modeling the difference in the probabilities between the possible labels rather than perfectly fitting the density function. However, in the context of multi-view segmentation, labeling 3D points relies on the fusion of information from different views and it is of fundamental importance to correctly approximate the different probability density functions. Using point estimate methods like the EM algorithm requires a good prior on the number of mixture components. In the general case, cross-validation techniques are used to identify the best number of components [114, 115]. Just using a large number of components is not a good solution in this case, as it quickly leads to over-fitting and singularity problems. This issue is mentioned by Kolev. *et al.* [2], where only a single Gaussian is used to approximate foreground and background distributions. Reinbacher *et al.* [96], while extending the work in [2], do not provide details on how the over fitting problem is addressed.

In this section, we show how variational inference techniques can be used to estimate Gaussian mixture models of foreground and background in the context of the generative model proposed in the previous chapter. By using variational inference methods, we avoid typical over fitting problems due to incorrect number of mixture components that arise in classic maximum-likelihood point estimation like EM [116].

4.2.1 Variational mixture of Gaussians

The estimation of a variational mixture of Gaussians is a classic case study in Bayesian inference. This section presents the main ideas developed in [59, 107, 116] which we will build on

later to estimate foreground/background appearance in the multi-view segmentation generative model. The classical mixture of Gaussians model is usually described by the graphical model in Fig. 4.1.

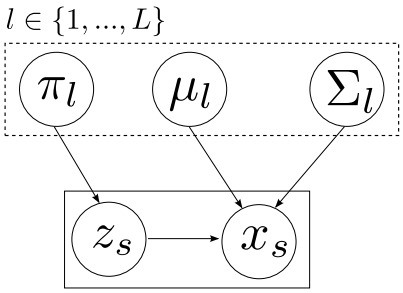


FIGURE 4.1: Graphical model representing the relationship between the different variables in a Gaussian mixture model. Values of parameters μ_l (mean), Σ_l (covariance matrix) and π_l(mixture proportion) of the L components of the mixture, are estimated using the expectation maximization algorithm.

The data set of observations $\{x_1, ..., x_{N_s}\}$ is assumed to be generated by one of the L components of the mixture. The latent variable z_s, associated to each observed data point x_s, indicates which component (with mixing coefficient π_l, mean μ_l and covariance matrix Σ_l) is responsible for this observation. The corresponding log-likelihood function is

$$\log p(X|\pi, \mu, \Sigma) = \sum_s \log \left(\sum_{l=1}^{L} \pi_l \mathcal{N}(x_s|\mu_l, \Sigma_l) \right). \tag{4.1}$$

In this scenario, the expectation maximization algorithm is usually used to find the maximum likelihood solutions, and results in an estimation algorithm that iterates between an expectation step (E Step), where the posterior probabilities for each possible value of the latent variables z_s is computed, and a maximization step (M Step), where models are updated according to previously estimated probabilities. The expectation step corresponds to

$$p(z_{sl}) = p(z_s = l|X, \pi, \mu, \Sigma) = \frac{\pi_l \mathcal{N}(x_s|\mu_l, \Sigma_l)}{\sum_{j=1}^{L} \pi_j \mathcal{N}(x_s|\mu_j, \Sigma_j)}. \qquad \text{(E Step)} \tag{4.2}$$

The new model parameters are estimated as follows:

$$\begin{aligned}
\pi_l^{\text{new}} &= \frac{N_l}{N_s} = \frac{\sum_{s=1}^{N_s} p(z_{sl})}{N_s}, && \text{(M Step for mixing coefficients)} \\
\mu_l^{\text{new}} &= \frac{1}{N_l} \sum_{s=1}^{N_s} p(z_{sl}) x_s, && \text{(M Step for mean values)} \\
\Sigma_l^{\text{new}} &= \frac{1}{N_l} \sum_{s=1}^{N_s} p(z_{sl})(x_s - \mu_l^{\text{new}})(x_s - \mu_l^{\text{new}})^T. && \text{(M Step for covariance matrices)}
\end{aligned} \tag{4.3}$$

As pointed by Bishop [59], applying the maximum likelihood approach to the mixture of Gaussians has several problems due to the presence of singularities such as components collapsing into specific points. To circumvent this sort of problems, we adopt a variational inference method.

We first describe the variational inference in the general case. If we consider the set of variables X and the set of unknown parameters and latent variables Z, then the probabilistic model specifies the joint distribution $p(X, Z)$ and the goal is to approximate the posterior distribution $p(Z|X)$ and model evidence $p(X)$. If we introduce a distribution $q(Z)$ over the set of unknown variables, then

$$\log p(X) = \mathcal{L}(q) + \mathrm{KL}(q||p) \tag{4.4}$$

with

$$\begin{aligned} \mathcal{L}(q) &= \sum_Z q(Z) \log \left(\frac{p(X, Z)}{q(Z)} \right) \\ \mathrm{KL}(q||p) &= -\sum_Z q(Z) \log \left(\frac{p(Z|X)}{q(Z)} \right) \end{aligned} \tag{4.5}$$

From the properties the Kullback-Leibler divergence we can show that $\mathrm{KL}(q||p) \geq 0$ and thus $\mathcal{L}(q)$ is a lower bound on $\log p(X)$. We can maximize this lower bound by optimizing with respect to $q(Z)$. In this case, it is equivalent to minimizing the Kullback-Leibler divergence, which vanishes when $q(Z)$ equals the posterior distribution $p(Z|X)$.

In variational inference, the distribution $p(Z|X)$ is supposed to be intractable, but considering only a limited family of distributions $q(Z)$ of simpler form, we try to find the closest to the true posterior distribution. The limitation is imposed to achieve tractability, and there is no overfitting issues with a higher flexibility since this will only result in a better approximation of the posterior.

We will not present the detailed derivation of inference equations, but only the main ideas. If we assume a particular model, where the latent variable set Z consists of M disjoint and independent groups. Then, the distribution $q(Z)$ can be factorized as follows:

$$q(Z) = \prod_{i=1}^{M} q_i(Z_i). \tag{4.6}$$

We use this factorization in the expression of $\mathcal{L}(q)$. If we consider the particular set Z_j, we can look for the optimal solution for the distribution $q_j(Z_j)$ while keeping all the other distributions $\{q_{i \neq j}(Z_i)\}$ fixed. The general expression of the optimal solution $q_j^\star(Z_j)$ is

$$\log q_j^\star(Z_j) = \mathbb{E}_{i \neq j}[\log p(X, Z)] + \text{const.} \tag{4.7}$$

This solution depends on the expectation computed over the other factors $q_{i \neq j}(Z_i)$ and therefore to find all the solutions, these factors are first initialized with an appropriate solution, then an iterative process cycles through all the factors, replacing each one of them with the updated

value. In the equations derived so far there is no limitation on the form of the factors $q_i(Z_i)$, but to keep the model tractable they are usually restricted to a certain family of functions.

In the case of Gaussian mixture model, we use a conjugate prior distribution over the model parameters. A Dirichlet distribution over the mixing coefficient π with normalized constant $C(\alpha_0)$ is used

$$p(\pi) = \text{Dir}(\pi|\alpha_0) = C(\alpha_0) \prod_{l=1}^{L} \pi_l^{\alpha_0 - 1} \tag{4.8}$$

and a Gaussian-Wishart prior on the mean and precision of each component

$$p(\mu, \Lambda) = p(\mu|\Lambda)p(\Lambda) = \prod_{l=1}^{L} \mathcal{N}(\mu_l|m_0, (\beta_0\Lambda_l)^{-1})\, \mathcal{W}(\Lambda_l|W_0, \nu_0). \tag{4.9}$$

For the sake of simplicity, we use precision matrices instead of covariance matrices ($\Lambda_l = \Sigma_l^{-1}$) and the parameters of the Gaussian-Wishart distribution are set according to available prior information on mean values and precision matrices. Otherwise the least informative priors are used. In this work, we use the notations and expressions for the different distributions as defined in [59].

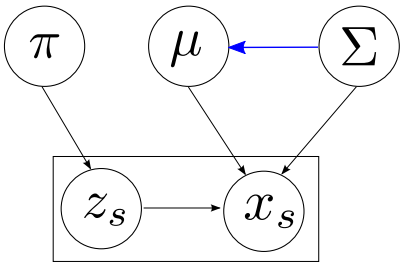


FIGURE 4.2: Graphical model representing the relationship between the different variables in a Bayesian Gaussian mixture model. Here we consider a Gaussian-Wishart prior on mean and covariance matrix. This prior implies a link between the covariance matrix and the mean as illustrated by the extra arrow not present in classic GMM (Fig. 4.1).

The relations between the variables in the Bayesian mixture of Gaussians is described in Fig. 4.2. Following this model, the joint distribution is

$$p(X, Z, \pi, \mu, \Lambda) = p(X|Z, \mu, \Lambda)\; p(Z|\pi)\; p(\pi)\; p(\mu|\Lambda)\; p(\Lambda). \tag{4.10}$$

The variational distribution over the set of latent variables and model parameters is

$$q(Z, \pi, \mu, \Lambda) = q(Z)\; q(\pi, \mu, \Lambda) = q(Z)\; q(\pi) \prod_{l=1}^{L} q(\mu_l, \Sigma_l). \tag{4.11}$$

Using this decomposition with the general result of Eq. 4.7, the update equation for the factor $q(Z)$ can be derived as follows

$$\log q^\star(Z) = \mathbb{E}_\pi[\log p(Z|\pi)] + \mathbb{E}_{\mu,\Lambda}[\log p(X|Z, \mu\Lambda)] + \text{const.} \tag{4.12}$$

Since the labels z_n are independent for each data point x_n

$$\log q^\star(Z) = \sum_{s=1}^{N_s} \sum_{l=1}^{L} z_{sl} \log \rho_{sl}, \tag{4.13}$$

where for a D dimensional data variable x:

$$\log \rho_{sl} = \mathbb{E}[\log \pi_l] + \frac{1}{2}\mathbb{E}[\log |\Lambda_l|] - \frac{D}{2}\log(2\pi) + \frac{1}{2}\mathbb{E}_{\mu_l,\Lambda_l}[(x_s - \mu_l)^T \Lambda_l (x_s - \mu_l)]. \tag{4.14}$$

Each label z_{sl} has an expected value

$$\mathbb{E}[z_{sl}|\text{all the rest}] = r_{sl} = \frac{\rho_{sl}}{\sum_{j=1}^{L} \rho_{sj}}. \tag{4.15}$$

As already mentioned, this optimal solution $\log q^\star(Z)$ depends on the other variables and the different equations must be solved iteratively. Before estimating the optimal solutions $\log q^\star(\pi, \mu, \Lambda)$ we define the quantities

$$\begin{aligned} N_l &= \sum_{s=1}^{N_s} r_{sl} \\ \bar{x}_l &= \frac{1}{N_l} \sum_{s=1}^{N_s} r_{sl} x_s \\ S_l &= \frac{1}{N_l} \sum_{s=1}^{N_s} r_{sl} (x_s - \bar{x}_l)(x_s - \bar{x}_l)^T, \end{aligned} \tag{4.16}$$

and the optimal solution for the other variables is a Dirichlet distribution for $q^\star(\pi)$

$$q^\star(\pi) = \text{Dir}(\pi|\alpha_1, ..., \alpha_L) \qquad \text{with} \qquad \alpha_l = \alpha_0 + N_l, \tag{4.17}$$

and a Gaussian-Wishart distribution for each component

$$q^\star(\mu_l, \Lambda_l) = \mathcal{N}(\mu_l|m_l, (\beta_l \Lambda_l)^{-1}) \mathcal{W}(\Lambda_l|W_l, \nu_l), \tag{4.18}$$

with

$$\begin{aligned} \beta_l &= \beta_0 + N_l \\ m_l &= \frac{1}{N}(\beta_0 m_0 + N_l \bar{x}_l) \\ W_l^{-1} &= W_0^{-1} + N_l S_l + \frac{\beta_0 N_l}{\beta_0 + N_l}(\bar{x}_l - m_0)(\bar{x}_l - m_0)^T \\ \nu_l &= \nu_0 + N_l. \end{aligned} \tag{4.19}$$

We note that the derived estimation algorithm is closely related to the expectation maximization and involves cycling between two steps. The first where the expectation for each data label $\mathbb{E}[z_{nl}]$ is estimated using Eq. 4.15 and the second where the variational distribution of the parameters is updated using Eq. 4.17 and Eq. 4.18.

4.2.2 Color GMM as appearance model for multi-view segmentation

The sparse multi-view segmentation model described in chapter 3 was presented in very general terms. A generic appearance model set Θ was used to describe both foreground and background and the maximum *a posteriori* estimation algorithm was derived according to this generic definition up to the resulting equations for both the expectation step (Eq. 3.19) and the maximization step (Eq. 3.20).

In this section, we describe how to integrate Gaussian mixture models to model foreground and background appearance. The Bayesian variational inference technique presented earlier is used to estimate color models for foreground and background. First, we update the probability density function in sample labeling equations Eq. 3.4 and Eq. 3.5. The density functions corresponds now to the mixture of Gaussian distributions. Each color value I_i^s is predicted according to

$$p(I_i^s|\Theta_i, k_s) = \begin{cases} \sum_{l=1}^{L} \pi_{i,l}^B \mathcal{N}(I_i^s|\mu_{i,l}^B, \Sigma_{i,l}^B) & \text{if } k_s = \mathrm{b}_i, \\ \sum_{l=1}^{L} \pi_l^F \mathcal{N}(I_i^s|\mu_l^F, \Sigma_l^F) & \text{if } k_s = \mathrm{f}, \\ \sum_{l=1}^{L} \pi_{i,l}^{\text{Int}} \mathcal{N}(I_i^s|\mu_{i,l}^{\text{Int}}, \Sigma_{i,l}^{\text{Int}}) & \text{if } k_s = \mathrm{b}_j \text{ and } i \neq j, \end{cases} \tag{4.20}$$

where each density is estimated according to the L components. Each component follows a Gaussian distribution $\mathcal{N}(x|\mu_{i,l}, \Sigma_{i,l})$ with mean value $\mu_{i,l}$ and covariance matrix $\Sigma_{i,l}$. The proportions of the components in the mixture are $\pi_{i,l}$.

The parameters related to the general image models, $\mu_{i,l}^{\text{Int}}$, $\Sigma_{i,l}^{\text{Int}}$ and $\alpha_{i,l}^{\text{Int}}$, are constant. They are estimated once in a preprocessing step following the variational inference described in the previous section. However, the foreground and background models have to be estimated at each iteration. We posit the existence of a label for each observation, whose value informs us which components generated the considered observation. Using the variational framework, updating color models can be shown to be similar to the steps of the previous variational inference for Gaussian mixture model. The difference lies in the extra terms corresponding to sample label probabilities and the bias from known background pixels.

Background color models The objective is to find for each view i the new set of parameters that maximizes the corresponding term in Eq. (3.20):

$$G_i = \sum_{s \in \mathcal{S}} p_s^{\mathrm{b}_i} \log p(I_i^s|\Theta_i, k_s = \mathrm{b}_i) + \lambda_i \sum_{p \in R_i^{\text{Ext}}} \log(p(I_i^p|\Theta_i^B)). \tag{4.21}$$

If we note by x_s^i the color information associated to each sample s in view i ($x_s^i = I_i^s, \forall\, s \in \mathcal{S}$) and component parameters for the background model of view i as $\mu_l^i = \mu_{i,l}^B$ and $\Lambda_l^i = (\Sigma_{i,l}^B)^{-1}$, then using Bayesian estimation results in the same equations for the optimization over the labels as derived in Eq. (4.7), replacing only the general data variables like x_s and z_{sl} by their view dependent equivalent x_s^i and z_{sl}^i.

The only difference appears at the update step for color models that accounts for sample background probabilities. More precisely, it will affect the quantities defined in Eq. (4.16) as follows:

$$\begin{aligned} N_l^i &= \sum_s p_s^{\mathrm{b}_i} r_{sl}^i + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} r_{pl}, \\ \bar{x}_l^i &= \frac{1}{N_l^i}\Big(\sum_s p_s^{\mathrm{b}_i} r_{sl}^i x_s^i + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} r_{pl} I_i^p\Big), \\ S_l^i &= \frac{1}{N_l^i}\Big(\sum_s p_s^{\mathrm{b}_i} r_{sl} (x_s^i - \bar{x}_l^i)(x_s^i - \bar{x}_l^i)^T + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} r_{pl} (I_i^p - \bar{x}_l)(I_i^p - \bar{x}_l)^T\Big), \end{aligned} \tag{4.22}$$

with $r_{pl} = \mathbb{E}[z_{pl}|\text{all the rest}]$ which is the expected value of the binary variable z_{pl} indicating if the pixel p is generated by the component l.

Foreground color models As for background models, the objective is to find the new set of foreground parameters that maximizes the corresponding term in Eq. (3.20):

$$F = \sum_i \sum_s p_s^{\mathrm{f}} \log p(I_i^s | \Theta_i, k_s = \mathrm{f}). \tag{4.23}$$

Using the same simplified notation as for the background model estimation, we end up with the same equations for the foreground case. The optimization over the labels results in the same equation as in the general case (Eq. (4.7)), but the optimization over color model induces the following changes in the quantities defined in Eq. (4.16):

$$\begin{aligned} N_l &= \sum_i \sum_s p_s^{\mathrm{f}} r_{sl}^i, \\ \bar{x}_l &= \frac{1}{N_l} \sum_i \sum_s p_s^{\mathrm{f}} r_{sl}^i x_s^i, \\ S_l &= \frac{1}{N_l} \sum_i \sum_s p_s^{\mathrm{f}} r_{sl} (x_s^i - \bar{x}_l)(x_s^i - \bar{x}_l)^T. \end{aligned} \tag{4.24}$$

These equations show that in both cases, foreground and background model update, the importance of color data information is proportional to the probabilities obtained at n-tuple level. We also note the participation of known background pixels inducing a bias toward a model explaining these image regions.

4.2.3 Results using Bayesian Gaussian mixture model

In this section we present segmentation results of the proposed variational approach on the multi-view datasets. The objective is to validate the extension of the proposed sparse sampling framework to more complex models. We also compare the variational approach with the histogram based method presented in Chapter 3. We use the same datasets: *Arts Martiaux*, *Bear*, *Couch*, *Buste*, *Bike* and *Car*. In the previous chapter, we used histograms of size 32^3 on the HSV color space. Here we use the variational GMM approach with 500 components on the

Lab color space. We use the same experimental protocol presented in chapter 3. Samples in $\mathcal{S}$ are drawn from the common visibility domain of all cameras. This defines a bounding volume which is used to define regions R_i^{Int} in each image i and to find a first set of background pixels. We used a regular 3D sampling with about $N = 50^3$ samples. All the labels are set to the same probability for all the samples and we start the iterative process by a maximization step. We run our algorithm on a 2.3 GHz Intel Xeon PC with 32GB RAM, using a sequential C++ implementation.

Qualitatives observations. The main advantage of using Gaussian mixture models is to have a better approximation of distribution functions. In histogram based models, the bin discretization stops information propagation between similar color if they belong to different bins. This is not the case, with Gaussian distributions. The main improvement we expect from the new model using variational GMMs is a better propagation of background color information. This is verified in most of the datasets as is illustrated in figure 4.3.

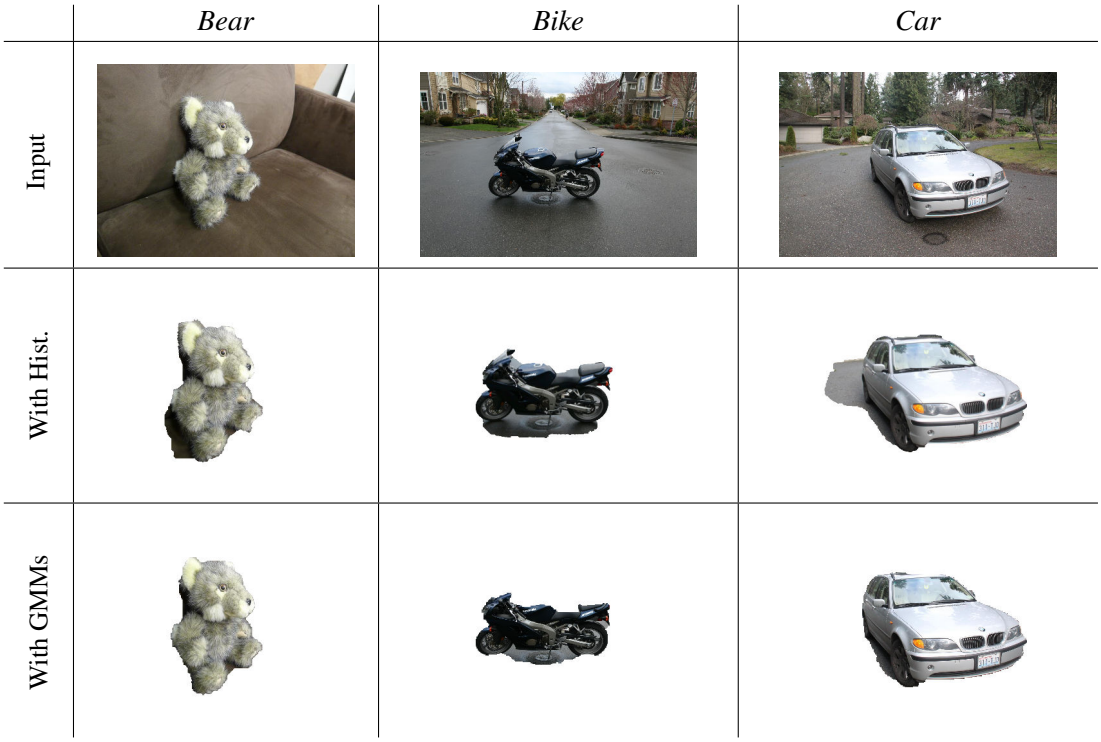


FIGURE 4.3: Comparison between GMMs and histogram color models for multi-view segmentation with 3D samples: we can see the advantages of using GMM as background propagation is more effective. However, this can result in under segmentation of the foreground object as for the bike.

In figure 4.3, we can see that background information is better propagated with GMMs as color models. On the *Bear* dataset, the shaded region of the couch near the teddy bear is now correctly segmented as background. We observe a similar result on the *Bike* and *Car* datasets, where the ground regions are more accurately labelled. In general, this behavior of GMMs is beneficial to the segmentation when there is a clear difference in foreground and background general appearances. However, with more complex scenes (*Bike* dataset for example), the foreground object is usually under-estimated. This can be observed on the segmentation image of the *Bike* (see

Dataset	Mean Error (%)		Hit Rate (%)		False Alarms (%)	
	GMMs	Histograms	GMMs	Histograms	GMMs	Histograms
Bust	0.4 ± 0.2	$\mathbf{0.2 \pm 0.1}$	98.7 ± 0.9	$\mathbf{99.4 \pm 0.01}$	$\mathbf{0.2 \pm 0.1}$	0.7 ± 0.3
Arts Martiaux	0.6 ± 0.4	$\mathbf{0.5 \pm 0.2}$	97.4 ± 0.7	$\mathbf{97.5 \pm 0.3}$	$\mathbf{0.4 \pm 0.46}$	2.7 ± 1.4
Couch	$\mathbf{0.6 \pm 0.1}$	1.2 ± 0.8	$\mathbf{98.7 \pm 0.2}$	97.0 ± 2.8	0.2 ± 0.1	$\mathbf{0.1 \pm 0.1}$
Bear	$\mathbf{1.9 \pm 0.4}$	2.7 ± 1.5	91.8 ± 2.2	$\mathbf{94.5 \pm 3.0}$	$\mathbf{0.2 \pm 1.9}$	7.0 ± 9.0
Car	2.8 ± 1.4	$\mathbf{2.8 \pm 0.8}$	93.9 ± 2.3	$\mathbf{98.8 \pm 0.8}$	$\mathbf{2.2 \pm 1.9}$	16.7 ± 8.8
Bike	$\mathbf{1.9 \pm 0.4}$	2.4 ± 1.1	77.9 ± 10.0	$\mathbf{96.7 \pm 2.1}$	$\mathbf{0.3 \pm 0.4}$	25.0 ± 13.3

TABLE 4.1: Full evaluation of the proposed approach on the different datasets. Best results are indicated in bold font. The GMM based approach produces better segmentation results. It particularly allows a smoother propagation of color information than when using histograms.

Fig. 4.3) where the bottom part of the tires is incorrectly labeled as background. This is further verified and discussed in the quantitative evaluation.

Quantitative evaluation. In this part, we perform the full evaluation of multi-view segmentation using GMMs as color models. We use the measures presented in Chapter 3: *Mean Error* (Eq. 3.34), *Hit Rate* (Eq. 3.35) and *False Alarms* (Eq. 3.36). For the comparison, we used all the views and the same number of samples. The results are summarized in Table. 4.1. For each dataset and measure, the best result is indicated in bold.

The quantitative evaluation is in line with the previous qualitative observations. Segmentation using GMMs as appearance models allows a better propagation of background regions and, as expected, performs best on the *False Alarms* measure. In general, both GMMs and histograms achieve close values on the *Hit Rate* measure except on complex datasets like *Bike* and *Car*. Here the histogram version performs better but only because larger regions around the object of interest are incorrectly segmented as foreground. To summarize, multi-view segmentation using GMMs achieves similar or better results than the histogram version (see *Mean Error* in table 4.1). The method is, however, similarly sensitive to appearance ambiguities between foreground and background objects.

Importance of using many components. Usually, the critical point in using GMMs is the appropriate selection of the number of components. Using a variational approach alleviates the need to tune this parameters for each dataset. Given any number of components, the method automatically adapts the model parameters to best approximate the probability distributions, avoiding singularities. The *Arts Martiaux* dataset is a good illustration of this (Fig. 4.4). Despite the relatively limited colors present in the scene, the method is not able to achieve good segmentation results with 50 Gaussian components. To avoid preselecting the number of components to use for each dataset, the method is initialized with a large number (500) that is the same for all datasets, achieving better results (See Fig. 4.4).

If the segmentation results are of better quality using GMMs, this comes at a cost. The processing time and the amount of memory needed to process GMMs updates is very large. With our unoptimized sequential c++ implementation several hours are needed to process each dataset. Clearly, the segmentation results obtained do not justify the increased computation. Moreover, we lose the main advantage of using a sparse sampling approach.

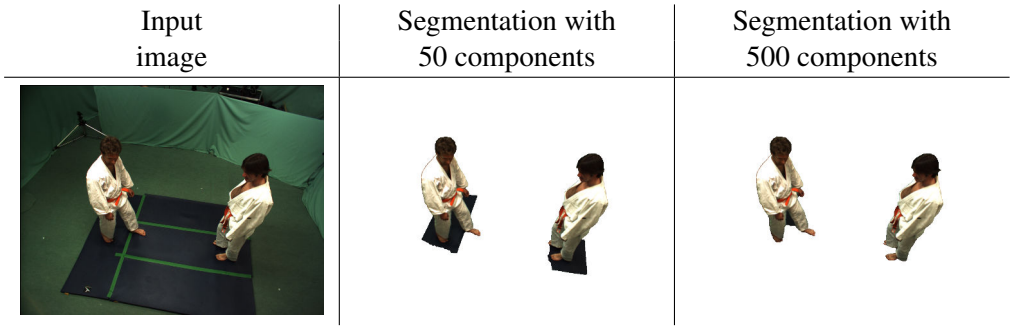


FIGURE 4.4: Using the appropriate number of components is fundamental to obtain a good approximation of the probability density functions for foreground and background models. Using a variational approach, it is possible to set a high number of components and the method automatically adapts the parameters to best approximate the probability density function, avoiding singularities.

4.2.4 Discussion on using GMMs

We have seen in this first part, how the model can be extended to GMMs using a variational approach. Segmentation results are improved thanks to the smooth nature of Gaussian models, contrary to the histogram version. However, the computation time is greatly increased, leading to hours of processing time for all the datasets. This is due to the extra iterative process needed to update color models, adapting the components to the new probability values.

The limited improvement in segmentation results does not justify the huge computations needed by this more complex model. These results motivated the exploration of other means of representing complex appearance models that would be compatible with the low complexity of the sampling approach. They also motivate the extension of the sparse sampling approach to the case of range cameras, presented in the following sections.

4.3 Multi-view Segmentation with range cameras

In recent years there has been a fast growing interest in hybrid multiple camera systems using depth information alongside color, thanks to the emergence of new sensors such as Time of Flight (ToF) cameras and affordable new mass-produced technologies such as Kinect. Depth sensors have been used in conjunction with color imaging in monocular setups [117, 118]. This raises a number of interesting questions. Given the availability of affordable multi-view, multi-modal platforms, can we make good use of the depth modality to improve multi-view segmentation? Can we use multiple depth and color cameras from different viewpoints to improve over monocular depth-color or color only segmentation? One could expect the different modalities to be complementary and compensate for each other's defects, e.g. enabling to disambiguate ambiguous color regions.

In this section we try to address these questions by extending the sparse multi-view consistency model to handle depth information, enabling us to test different expressions of the modality fusion and quantify each. We demonstrate, in §4.3.3, how depth can be included in the generative

model, to provide fast elimination of 3D regions that appear empty according to the depth sensor. We then show various experiments quantifying the benefits of using depth information, in conjunction with different color representation possibilities. We also show the improvement of using multi-view depth-color segmentation with respect to a state-of-the-art monocular depth-color segmentation approach (§4.3.4).

4.3.1 Related work

A number of works have investigated using several depth cameras in conjunction with color cameras, targeting improvements in 3D surface reconstruction. Guan *et al.* [119] propose a probabilistic framework to volumetrically fuse depth and silhouette information. Kim *et al.* [120] propose a similar approach, this time adding a photoconsistency term, with the goal of improving multi-view stereo with depth information. Although in some cases it might be possible to extract segmentations from the surface representation, clearly the aim of these works is 3D reconstruction: they do not specifically address how the different depth and color cues could be combined directly and efficiently for the purpose of reliably improving the segmentation in all views, as proposed in this thesis.

As opposed to the multi-view case, single-view segmentation has been investigated in several works, which consider depth measurements in conjunction with color measurements, to address foreground-background color ambiguity, lighting changes and occlusion. This was first experimented with stereo data [117], optimizing segmentation with color and depth region discontinuity terms. The technique obtains depth from binocular stereo, but similar techniques emerged for depth imaging. For complete automation, the methods usually need to introduce additional information defining the object of interest. A typical assumption is to identify front-most depth regions as the object of interest, for simple extraction scenarios such as videoconferencing. This can be enforced using a depth threshold [121], or by considering two depth distributions for foreground and background [118]. This implies that the methods only work for good depth separation. In [122] user input is required and a segmentation manually defined in the first and the last frame, is propagated using information from different modalities (e.g. range cameras, thermal camera).

To the best of our reading, no method has yet considered to model the case of multiple depth and color cameras for the segmentation problem. We will show that in this case, only weaker priors are needed to isolate objects of interest automatically, enabling to address more general setups. In particular, we are able to extract segmentations even when there is no clear depth separation between foreground and background - typically when the object of interest rests on another flat object such as a table (§4.3.4). Also, as opposed to monocular setups [118], the depth and color cameras need not be placed or rectified to a single viewpoint: our model deals with arbitrary depth and color camera setups.

4.3.2 Principle

Range cameras provide measurements of the distance between the sensor and the scene objects. Even though these measurements are often noisy and sometimes locally inaccurate, they are informative of space occupancy in the scene and, in that respect, provide useful cues for the classification of samples in the presented framework.

Consider a sample s and a given range camera. Let d_s be the known ground truth distance of s to the center of this camera, z_s the depth provided by the range sensor in corresponding direction and d_{max} the maximum depth range. As identified by Guan *et al.* [119], the information that can be inferred on the occupancy of s by a foreground object depends on the relative values of d_s and z_s (See Fig. 4.5):

- $z_s > d_s$: the sample is not occupied and should be classified as background.
- $z_s \simeq d_s$: this configuration corresponds to the highest probability that the sample is occupied by a foreground object.
- $z_s < d_s$: the sample lies behind an occluding object on its line of sight, nothing can be inferred about its occupancy.

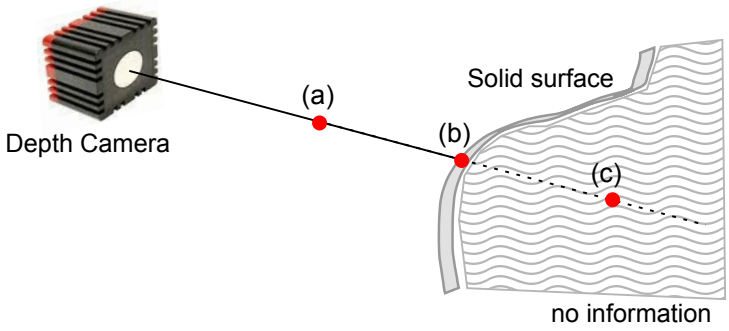


FIGURE 4.5: Depth sampling situation on one projection line. Red dots indicate possible sampling positions: (a) Depth measure is higher than sample distance to the camera ($z_s > d_s$). (b) Depth measure corresponds to sample position ($z_s \simeq d_s$). (c) Depth measure is lower than sample distance ($z_s < d_s$).

These considerations should drive the choice of the space occupancy model from the depth observations and in some configurations, we may want to modulate the behavior of the algorithm in order to better adapt to the semantics of the scene. Consider for instance a scene with a can lying on a table. Though the table is a solid object inside this common visibility volume, because a significant part of the table lies outside of the common viewing volume, we may want the algorithm to single out only the can as foreground (See Fig. 4.9(a) for an example). We will show in the next sections that our modeling framework allows us to force either of these behaviors (select the table as foreground or background), by selecting an appropriate probabilistic model for space occupancy from the depth observations.

4.3.3 Depth-sensor enabled model

Let m be the number of depth maps and z_j^s the depth information associated with sample s in depth map j. We propose a new graphical model (Fig. 4.6), where the color tuple $I_{1:n}^s$ and the depth reading vector $z_{1:m}^s$ of each 3D sample $s \in \mathcal{S}$ are predicted according to its classification label k_s with priors π_k and the global color models Θ_i.

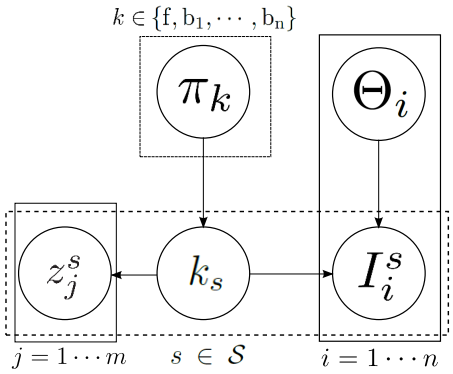


FIGURE 4.6: Graphical model for color and depth: the generative model for color I_i^s and depth observations z_j^s is conditioned on the samples labels k_s.

Posterior distribution. Given the model, our goal is to find the parameters that maximize the *a posteriori* density given the observations. Noting $Z = \{z_{1:m}^s\}_{s \in \mathcal{S}}$, the likelihood function (Eq. 3.2) becomes:

$$\begin{aligned} \mathcal{L}(\Theta, \pi | I, Z, K) &= p(I, Z, K | \Theta, \pi) \\ &= \prod_{s \in \mathcal{S}} p(k_s, I_{1:n}^s, z_{1:m}^s | \Theta, \pi), \end{aligned} \tag{4.25}$$

with:

$$p(k_s, I_{1:n}^s, z_{1:m}^s | \Theta, \pi) = p(k_s | \pi) \prod_{i=1}^{n} p(I_i^s | \Theta_i, k_s) \prod_{j=1}^{m} p(z_j^s | k_s), \tag{4.26}$$

where color distributions $p(I_i^s | \Theta_i, k_s)$ are defined by (3.21) and depth distributions $p(z_j^s | k_s)$ will be defined below.

Estimation. We follow the same EM scheme as in §4.2 to solve this MAP problem with latent variables. The main difference will appear at the expectation step (Eq. 3.19), where the new update of sample label posterior includes depth information:

$$\forall k \in \mathcal{K}, p(k_s = | I_{1:n}^s, z_{1:m}^s, \Theta_i^g) = \frac{\pi_k^g \prod_{i=1}^{n} p(I_i^s | \Theta_i^g, k_s = k) \prod_{j=1}^{m} p(z_j^s | k_s = k)}{\sum_{\ell \in \mathcal{K}} \pi_\ell^g \left[\prod_{i=1}^{n} p(I_i^s | \Theta_i^g, k_s = \ell) \prod_{j=1}^{m} p(z_j^s | k_s = \ell) \right]}. \tag{4.27}$$

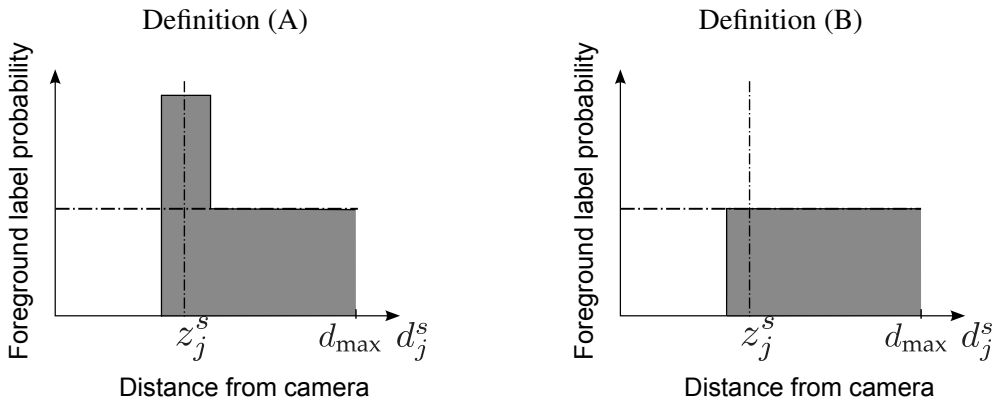


FIGURE 4.7: Probability for a sample s to have a foreground label in the two situations A and B (See text for details). This probability depends on the sample distance from the camera d_j^s, the depth measure z_j^s and the maximum depth range $d_{\max}$.

The maximization step does not change (*i.e.*, equations 3.27 and 3.28). In the expectation step, terms related to the depth measurement act as priors on the samples labels. These terms can be computed once and for all in the initialization stage.

Modeling for depth. Following the principles given in §4.3.2, our depth sensor model needs to classify as background all samples lying between the camera and the depth measurement along each line of sight and shouldn't give any information about samples behind front objects. This is expressed by giving, conditioned on the f label, 0 probability to samples whose depth verifies $d_j^s < z_j^s - \varepsilon$, where ϵ is a conservative depth noise threshold. The sensor model behavior must also be defined when the position of the sample coincides with the depth measurement provided by the range camera. Different modeling possibilities exist and we empirically define distributions for two cases, following [119] and [120]:

(A) *Regions of space around the measured depth are more likely to contain an object than regions further away:*

$$p(z_j^s|k_s=\mathrm{f}) = \begin{cases} 1/d_{\max} & \text{if } z_j^s < d_j^s - \varepsilon, \\ (d_{\max} - d_j^s + \varepsilon)/(2\varepsilon d_{\max}) & \text{if } |d_j^s - z_j^s| \leq \varepsilon, \\ 0 & \text{if } z_j^s > d_j^s + \varepsilon. \end{cases} \tag{4.28}$$

(B) *Regions further away than the measured depth are equally likely to contain an object:*

$$p(z_j^s|k_s = \mathrm{f}) = \begin{cases} 1/(d_j^s + \varepsilon) & \text{if } z_j^s < d_j^s + \varepsilon, \\ 0 & \text{otherwise.} \end{cases} \tag{4.29}$$

For a background label b_i, the depth measurement is not informative and does not depend on these definitions:

$$p(z_j^s|k_s = \mathrm{b}_i) = 1/d_{\max}. \tag{4.30}$$

To give an intuition of what happens with these two different models, let's consider the simple situation of one depth camera, with all the priors and mixture coefficients set to uniform and a sample s, with its associated depth measure z_j^s. Fig. 4.7 shows the probability $p(k_s = \mathrm{f}|z_j^s)$ for this sample to have the foreground label according to its actual distance (d_j^s) from the camera in the two situations.

It turns out that the first choice (Eq. 4.28) is detrimental to the segmentation performance when parts of the background are close to the foreground object, as illustrated on Fig. 4.8. In this case, samples on the table will have a high probability to be assigned foreground labels, owing to depth measurements. The colors at their projections in the views will be integrated in the foreground color model and will never be considered as background, despite the presence of similar colors in the known background region R_i^{Ext}. Consequently, in the sequel we will use the second model (Eq. 4.29).

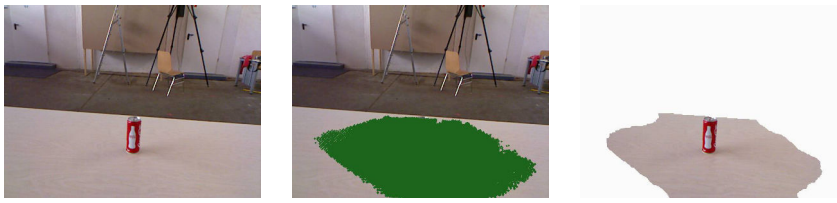


FIGURE 4.8: Regions around the measured depth are assumed to be foreground objects (Eq. 4.28). Green points are projection of samples with $p_s^{\mathrm{f}} > 0.8$.

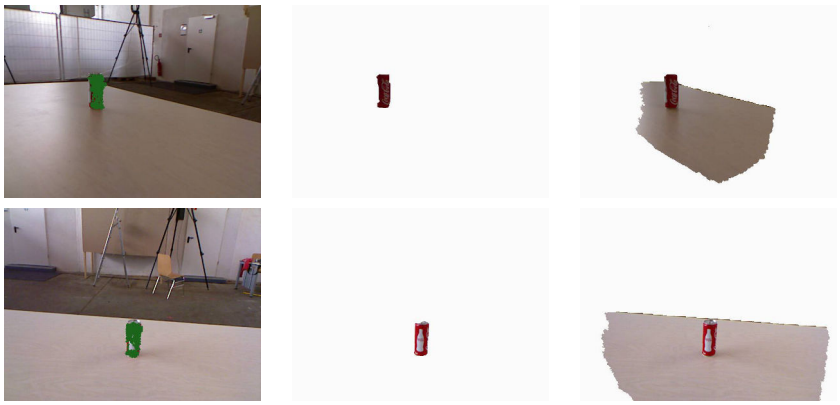
4.3.4 Evaluation of depth contribution

In this section, we show how the proposed method behaves in a multi-view context including depth cameras and how depth and color observations influence the results.

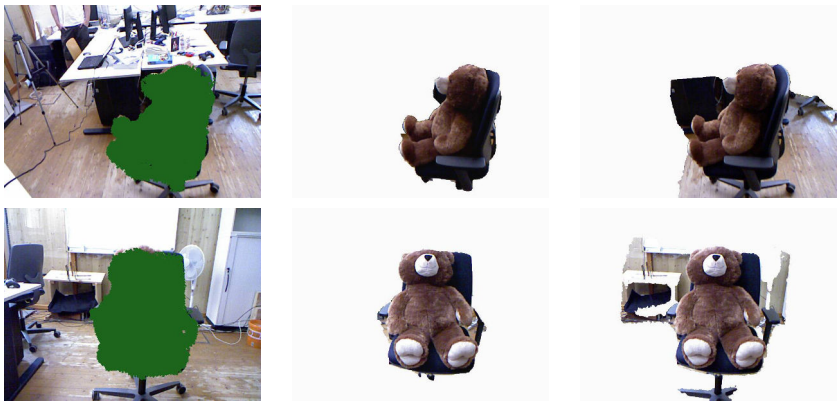
We choose to run tests on two different datasets. The first dataset was captured using a multi-view system consisting of $n = 2$ color cameras and $m = 2$ Swiss Ranger SR-4000 time-of-flight cameras (Fig. 4.10). The second dataset consists of three different Kinect video sequences[1]: *Coke*, *Plant* and *TeddyBear*. We select up to ten different frames from each sequence to constitute multi-view data-sets (see Figs. 4.9 and 4.11). We use the color model described in the previous section (except for the *Plant* dataset, see the related discussion for details).

Comparison with a monocular approach. The results obtained by our approach on the considered datasets are shown on Figs. 4.9 and 4.10. An average value of 5 iterations of the EM was needed to reach convergence. The method adapts well to the various configurations. We compare our method with TofCut [118], a state-of-art monocular approach that describes foreground and background pixels using a weighted combination of color and depth models. The original algorithm was designed for datasets with a good discrimination in depth between foreground and background. In order to improve its robustness to overlapping depth distributions, we kept the same models but adopted an iterative approach. Initially, pixels inside the region of

[1] http://vision.in.tum.de/data/datasets/rgbd-dataset



(a) *Coke* dataset - 10 views



(b) *TeddyBear* dataset - 8 views

FIGURE 4.9: Results on Kinect datasets. Green dots indicate projections of samples labeled foreground with $p_s^{\mathrm{f}} > 0.8$. Second column is our segmentation and the third column shows results with our implementation of *TofCut* (see text for details).

interest R_i^{Int} are set to foreground, all other pixels are labeled background. Next, we alternate between pixel relabeling and model update, and iterate until convergence. Unlike the original TofCut algorithm, we choose to model foreground and background appearance and depth using histograms. This is of particular interest for depth, where the discrimination between foreground and background is not as strong as in [118]. Comparative results show that our approach consistently outperforms the modified TofCut method. Fig. 4.9(a) illustrates a typical failure case for monocular approaches such as TofCut, when the discrimination in depth between foreground and background is poor. Our method, in contrast, successfully handles the depth ambiguities, owing to the integration of information from all the views.

Our approach is also able to handle more complex acquisition situations where color and depth cameras are not aligned, for which depth-based monocular approaches would not be applicable. Owing to the 3D samples, information is propagated from depth cameras to color cameras. To be fair, if color and depth cameras were aligned, a monocular approach would certainly work for this dataset. Nevertheless, fig. 4.10 shows the ability of our method to handle more complex acquisition configurations.

A more complex scenario - case study The difference in appearance between foreground and background is a key assumption of our approach. Not meeting this condition will lead

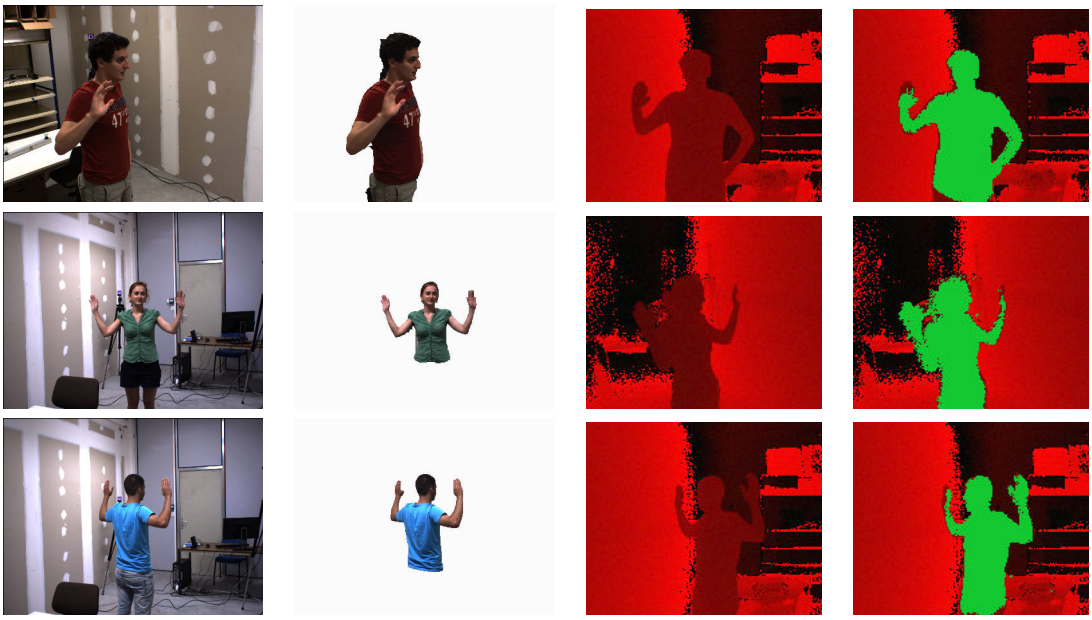


FIGURE 4.10: Results on a multi-view dataset including $n = 2$ color cameras and $m = 2$ ToF cameras. On the left side: image and foreground segmentation in one the views. On the right side: ToF depth image and projections of foreground samples at convergence.

to segmentation errors. With the *Plant* dataset, there is an ambiguity between foreground and background both in color and depth. If we define foreground to be any solid object inside the common visibility volume (using the model defined by equation 4.28), then the table will be segmented as foreground. If we add the assumption that the foreground must be visually different from the background, then only the green leaves are segmented as foreground and the blue pot is identified as background due to the blue objects in the background. In order to obtain a semantically meaningful segmentation, we use localized histograms to limit the propagation of background labels: the image is subdivided in rectangular regions, each region having its own histogram. Pixels participate in the histograms of the 4 closest regions. In this case, the combination of color and depth cues yields the expected segmentation (Fig. 4.11).

This dataset illustrates the complexity of the multi-view segmentation problem. It shows that color information is not enough to perform segmentation in complex scenarios and that depth information must be used with caution. This result also shows the adaptability of the n-tuple approach to various appearance models, which may be selected to better suit the desired segmentation semantics.

Quantitative results. We perform a quantitative comparison with ground truth to see how the combination of the two sources of information influences the results (Fig. 4.12). We compute false alarm and hit rates on segmentation results in three scenarios: color only, depth only and combining both.

The color only method is sensitive to the resemblance between foreground and background. This explains the results on the Kinect datasets. It is also sensitive to the number of cameras. With only two color cameras, results on the second dataset are not representative and are therefore not shown on the figure. Results using only depth information are of better quality. The

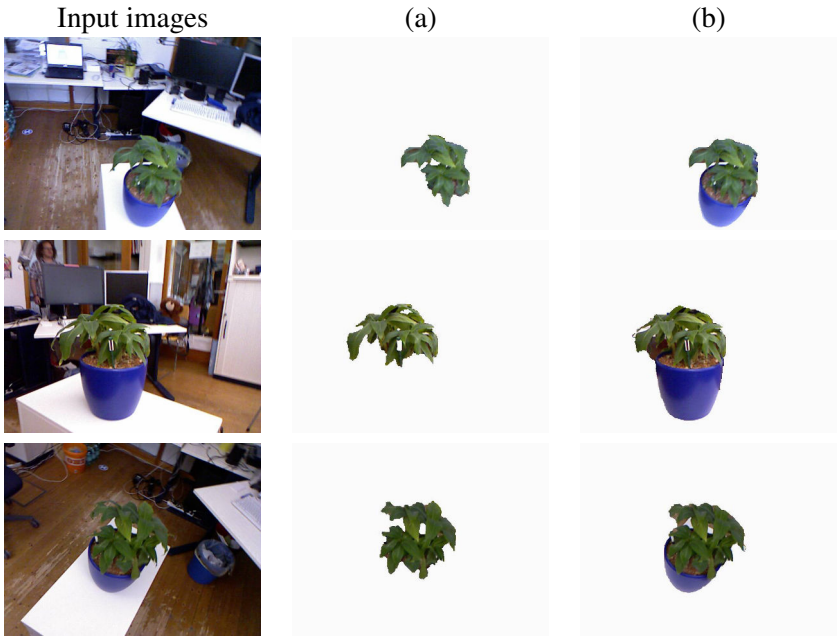


FIGURE 4.11: Segmentation results on *Plant* dataset: (a) with a shared foreground color histogram; (b) with local histograms.

results on the *TeddyBear* dataset are very close to ground truth because all solid objects in the common visibility volume are parts of the foreground. However, using depth-only on the *Plant* dataset fails to correctly segment the table. Combining depth and color is very effective in these scenarios, where depth allows a quick identification and elimination of background regions and color allows a better accuracy of the foreground segmentation.

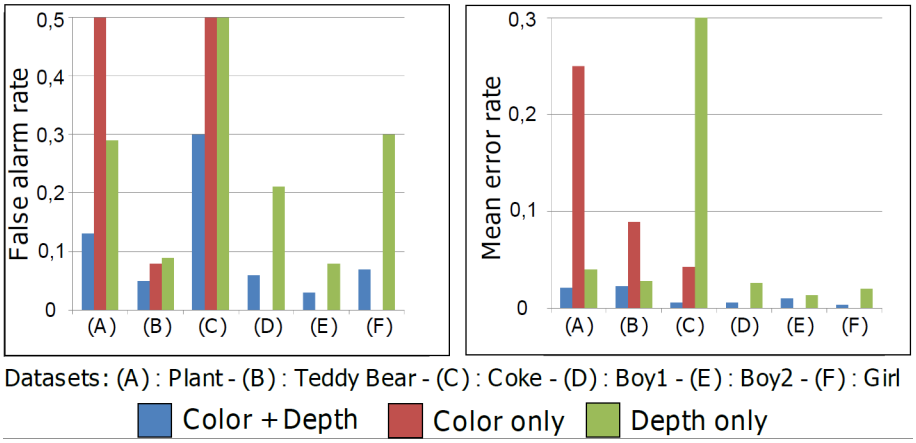


FIGURE 4.12: Quantitative results: Comparison with ground truth in three configurations: depth only, color only and combining depth and color. Error rate gives a global measure for segmentation errors. False alarm rate indicates the proportion of background segmented as foreground. "Color only" is not given on last three sequences due to their insufficient number of image viewpoints ($n = 2$).

4.3.5 Discussion on using depth information

The n-tuple framework was extended to a different modality integrating data from depth and color cameras and experiments show the efficiency of the method, including in difficult test scenarios where objects of interest are not straightforwardly distinguishable in depth images, and lean on background objects. As experiments show, the method can deal with flexible viewpoint configurations. Best results are obtained with depth cameras at a maximal relative angle of 90 degrees. Further studies could be conducted to quantify the influence of depth and color viewpoints, but these results already confirm the advantages in using our generative model with multi-modal cues for improvements. Thanks to additional depth information, the method is quite successful and resilient, more so than using color alone. The depth extension also enables some progress in understanding how far such a weak and generic model of viewpoint correlations can take us to solve the multi-view segmentation problem. Some failure cases still appear, when color and depth happen to be simultaneously undiscriminating, or when objects of interest do not fit the initial assumptions of the model (is the flower pot part of the object of interest?). Still, the method is largely successful despite its use of weak interview cues, with no priors other than geometric.

4.4 Conclusion

In this chapter we show how the n-ntuple framework can be adapted to make use of different appearance models, with the GMM example, and other modalities, like depth information from range cameras. This chapter is extremely helpful in understanding the limitations of the current method. Using a more complex color only appearance model or resorting to other modalities improves the segmentation results but does not provide the important gain one would expect. This clearly shows that the model proposed so far is missing some aspects of the problems. The results in this chapter hints toward different directions for improvements. The first is that color only appearance models are not sufficient to distinguish between foreground and background, even in multi-view setups with information fusion. The second is that despite the geometric consistency expressed in the n-tuple model, the method is still not fully taking advantage of the multi-view setup. Finally, using information from videos should considered. In particular using motion to identify and segment objects of interest. These findings are important and they are the motivation behind the multi-view graph-cut model proposed in the following chapter.

Chapter 5

Multi-view graph cut for object segmentation

5.1 Introduction

In the previous chapters, a multi-view segmentation method based on sparse sampling of the space was presented. Despite the sound probabilistic model it exploits, the quality of the results in challenging outdoor situations suggests that the method is still not entirely taking advantage of all the information present in a multi-view setup. In particular, the method is sensitive to similarity of appearance between foreground and background regions, which proves that color only appearance models are not discriminative enough to identify and segment the foreground object(s). This problem of limited differentiation power of color only models is also encountered in monocular segmentation and some methods suggest to reason at superpixel level rather than on pixels [123]. Reasoning on theses local clusters of similar pixels [124, 125] (i.e. *superpixels*) permits a richer region description (using texture information for example) and makes possible to discriminate between two image regions despite their color similarity. Many research works have explored this direction both in image [25, 126–128] and video segmentation [129, 130] with promising results. In the multi-camera context only Campbell *et al.* [37] proposed a method based on superpixels but this is only used with the objective of reducing computation time (texture information is not used for example). Indeed, their method relies on epipolar geometry to relate segmentation between the views, which implies a point to line relation and reasoning on pixels would quickly raise combinatorial issues.

In this chapter we propose a new iterative formulation of multiple view object segmentation that is using a joint graphcut linking pixels through space and time. This formulation is clearly related to some methods in cosegmentation [75] and multi-view segmentation [37]. However, it differs by the graph coupling that our framework introduces at the geometric level which is based on the 3D samples. This method brings several key contributions, validated in §5.6: first, it is noticeably efficient in convergence and computational requirements, using only sparse interview links. Second, the graph structure intrinsically takes into account the constraints of

the multi-view segmentation problem, producing coherent segmentations both at the geometric level and the image level. Third, the ability to handle few viewpoints, much further apart than most state of the art approaches require. This is a situation that naturally arises in practice and for which none of the previous works is giving results below 6 viewpoints. Fourth, in the video case, the framework straightforwardly extends to the use of temporal links with the aim to propagate momentarily reliable segmentation evidences across time in multi-view setups. To the best of our knowledge, this is the first approach to leverage temporal cues for multiple video segmentation, with significant future applications.

5.2 Related work in video segmentation

In video segmentation, ideas developed for image segmentation like contours based approaches or pixel clustering, have been extended to take into account the extra temporal dimension. This implies the necessity to develop algorithms that enforce temporal coherence but opens the possibility of using information from motion to improve segmentation results and alleviate the need for user interaction, particularly in the monocular case.

Among the first work we can cite the natural extension of active contours to videos [131] or trimap based alpha-matting [132] using optical flow. Agarwala *et al.* [133] propose a rotoscoping technique based on user interaction where the tracking algorithm is cast as a space-time optimization problem that solves for time-varying curve shapes based on an input video sequence and user-specified constraints. In [134], Li *et al.* first segment video frames into watershed region partitions with temporal coherence, then foreground/background region is propagated between two frames where user indicates the object to segment. This segmentation is locally refined using color models and edge preservation. Similarly, Wang *et al.* [135] use a preprocessing step based on mean-shift to obtain a small number of regions on which a min-cut based algorithm operates. In other methods [136], user interaction is propagated using geodesic distance without complex estimation of optical flow. Using a hierarchy of tobogganed regions [137], an interactive segmentation tool is proposed with seed points that can be placed at different voxels in the volumetric video representation. Video SnapCut [6] is a video segmentation method based on a set of localized classifiers on the object borders that locally assign pixel foreground/background probabilities according to estimated color models and object shape. Multi-frame propagation is achieved by moving the classifiers according to optical flow, and topological changes are handled by adding or removing classifiers when needed. It is also possible to model self-occlusions and dis-occlusions in a joint shape and appearance tracking framework [138].

Instead of relying on user interaction, some other methods try to learn some statistics on the background. This learning process is usually performed on empty images of the scene in a preprocessing step. If segmentation is possible using only these learned statistics, usually background subtraction techniques try to update color models [139–141] in order to adapt to possible variations in the scene, due to different illumination and shadows. In a more recent recent work, Criminisi et al. [142] propose to learn low-level statistics using a Hidden Markov Model. Pixels can have foreground or background states and the proposed HMM is used to model transitions

from one state to the other according to variations in pixel colors. This requires a learning step, but reasoning on optical flow is not used, which makes it suitable for real time applications such as video conference. Changes of illuminations may also be taken into account as proposed in [143].

Just like image segmentation, some methods aim at grouping pixels into coherent spatio-temporal volumes [144, 145] or using a hierarchical graph-based approach [146]. These regions were used in many segmentation methods to propose temporally consistent segmentations [147] or to propagate user interaction in video segmentation [129].

Using motion to identify the different objects present in a scene is also possible, as was shown by recent results in automatic video object segmentation using motion saliency [148], segmentation proposals [11] or point trajectories [149].

We should also mention the recent results in video cosegmentation like the multi-class video segmentation method proposed by Chiu *et al.* [150], formulating the problem as a non-parametric Bayesian modelling across video sequences.

Multi-view segmentation is addressed here in the static and dynamic cases, and we show in our experiments the benefits of using a multi-view approach through a comparison with Video SnapCut [6].

5.3 Principle

We keep the same definition of foreground used until now, that is, an object of interest should satisfy two constraints: be fully visible in all considered views, and its general appearance should be different from the background's general appearance. In the previous chapters, we have seen that a generative model for the color n-tuples associated with a sparse set of 3D samples can be used to coherently infer foreground and background color models in different views. In this chapter we cast the multi-view object segmentation problem as a joint labelling problem among the n input views, governed by a single MRF energy discussed in §5.4. The driving idea here is that the generative model as presented is still missing some of the problem constraints. The energy based approach we present in this chapter, while building on the notion of sparse 3D sampling, provides a powerful way of representing interactions and relations between the different elements of the problem both at the image level and the inter-view level.

The proposed labelling MRF energy can be described as follows: First, in order to ensure inter-view propagation of segmentation information, we use a sparse set of 3D points (or samples) randomly picked in the region of interest (common field of view of all the cameras) to provide geometric consistency between the images. Each sample is used to create links in the graph between itself and pixels at its projection, whose strength reflects the object coherence probability of the sample. Second, to ensure efficient intra-frame propagation, we compute a superpixel oversegmentation of each image, and define two neighborhood sets on each superpixel in the graph based on image-space and texture-space proximity. Resorting to superpixels also allows

FIGURE 5.1: Multi-view object segmentation (MVOS) using our method with the 3 wide-baseline views shown only, with no photo-consistency hypothesis and no user interaction.

one to benefit from richer region characterizations reducing color-space ambiguity. Third, the resulting MRF energy is minimized using s-t mincut [151] and resultant segmented regions are used to re-estimate per-view foreground/background appearance models, which are, in turn, used to update 3D sample object coherence probabilities. The method is also easily extended to handle videos by adding temporal links between superpixels using tracked interest points or optical flow. We present the details of each stage of the algorithm below.

5.4 Formulation

We are given a set of input images $I^t = \{I^{1,t}, ..., I^{n,t}\}$ at instant t. For each image i at t we have the set $\mathcal{P}_i^t$ of its superpixels p. We use superscript t for time for all terms, generally keeping it implicit for concision unless terms from different instants are involved. Segmenting the object in all the views consists in finding for every superpixel $p \in \mathcal{P}_i^t$ its label k_p with $k_p \in \{\mathrm{f}, \mathrm{b}\}$, the foreground and background labels. We denote $\mathcal{S}^t$ the set of 3D samples used to model dependencies between the views at instant t. These points are uniformly sampled in the common visibility volume.

Contrary to the previous model where all the inference was performed on the n-tuples, the proposed MRF energy tries to label superpixels and the 3D samples only act as geometric constraints. Computing foreground and background label probabilities is performed as an intermediate step before the energy optimization and it is estimated following equation (3.19). More details are given when the corresponding energy term is presented.

5.4.1 MRF Energy Principles

Given the superpixel decomposition (shown in Fig. 5.2 on two examples) and 3D samples, we wish the MRF energy to reward a given labelling of all superpixels as follows, each principle leading to MRF energy terms described in the next subsections.

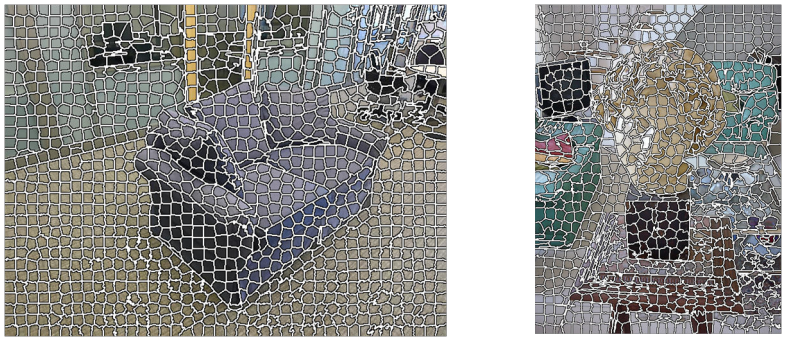


FIGURE 5.2: Example of superpixel segmentation on *Couch* and *Buste* datasets using SLIC [124].

Individual appearance. The appearance of a superpixel should comply with image-wide foreground or background models, depending on its label.

Appearance continuity. Neighboring superpixels likely have the same labels if they have similar appearance.

Appearance similarity. Two superpixels with similar color/texture are more likely to be part of the same object and thus, more likely to have the same label. These superpixels may not be neighbors due to occluding objects.

Multi-view coherence constraints. Here we use the 3D samples to enforce geometric consistency of the segmentations. More precisely, assuming sufficient 3D sampling of the scene, a superpixel should be foreground if it sees at least one object-consistent sample in the scene. Conversely, a superpixel should be background if it sees no object-consistent 3D sample. These links between superpixels and samples are only possible if the 3D samples are themselves included in the MRF. 3D samples are considered object consistent (i.e. foreground) if they project in foreground regions in all the views.

Time consistency. In the case of video data, superpixels in a sequence likely have the same label when they share similar appearance and are temporally linked through an observed flow field (*e.g.* optic flow, SIFT flow).

The principles described here lead to three distinct categories of energy terms. The first category gathers all the intra-view relations which lead to energy terms standard in image segmentation. The second category represents the core contribution of this chapter. It models multi-view segmentation constraints through unary and binary energy terms in an original way thanks to the 3D samples. Finally the third category shows how to correlate segmentation between different instants in the case of videos.

5.4.2 Intra-view appearance terms

The terms presented here correspond to classic energy terms used in image segmentation: unary data and binary spatial smoothness terms defined on superpixels, to which we add non-local appearance similarity terms on superpixel pairs for broader information propagation.

The appearance is modelled using a richer description based on color and texture. Originally [152], this method was proposed with simple texture description based on a limited number of Laplacian and gradient filters. We will try to stay as general as possible in energy term description, postponing the discussion and decision on the appearance models to a later section. For each superpixel p we define its corresponding image region $\mathcal{R}_p$ containing all its associated pixels and each pixel r defines an $(l+3)$-dimension vector that includes both a 3 dimensional color information and an l dimensional texture information. The texture is defined as the response to a bank of Laplacian, gradient and Gabor filters [123, 153].

Foreground and background appearance models are defined on these multi-dimensional vectors and respectively noted Θ_i^F and Θ_i^B.

Individual appearance term. We denote E_c the unary data-term related to each superpixel appearance. We characterize appearance by the sum of pixel-wise log-probabilities of being predicted by an image-wide foreground or background appearance distribution:

$$E_c(k_p) = \begin{cases} \sum_{r \in \mathcal{R}_p} -\log p(I_r^i|\Theta_i^B) & \text{if } k_p = \text{b}, \\ \sum_{r \in \mathcal{R}_p} -\log p(I_r^i|\Theta_i^F) & \text{if } k_p = \text{f}. \end{cases} \tag{5.1}$$

Similarly to the generative model for the color n-tuples proposed in Chapter 3, foreground and background models are defined in very general terms. They can be modelled by histograms or any other more complex distribution. We also note that contrary to the n-tuples based EM inference developed earlier, the foreground model is not necessarily shared between the views. This point will be further investigated in the experimental section.

Appearance continuity term. This binary term, denoted E_n, discourages the assignment of different labels to neighboring superpixels that exhibit similar appearance. It is of the form of a contrast sensitive Potts model [151]. To model this similarity, we use the previously defined texture and color information to create superpixel descriptors. The descriptor for a given superpixel p consists of a histogram on the color and texture information noted A_p.

Let $\mathcal{N}_p^{i,t}$ define the set of adjacent superpixel pairs in view i at time t. For $(p,q) \in \mathcal{N}_p^{i,t}$, the proposed E_n is inversely proportional to the distance between the two superpixel descriptors, as follows:

$$E_n(k_p, k_q) = \begin{cases} \exp\left(\frac{-d(A_p, A_q)^2}{2 < d(A_p, A_q) >^2}\right) & \text{if } k_p \neq k_q, \\ 0 & \text{otherwise.} \end{cases} \tag{5.2}$$

The distance $d(.,.)$ here is the χ^2 distance between the superpixel descriptors. $< d(A_p, A_q) >$ indicates expectation over all neighboring superpixels in the image.

Appearance similarity term. To favor consistent labels and efficient propagation among similar superpixels, we introduce a second binary term E_a of the same form as E_n but defined non-locally. Retrieving for each superpixel p its k-nearest neighbors for the χ^2 distance, defines a set $\mathcal{N}_a^{i,t}(p)$ of superpixel pairs. The set $\mathcal{N}_a^{i,t}$ is defined as follows:

$$\mathcal{N}_a^{i,t} = \bigcup_{p \in \mathcal{P}_i^t} \mathcal{N}_a^{i,t}(p) \tag{5.3}$$

resulting in all the superpixel pairs to be linked by an appearance similarity term:

$$E_a(k_p, k_q) = \begin{cases} \exp(\dfrac{-d(A_p, A_q)^2}{2 < d(A_p, A_q >^2}) & \text{if } k_p \neq k_q, \\ 0 & \text{otherwise.} \end{cases} \tag{5.4}$$

Contrary to the binary continuity energy term (E_n), this similarity term generates links of longer range, hence not limited to direct neighbors.

5.4.3 Inter-view geometric consistency terms

To propagate inter-view information, we use a graph structure connecting a 3D sample to pixels it projects on. While this leads to a structure similar to [75], the latter builds inter-pixel hard links that are always active based on common histogram binning of pixels. In [37], the superpixels of different views are linked using epipolar geometry and the number of links is reduced based on depth binning obtained from stereo.

A key difference in our approach is that linking superpixels though the 3D samples naturally results in a sparse set of inter-view edges, while expressing more subtle and complex multi-view geometry consistency constraints. A difficulty we have to cope with is that geometric consistency of samples may change during iterations because of evolving segmentations. We thus evaluate before each iteration an "objectness" probability measuring consistency with current segmentation, and use it to reweigh the propagation strength of the sample.

The constraints we want to express using 3D samples are the following:

- Labelling a superpixel p as background is only possible if the 3D samples on its line of view are labelled as background.
- To label a superpixel p as foreground it must *see* a foreground 3D sample.

One way to express these constraints is to reason through a deterministic approach on 3D sample label probabilities, by setting a threshold over foreground label probability for example. However, in an iterative inference process where the foreground and background models are still to be estimated, it is a better choice to take advantage of these probabilities by integrating sample foreground/background labelling in the MRF energy, as is described below.

Sample-pixel junction term. The objective of this term is to enforce the following desirable projection consistency property: labelling a superpixel p as background is only possible if it is coherent to label all the samples s projecting on it as background.

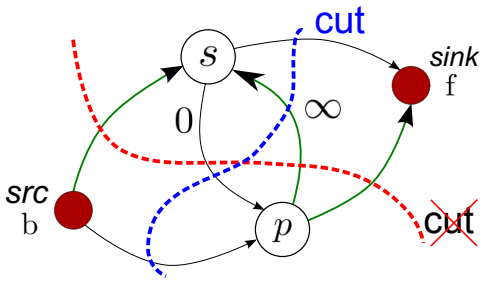


FIGURE 5.3: Relation between samples and superpixels. If a sample s is labelled as foreground then superpixels at its projection positions can not be labelled as background. This corresponds to an impossible cut, as illustrated here.

The desired property is illustrated in Fig. 5.3 with the s-t graph corresponding to the sample-pixel junction energy term. With this term we would like to make it impossible for a cut classifying the superpixel as background while keeping a foreground label for the sample, to be minimal (red cut in Fig. 5.3). On the other hand, a cut classifying the sample as background while keeping a superpixel it projects on as foreground must remain possible (blue cut). Indeed, it is sufficient for a superpixel to see one foreground sample on its line of view to be labelled as foreground.

To ensure this projection consistency, we connect each sample s to the superpixels p it projects onto in all views, which defines a neighborhood $N_{\mathcal{S}}$. We define a binary term E_j as follows:

$$E_j(k_s, k_p) = \begin{cases} \infty & \text{if } k_s = \text{f and } k_p = \text{b}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.5}$$

This energy term corresponds to adding an infinitely weighted oriented edge from the superpixels to the samples on their line of view. The key property of this energy is that, as shown in Fig. 5.3, a cut assigning simultaneously to background a superpixel p and to foreground a sample s that projects on p, can never be a minimal.

Sample objectness term. In order for the previously defined constraints to be effective, the inference must include unary labelling energy term for the samples. We call this term *sample objectness* because it provides the cut algorithm with the flexibility of deciding on the fly whether to include s in the object segmentation, based on all MRF terms.

Let P_s^{f} be the coherence probability of a sample $s \in \mathcal{S}^t$. This probability is estimated with equation (3.19) making use of the probabilistic framework developed in the previous chapters. A sample is more likely to be labelled as foreground if this is coherent with the current estimation of appearance models. The notion of coherence here corresponds to the ideas developed in §3.2.

We associate a unary term and a label k_s to sample s,

$$E_s(k_s) = \begin{cases} -\log(1 - P_s^{\mathrm{f}}) & \text{if} \quad k_s = \mathrm{b}, \\ -\log P_s^{\mathrm{f}} & \text{if} \quad k_s = \mathrm{f}. \end{cases} \tag{5.6}$$

As a result of the terms defined so far, image regions (or pixels) and samples are not labelled independently anymore, but are closely related in a more complex way than proposed by Campbell *et al.* [37], better describing multi-view geometric constraints. The influence of the links created is not limited to neighbor views but extends to a larger range, while keeping the number of links reasonable thanks to the sparse 3D sampling.

Sample projection term. The energy terms E_s and E_j correspond to the first condition on superpixel/sample relationship. The converse property, inclusion of segmentations in the sample's projected set, requires enforcing a foreground superpixel p to see at least one foreground sample s, which can only be expressed with higher order MRF terms (Fig. 5.4). We opt to keep a first order MRF by modelling this behavior through an iteratively reweighted unary term.

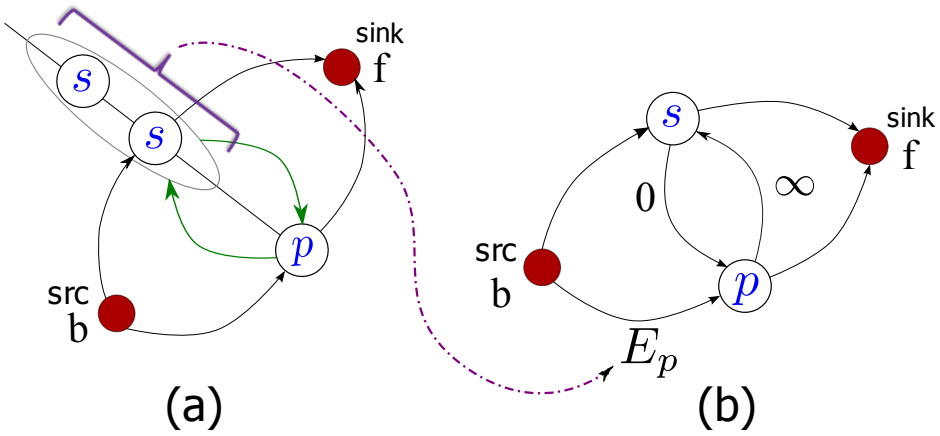


FIGURE 5.4: Relation between samples and superpixels: To be labelled as foreground, a superpixel p must see at least one foreground sample on its line of view. (a) This implies a one-to-many relationship (green edges) that can only be modeled with higher order MRF terms. (b) To avoid this situation we opt for an unary energy term (E_p) that is proportional to the maximum foreground probability for the samples on the superpixel line of view.

The desired behavior is achieved by associating to each superpixel p a sample reprojection term E_p. Its purpose is to discourage foreground labeling of p when no sample was labelled foreground in the 3D region $\mathcal{V}_p$ seen by the superpixel, and conversely encouraging foreground superpixel labelling as soon as a sample s in $\mathcal{V}_p$ is foreground. This leads to a simple unary term:

$$E_p(k_p) = -\log P(k_p|\mathcal{V}_p) \quad \text{where} \quad \mathcal{V}_p = \max_{s \in \mathcal{V}_p}(P_s^{\mathrm{f}}). \tag{5.7}$$

Since the sampling is sparse, the term $P(k_p|\mathcal{V}_p)$ is the maximum foreground probability $\mathcal{V}_p$ multiplied by a weight inversely proportional to the distance of the closest sample projection.

Consequently, foreground labelling of a superpixel that is not in the neighborhood of the projection of a sample likely to be foreground, is penalized.

5.4.4 Time consistency terms

In the case of video segmentation, the idea is to benefit from information at different instants and to propagate consistent foreground/background labelling for the frames of the same viewpoint. A set $\mathcal{N}_f^i$ of related superpixels between frames can be estimated by matching interest points or using optical flow. The propagation is done through the energy term E_f that enforces consistent labelling of linked superpixels $(p^t, q^{t+1}) \in \mathcal{N}_f^i$ as follows:

$$E_f(k_{p^t}, k_{q^{t+1}}) = \begin{cases} \theta_f \exp(\dfrac{-d(A_{p^t}, A_{q^{t+1}})^2}{2 < d(A_{p^t}, A_{q^{t+1}}) >^2}) & \text{if } k_{p^t} \neq k_{q^{t+1}}, \\ 0 & \text{otherwise.} \end{cases} \tag{5.8}$$

In this equation, θ_f will depend on the considered links: in the case of SIFT based links, θ_f is inversely proportional to the descriptor distance between the matched points. Thus, a good matching will constrain the two linked superpixels to have the same label.

In our case, we use optical flow, and θ_f depends on the proportion of pixels in a superpixel linked with flow vectors between consecutive frames. More precisely, for two superpixels p^t and q^{t+1}, if we define:

$$\theta_f(p^t, q^{t+1}) = \frac{\text{Number of pixels from } p^t \text{ linked with optical flow to } q^{t+1}}{\text{Total number of pixel in } p^t} \tag{5.9}$$

then,

$$\theta_f = \theta_f(p^t, q^{t+1}).\theta_f(q^{t+1}, p^t) \tag{5.10}$$

5.4.5 MRF energy and graph construction

Let X be the conjunction of all possible sample and superpixel labels. Our MRF energy can thus be written with the three groups of terms: the intra-view group, the inter-view group with its own multi-view binary and unary terms, and finally the time consistency group with only binary terms between successive instants t and $t+1$. $\lambda_1, \lambda_2, \lambda_3$ are relative weighing constant parameters. Finding a multi-view segmentation for our set of images, given the set of histograms H_i^B and H_i^F, and the probabilities P_s^{f}, consists in finding the labeling X minimizing:

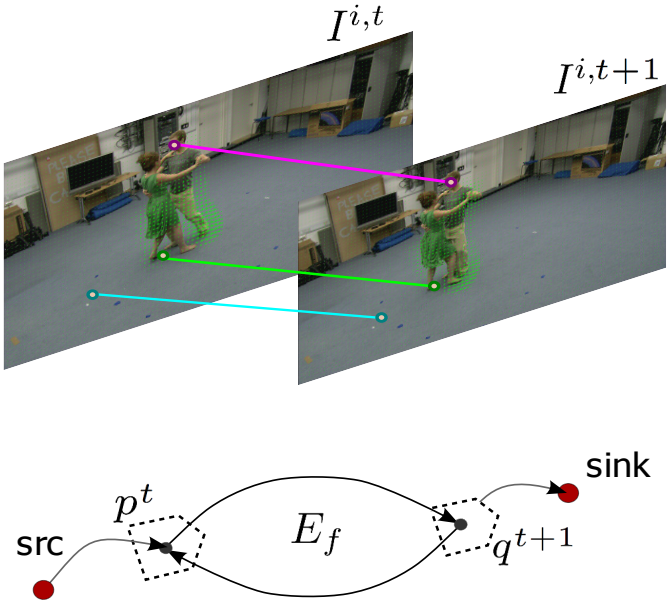


FIGURE 5.5: Temporal links between superpixels of different frames are estimated using both interest points and optical flow.

$$E(X) = \sum_{t,i} \sum_{(p^t,q^{t+1})\in\mathcal{N}_f^i} E_f(k_{p^t}, k_{q^{t+1}}) \quad + \tag{5.11}$$

$$\sum_{t,i}\left[\sum_{p\in\mathcal{P}_i^t} E_c(k_p) + \lambda_1 \sum_{(p,q)\in\mathcal{N}_p^{i,t}} E_n(k_p,k_q) + \lambda_2 \sum_{(p,q)\in\mathcal{N}_a^{i,t}} E_a(k_p,k_q)\right]$$

$$+\sum_{t}\left[\sum_{s\in\mathcal{S}^t} \lambda_3 E_s(k_s) + \sum_{(s,p)\in N_{\mathcal{S}}^t} E_j(k_s,k_p) + \sum_i \sum_{p\in\mathcal{P}_i^t} E_p(k_p)\right].$$

The submodularity constraint being satisfied in our model, we can build an s-t graph G where the min-cut will provide the solution to our energy minimization problem. This graph contains the two terminal nodes *source* and *sink*, one node for each superpixel p and one node for each 3D sample s. Edges are added between superpixels and samples according to the energy terms previously defined. Fig. 5.6 shows the resulting graph.

5.5 Computational approach

Similar to most of state of the art segmentation methods, we adopt an iterative scheme where we alternate between the previous graph cut optimization, and an update of the color models. The common visibility constraint can be used to initialize color models as in [35].

Fig. 5.7 gives an overview of the whole method. The extraction, description and linking of superpixels is done once, at initialization time. In the iterative process, the unary terms (objectness,

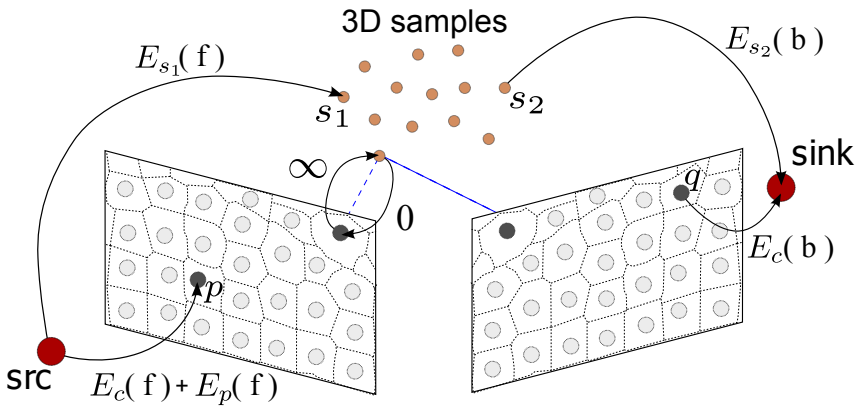


FIGURE 5.6: Graph construction. Superpixels and 3D samples are the nodes of our graph. Edges contain the different terms of our energy. A min-cut in this graph provides the solution to our energy minimization problem. The links between superpixels of different frames are not shown here for clarity.

Initialization (for each sequence instant)

1. Divide the images into superpixels.
2. Compute descriptors for all the superpixels.
3. Link similar superpixels.
4. Link superpixels from successive frames.
5. Randomly draw 3D sample positions.
6. Initialize background/foreground appearance models.

Iterated steps

7. Compute unary terms of energy from the models.
8. Minimize energy with s-t mincut.
9. Update color models from graph cut results.

Finalization

10. Final segmentation: standard graphcut segmentation at pixel level using models derived from superpixel segmentation.

FIGURE 5.7: Algorithm overview.

superpixel sample projection and superpixel labelling probabilities) computed using the appearance models of the previous iteration. The algorithm converges when no more superpixels are re-labelled from an iteration to another. Superpixel labelling at convergence is used to estimate foreground/background appearance models which are used in a standard graphcut segmentation at pixel level, with unary terms based on appearance and smoothing binary terms using color dissimilarity.

In the case of video segmentation, the same scheme is applied over a sliding window of 5-10 frames. In this situation additional cues can be used, such as considering non-moving regions as background.

5.6 Experimental Results

In this section we present the results obtained by the proposed method in the context of the published work [152]. In this case, texture is defined as gradient magnitude response for 4 scales and Laplacian for 2 scales. As an initialization step, a k-means is run separately on color and texture values. This clustering is used to create texture and color vocabulary on which foreground and background histograms (H_i^F and H_i^B) are computed. They now represent respectively the colors models Θ_i^F and Θ_i^B. We implemented our approach using publicly available software for superpixel segmentation (SLIC [124]) and using Kolmogorov's s-t mincut implementation [151]. We use superpixel sizes of 30-50 pixels to ensure oversegmentation, obtaining around 2000 superpixels per image. For appearance models, we run K-means on texture and color values, to quantize texture and color into respectively 60 and 150 "words". The region of interest is computed by keeping only 3D samples in the common visibility domain, i.e. which project inside all views. We randomly generate 100k 3D samples for all tests. The only free parameters in the method, λ_1, λ_2 and λ_3 are respectively set to 2.0 , 4.0 and 0.05 for all datasets. No particular sensitivity was observed to these settings. The initialization of the algorithm is very weak, by setting H_i^F to the statistics of the projection region of the common visibility domain of all views, which is quite large on all datasets, only eliminating about 25% of pixels on outer regions of the image. Background histograms H_i^B are set to the statistics of the known background (outside the projection of visibility domain). The computation time depends on the number of viewpoints and the number of frames. In a static case with 10 viewpoints, each iteration of the algorithm takes less than 10s with our C++ implementation and convergence is reached in fewer than 10 iterations. Tests were run on a 2.3 GHz Intel i7 PC with 4GB memory.

5.6.1 Qualitative results

To validate our approach, we run our implementation on a dozen challenging datasets presented in the previous chapters: COUCH, BEAR, CAR, CHAIR1 from [1] which we use for qualitative and quantitative evaluation, BUSTE and PLANT[1] which we use for qualitative evaluation.

The figures from 5.8 to 5.9 show the results for our methods on the various datasets. We show the graph cut result on superpixels at convergence and the final segmentation at pixel level. We illustrate the resilience of the algorithm in particular with low numbers of viewpoints on all the datasets. Very good results are obtained with only 3 widespread viewpoints (such as Fig.5.1). This corresponds to a scenario where approaches that need numerous viewpoints, e.g. [1], are likely to fail.

In complex scenarios, such as in Fig. 5.10, our first approach (Chap. 3) relying only on color [108] fails to segment foreground objects. The approach presented in this chapter, benefits from a more complex appearance model. Fig. 5.9 shows that what is considered as *foreground* object depends on the viewpoints. For the first example with 8 viewpoints the table is seen by all the views and it is identified as part of the foreground. When adding more viewpoints, the table is no longer

[1] from http://vision.in.tum.de/data/datasets/rgbd-dataset

FIGURE 5.8: Results on BEAR (3 views), COUCH (3 views), CAR (5 views) and CHAIR (9 views) datasets. The first column corresponds to one of the input images. The second and third columns contain respectively superpixel and pixel level segmentation results.

FIGURE 5.9: Results on BUSTE dataset with different numbers of views. For the 8 views result, the table is seen by all the views. With 13 views some cameras eliminate parts of the table and it is thus classified as background. Finally with all the views, the black elements in the background appear close and similar to the black pedestal. Thus, only the head is identified as foreground in this case.

entirely seen by all the cameras, and thereby it is segmented as background. Using all the views, many cameras see the black elements in the background very close to the black pedestal. They are then cut out from the foreground and only the sculpture is left.

Input Our method Djelouah [108]

FIGURE 5.10: Results on PLANT dataset (3 views) with qualitative comparison with the first approach in chapter 3. The method presented in this chapter benefits from a richer appearance model and also from intra-image consistency constraints.

5.6.2 Quantitative and Comparative results

To illustrate the strength of the approach and for the purpose of comparison, we use the same protocol as [1], computing accuracy as the proportion of correctly labelled pixels (Fig. 5.11 and Table. 5.1). We evaluate here the sensitivity of our approach to the number of viewpoints and the quality of the segmentation result compared to state of the art approaches [1, 39, 108], by randomly picking 10 viewpoint subsets for a given tested number of viewpoints and averaging results.

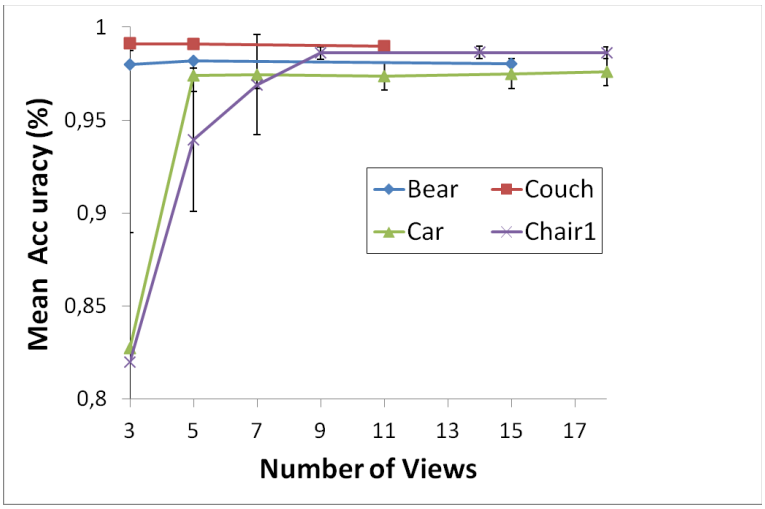


FIGURE 5.11: Sensitivity of the multi-view graph cut to the number of views.

Clearly Fig. 5.11 shows that our approach exhibits very little sensitivity to the number of viewpoints and achieves excellent segmentation results even with only 3 widespread viewpoints. Let us emphasize the excellent performance of the algorithm on CAR and CHAIR1 datasets, despite the very low number of viewpoints used and the challenging nature of color ambiguities in the datasets.

The difference of segmentation precision between approaches is mainly due to some difficult color ambiguities in the model, such as shadows that appear consistent both with hypotheses of geometric and photometric cosegmentation methods. In [1], it should be noted that depth information and plane detection significantly help, especially through the identification of the

Dataset	Our Method		Kowdle [1]	Djelouah [108]	Vicente [39]
Couch	3 99.1 ± 0.2	11 99.0 ± 0.2	11 99.6 ± 0.1	11 98.8 ± 08	not available
Bear	3 98.0 ± 1.0	15 98.0 ± 1.0	15 98.8 ± 0.4	15 98.8 ± 0.4	not available
Car	5 97.4 ± 0.8	44 97.0 ± 0.8	44 98.0 ± 0.7	44 0^*	44 91.4 ± 4.3
Chair1	9 98.6 ± 0.3	18 98.6 ± 0.3	18 99.2 ± 0.4	18 88.0 ± 2.0	18 86.9 ± 7.8

(*) Foreground is not identified in this dataset.

TABLE 5.1: Quantitative evaluation of our approach with a static scene. The table presents comparisons with state-of-the-art approaches (*nb* views, *Accuracy*). Notice that our approach achieves equivalent segmentation results with significantly fewer images than other approaches.

ground plane, which eliminates some ambiguities at the price of requiring more viewpoints for the purpose of obtaining the stereo.

From table 5.1, we observe that adding more views does not seem to improve segmentation results for the proposed method. This can be explained if we consider that the method is already taking a maximum advantage of the limited assumptions we have. To improve further the results, one should consider additional information present when using many input views. This is the case for plane base multi-view segmentation method [1] where depth information is used, but other assumptions can be considered, such as appearance similarity between close viewpoints.

5.6.3 Video segmentation results

In the case of video sequences, our framework has the ability to propagate multi-view segmentation evidence over time. It also enables the propagation of temporal evidences from a given viewpoint to other viewpoints, e.g. static background or moving foreground. They can help resolve local segmentation ambiguities in few views in time or space. In order to demonstrate these principles, we evaluated the approach with two datasets DANCERS and HALF-PIPE from [19] and [154] respectively. The first consists of 8 cameras in an indoor setup whereas the second is captured with 4 hand-held cameras in a challenging outdoor environment. Fig. 5.12 shows segmentation results with and without temporal consistency. Results on the DANCERS sequence (first row) show how temporal evidences help resolving background ambiguities. This is achieved by taking advantage of pixels with static values when building the background model.

With the HALF-PIPE video dataset (Fig. 5.12 second and third rows), we experiment propagation in time and space of user inputs. In this dataset, the complex nature of the environment, the hand-held cameras in general motion and non-static backgrounds, and the few, widespread viewpoints make the segmentation very challenging. As shown in Fig. 5.12, specifying ambiguous foreground/background regions with two strokes in a single view (second row, left image) is sufficient to obtain visually satisfying results. This demonstrates that cues in an image can benefit to other images with different viewpoints and at different times.

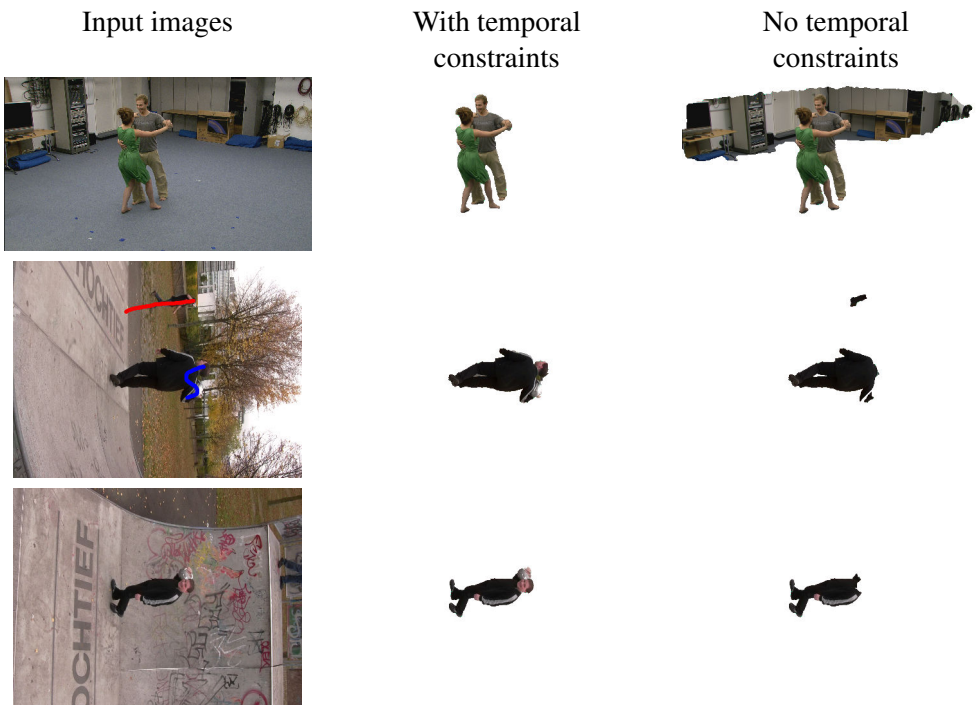


FIGURE 5.12: Multi-video segmentation results using space and time propagation (middle) vs. space only propagation (right) of information. Row 1: DANCERS dataset; Rows 2&3: HALF-PIPE dataset. User inputs are shown in blue (fg. region) and red (bg. region)

The main motivation behind this work was to provide a new solution to the multi-view segmentation problem and to show that the quality of the results can be greatly improved while keeping user interaction as minimal as possible. To illustrate the potential of the proposed multi-view framework (see Fig. 5.13), we perform a comparative study with a standard state of the art video segmentation tool [6]. In our case, user interaction is not used, and segmentation is automatically performed in all of the 8 viewpoints. Using video snap cut [6], a very detailed user guided segmentation is provided in the first frame. We let both algorithms perform the segmentation of the entire video. The monocular approach quickly starts to drift, losing important parts of the foreground but also including background elements in the object layer. This becomes even more pronounced when foreground contours see strong changes. The result after fewer than 200 frames ($\sim$ 8 seconds) is a video with severe segmentation errors on the majority of the frames. Using a multi-view approach, the constraints from the different views allow us to keep the segmentation on track all along the video. Some defects might be observed with strong motion, like around the dancer hands in frame 131. But this does not affect the whole sequence. As an illustration of this, a few instants later (frame 149), segmentation is almost perfect despite the strong change of topology and motion. To obtain similar results using Video SnapCut, one would need to keep correcting the segmentation as soon as it drifts away from the desired result. This represents a time consuming process that needs to be repeated for each view. With the multi-view approach, not only are the results better, but all the views are segmented simultaneously.

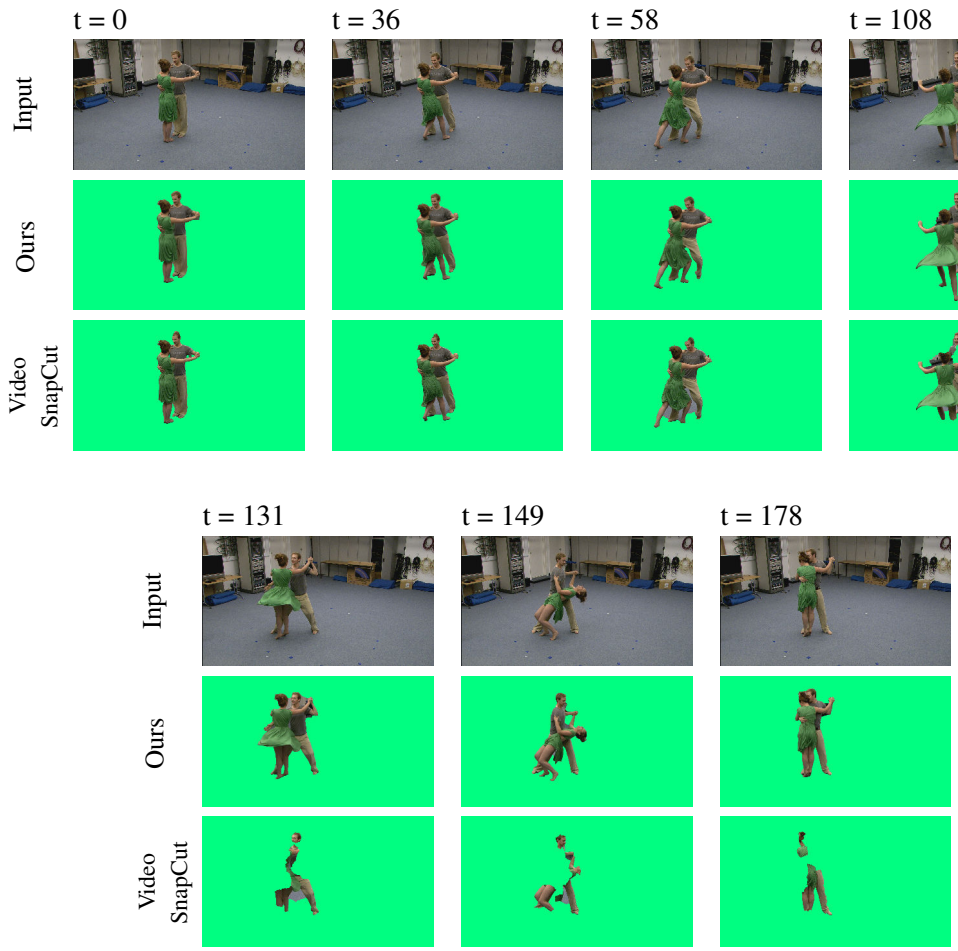


FIGURE 5.13: Comparative results on the DANCERS dataset: the Video SnapCut [6] tool is initialized with user interaction delimiting precisely the contour of the object. In our multi-view approach, segmentation is automatically obtained. During the course of the video, we see how the monocular approach starts to drift away, resulting in strong degradation of the segmentation. In the multi-view approach, some defects may appear locally (dancer hands in frame 131), but this is quickly corrected and segmentation is almost perfect few frames later (frame 149).

5.7 Conclusion

We have presented a new approach to solve the MVOS problem based on iterated joint graph cuts. To our knowledge we propose the first unified solution dealing with intra-view, inter-view, and temporal cues in a multi-view image and video segmentation context, into a single consistent MRF model. The approach is shown to cope with a low number of widespread viewpoints, many times achieving state of the art quality with only three wide baseline views. The algorithm has been evaluated on very challenging datasets, including MVOS segmentation with videos from four moving hand-held cameras. We believe that the framework is a solid basis to explore more complex multi-view motion models, which we suspect may even further improve segmentation quality in the video MVOS problem context.

Chapter 6

Tracking with uncalibrated cameras

6.1 Introduction

The work in this thesis has mainly targeted calibrated and synchronized multi-camera setups, with the objective of foreground/background segmentation. The promising results obtained in the case of video segmentation naturally raise the question of the potential of using movement in the task of segmentation and tracking. More precisely, one has the intuition that object motion in the multi-view context can be used to identify and differentiate between tracked objects/persons and to estimate pixel level segmentation. Since the multi-view video segmentation problem is not fully explored in the calibrated case, providing a solution when calibration information is not present represents a significant step. An important challenge on the way to achieve this objective is the ability to match action between uncalibrated views. In this chapter, we focus on this problem in the context of object tracking. After reviewing the main monocular [155, 156] and multi-view tracking methods [13], we will see that tracking in the uncalibrated multi-view scenario is a problem barely addressed in recent years. In this case, it is important to be able to compare apparent motions in different views, in order to decide if they are indeed the 2D projections of the same 3D motion. We therefore explore some of the different ways this problem has been addressed, particularly in the multi-view action recognition community.

In this chapter, we will see that using motion has been extensively explored in the monocular case with a wide variety of methods for tracking [157–159] and in the multi-view situation when camera calibration is available [22, 160, 161] or when some intermediate representation exists [162, 163] (i.e. a 2D occupancy map). Despite the important amount of work in this domain, we will see that it is still relatively easy to find situations where even the most recent tracking algorithms fail. Usually a multi-view approach remarkably solves many of the encountered problems when calibration is available, but to the best of our reading no approach proposes a solution in the uncalibrated case. If we assume that views are not affected by these issues at the same instants, then tracking can be greatly improved through a hypotheses validation framework. To address this situation, we propose to use the self-similarity matrix [164] in an energy minimization framework as a way to compare and match motions in different views. The preliminary results obtained on multi-view datasets are promising and suggests that using this

approach can provide more robustness to monocular tracking methods in a hypotheses validation framework.

6.2 Related work

In this chapter we are interested in two main problems. The first is object tracking both in single and multiple camera setups, and the second is motion comparison and matching between different views. In this section we review the related work for these two problems.

Monocular tracking and object/person tracking in general, is considered as one of the main tasks of computer vision and represents the necessary preliminary step for a wide variety of applications like action recognition, video indexing and video surveillance. Naturally, it is one of the most studied problems in computer vision, with a great number of methods addressing the problem in different ways. We will review here the main directions with the most recent works from the state of the art.

The representation scheme is an important characteristic of tracking algorithms and many approaches have been proposed: from the holistic templates used in [165–168] to the more recent sparse representation [169–171]. The choice of the features used to represent the object appearance is important to distinguish the tracked object from other parts of the scene, and to be able to identify it throughout the video sequence. Used features are based on gradient [172–175], color information [176, 177], texture [178, 179] or spatio-temporal information [180, 181].

When the objective is to track arbitrary objects (which can be seen as image regions), the current trend is to initialize and then update on the fly a suitable appearance model (either descriptive / generative or discriminative) thanks to online learning tools. In generative online methods, the object is defined as the region most similar to the target model. These trackers try to model object appearance using specific features like superpixels [182], integral histograms [183], local descriptors [184] or simply intensity, as in the seminal work of Ross *et al.* [185]. In discriminative methods, tracking is formulated as a classification problem between foreground and background. Online discriminative methods rely on a trained dynamic classifier with boosting [157–159], random forest [186] or SVM [155]. Another important aspect of tracking algorithms is the usage of contextual information and some approaches have used local information around the object to assist tracking [187–189]. This contextual information can be integrated in the object description as proposed by Chen *et al.* [156].

Contrary to online object tracking, where the user provides a bounding box of the object to track, multi-object tracking methods usually rely on detection or short term tracks as the main working element. It means that a specific category of objects is targeted: cars, people, faces, moving objects, etc. For example, in pedestrian tracking or sport events, these methods often rely on some person detector to provide the series of detections. Multi-object tracking is formulated as an association problem between tracklets. Among the proposed methods, we can cite the work of Huang *et al.* [190] and Perera *et al.* [191] based on the Hungarian algorithm. The association is generally formulated as an assignment problem, with costs depending on features based on

color and motion, solved using linear programming [192], cost-flow networks [193], continuous-discrete set optimization [194], higher-order motion models [195] or particle filtering [196–199]. Another interesting aspect here is that interaction between the different targets can be used to improve tracking using collisions [200], correlated motion [201] or priors from the context [202].

Multi-view tracking Despite the interesting results obtained in monocular situations, using a multi-camera approach is necessary to achieve more precise long term tracking. An example of the benefits of the multi-camera approach can be seen in the method proposed by Krumm *et al.* [203], where two stereo cameras are used to track people and maintain their identities. This problem has also been addressed in more general setups using a Kalman filter on 3D points [204]. Likewise, Black *et al.* [205] use Kalman filtering to track in 2D and 3D with trajectory prediction during occlusions. Otsuka and Mukawa [206] propose a Bayesian recursive framework for tracking objects moving on a two-dimensional plane with hypothesis generation and testing. Mittal *et al.* [207] use pixel segmentation in the images to estimate 3D locations on the ground plane, and tracking is performed with a Kalman filter. Techniques based on discretized 2D occupancy maps include [208, 209] and rely on depth estimation [208], a visual hull procedure [209] or a generative model combining color and motions [162]. Still in the case of 2D occupancy maps, Berclaz *et al.* [210] formulate the problem as finding the flow between the nodes, where the flow expresses the transition from one detection location to the other. The constraints are expressed as a linear program and can handle missing detections. To avoid identity switches, appearance cues are exploited [13] and show good results in handling occlusions.

Methods reasoning on 3D reconstruction also rely on camera calibration [22, 160, 161]. A maximum a posteriori approach is proposed in [22], where the function to be optimized takes into account both detection and data association. In [160], Leal-Taixe *et al.* propose to build a multi-layer graph from input detections. In the first layer, detections and links between nodes are used to describe possible trajectories whereas in the second layer, nodes correspond to reconstructions from detections. The assignment problem is formulated as a min-cost problem. Hofmann *et al.* [161] propose a combined maximum a posteriori (MAP) formulation to jointly model multi-camera reconstruction and global temporal data association.

In other situations, the objects are assumed to be tracked in the different views and the main problem is data association, usually formulated as a bipartite graph matching problem [211] in the case of two cameras. With more views, the proposed solutions are suboptimal using K-partite graph matching [212, 213]. The problem has also been addressed as a multi-dimensional assignment problem, solved with greedy randomized adaptive search procedures.

In this chapter we are more interested in how to model motion with the objective of comparison in different viewpoints. Morariu *et al.* [12] propose a solution using uncalibrated cameras, where appearance correspondences between different frames and temporal and spatial dynamics between views are learned. In a 2 camera setup, tracking in the view where there is no occlusion is used to predict the position of the object of interest in the other view. However, results are only shown for very simple situations.

Multi-camera action analysis and gait recognition are important problems in computer vision. In multi-view action analysis, the main objective is human action recognition. Two categories of work can be distinguished: view-invariant features and transfer learning across cameras. Among the proposed features we can cite tracked body parts [214], 2D trajectories [215], correlated feature extraction using calibration information [216] or reconstruction from silhouettes [15]. Some methods propose to learn a transfer function between the views [217] or a bag-of-bilingual words [218] to represent the action in two views. However, all these methods require a retraining stage in the case of different camera positions.

Junejo *et al.* [164] propose to use temporal self-similarity matrices for multi-view video synchronization and action recognition. They show the interest of using such a representation as it exhibits interesting stability across viewpoints.

This last work is inspired by tools developed in the context of gait recognition [219, 220]. In these two works, self-similarity matrices are used to analyze the motion of a pedestrian in order to identify the frequencies and to perform gait recognition. The objectives and the context are different than the one presented in this work, but they represent an interesting first usage of self-similarity matrices in motion analysis.

In the work of Johansson [221], point light displays are placed on body joints to visualize the motion of a person and it was later shown [222] that this representation can be used for person identification. Gait recognition can be classified into model based and image based. In model based approaches, a predefined model (2D or 3D) is assumed and a pattern is fitted to match the observations. In [223], the motion of the legs is analyzed to perform gait recognition. Using a 3D skeleton, Urtasun and Fua [224] try to recover gait parameters. Despite the interesting results they achieve, these methods are limited by the inherent ambiguity of passing from a 3D to a 2D representation. There is also a possible loss of information when fitting a generic model to a particular individual. Image based gait recognition techniques are based on silhouettes [225–228] or optical flow and feature trajectory points [229–232]. These works do not represent an exhaustive list of methods addressing gait recognition, but a selection from recent advances in this field. It is essentially the robustness to errors in the segmentation or noise in the data that has been explored, and very little is said about the view invariant nature of the proposed methods.

In this work, we are mainly interested in the capacity to distinguish between different motions in multi-camera videos. To this end, we explore the usage of self-similarity matrices for matching motions in different views and its potential for hypothesis selection and validation in a multi-view tracking framework.

6.3 View invariant motion description with self-similarity matrix

In this section we define self-similarity matrices in the context of multiple-view action matching. We assume an object k of interest is tracked for a given interval of time I. The self-similarity

matrix for this object on the interval I is

$$SSM(k, I) := [d_{ij}]_{i,j \in I} \tag{6.1}$$

The d_{ij} are the distance values between image descriptors computed at two instants i and j on an image region defined by the object bounding box. Following [164], we use Histograms of oriented Gradients (HoG) [175] and Histograms of oriented optical Flow relative to the bounding box (HoF) [180] as image descriptors. In the context of moving cameras we also use Motion Boundary Histograms (MBH) [233]. More precisely, the object bounding box is divided into N_{rows} by N_{cols} blocks and each pixel participates in the two closest blocks. The distances d_{ij} used in (6.1) can be more precisely defined now:

$$d_{ij} = \sum_{x=1}^{N_{\text{rows}}} \sum_{y=1}^{N_{\text{cols}}} \sum_{b} |H_{b,i}^{x,y} - H_{b,j}^{x,y}|, \tag{6.2}$$

with $H_{i,b}^{x,y}$ being the value for bin b, in the descriptor for block with coordinates (x, y) in the bounding box, at the instant i. An example of a tracking box is presented in Fig. 6.1.

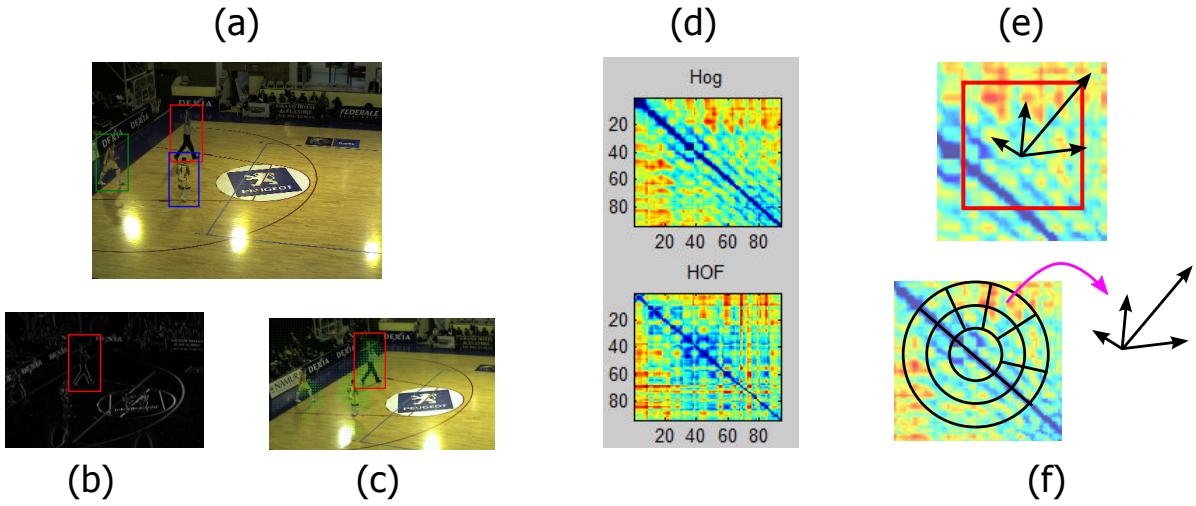


FIGURE 6.1: Elements of the multi-view tracking method are presented here: (a) In the input sequences a set of objects are tracked with a bounding box. (b) Using histograms of oriented gradient (c) or histograms of oriented optical flow (d) self-similarity matrices are estimated for each tracked object, over the tracking interval. To compare self-similarity matrices we use both (e) standard histograms of oriented gradient and (f) the log-polar block structure proposed in [164].

In this work we draw inspiration from [164] where self-similarity matrices are used for action recognition. The authors show the stability of SSMs over different views and persons for the same action. To get a fist insight into the representation power of SSMs, Junejo *et al.* [164] rely on the notion of "dynamic instances" proposed by Rao *et al* [214]. The authors argue that continuities and discontinuities in position, velocity and acceleration of a 3D trajectory of an object are preserved under 2D projections. In the SSMs, these particular dynamic instances correspond to valleys of different areas/spreads. The robustness of the SSM representation is further explored and tested in [164] for different actions and viewpoints. In this work we explore

the usage of self-similarity matrices as a way to match persons/objects across views based on their individual dynamics.

As a first example of this, we show self-similarity matrices for 3 basketball players tracked in time (Fig. 6.2). Visually, it is easy to match person tracking across views based on similarity matrices.

The question is now how to compare two self-similarity matrices. In [164], Junejo *et al.* use patch-based log-polar descriptors. The log-polar descriptors are computed on small patches around diagonal elements of the SSM. The descriptor structure is shown in figure 6.1(f). For each block, we compute a normalized histogram of oriented gradient. The final descriptor is the patch wise concatenation of block histograms.

In [164] the authors only consider local descriptors around diagonal elements, arguing that the uncertainty of SSM values increase far away from the diagonal. This argument is particularly valid in the context of action recognition where we try to find the most generic description of an action and where the action is visible in the entire interval. However, in the context of cross-view matching of tracked objects, occlusions can be present at various time intervals and in this case SSM regions far from the diagonal (hence, corresponding to instants that are distant in time) may contain variations useful for matching. Another argument in this direction comes from [164] where best results are obtained with larger log-polar patches. So we naturally choose to associate HoG descriptors over all the SSM to perform the matching.

The final descriptor for self-similarity matrices consists of two sub-vectors. The first is the log-polar descriptor proposed in [164]. For each element of the diagonal, it consists in a vector of HOG descriptors computed over each one of the 11 blocks of the polar structure. The second sub-vector is also a HOG descriptor, but computed on rectangular blocks. For a selected set of points in the SSM, a HOG descriptor (Fig. 6.1.(e)) is estimated inside a locally defined rectangular block (using the same temporal extent as for the polar structure). The set of points used to compute these descriptors is selected on a regular grid over the SSM. From now on the SSM descriptor is noted H_{SSM}.

6.4 Multi-view tracking with SSM

6.4.1 Direct object identification

We first present here the simple scenario of cross view identification. We assume m subjects are tracked in synchronized multi-view video sequences, without overlap or occlusions. The objective is to match the tracked subjects across views using only their motion.

Using SSMs one can expect to achieve this objective by directly comparing SSM descriptors. If we note $\mathcal{H}_i$ the set of tracks in view i and given a subject k_i in view i, finding the good match in

view $j \neq i$ amounts to solving

$$z_j^* = \underset{z_j \in \mathcal{H}_j}{\arg\min} \| H_{\text{SSM}}(z_j, I) - H_{\text{SSM}}(k_i, I) \|_2 \tag{6.3}$$

with $H_{\text{SSM}}(z_j, I))$ the gradient based descriptor for the SSMs associated with object z_j (a tracking hypothesis) on time interval I in view j. This descriptor is the concatenation of HOG and log-polar block descriptors as previously defined. We note by $\mathcal{H}_j$ the set of all tracking hypotheses for view j.

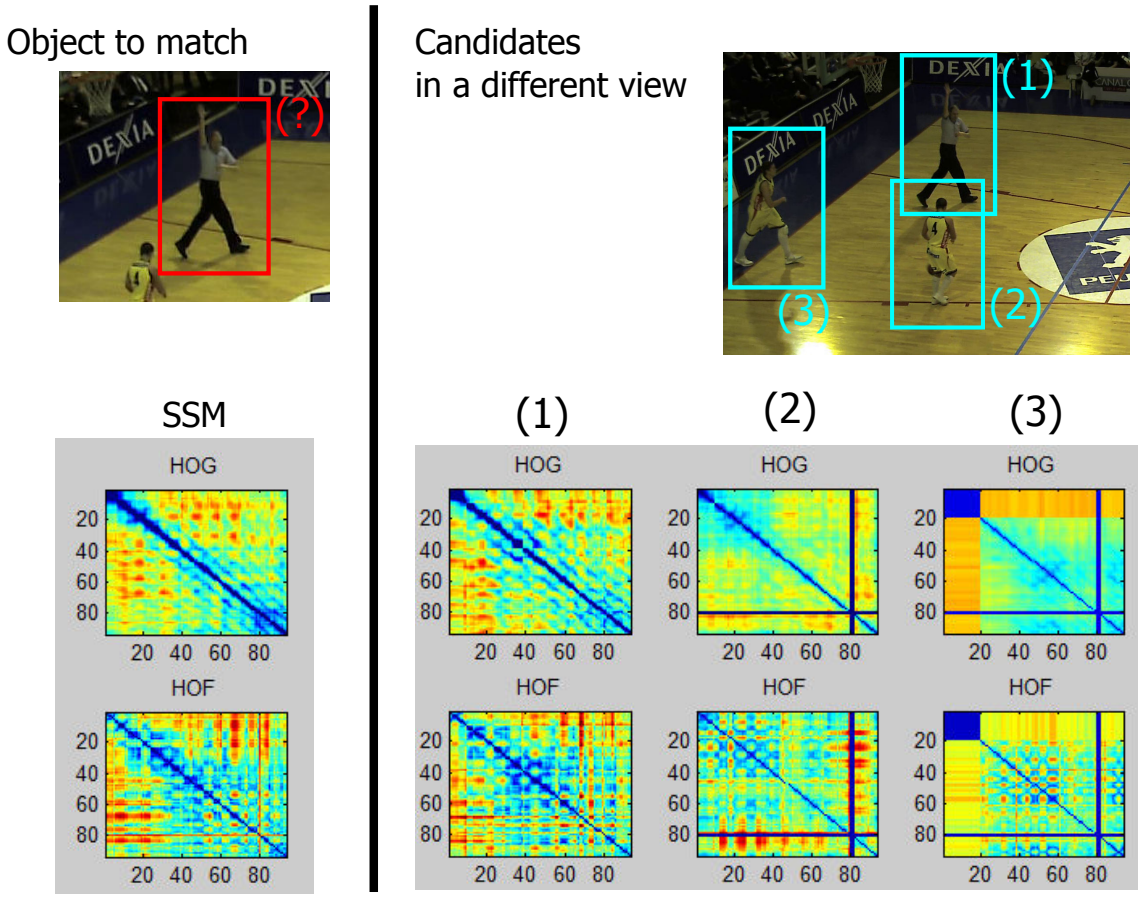


FIGURE 6.2: Hypotheses matching in multi-view tracking using SSMs: in a given view, an object and the bounding box is used to estimate self-similarity matrices based on gradients (HOG) and optical flow (HOF). In the second view, different subjects are candidates in the matching process. It is visually easy to identify the correct hypothesis using the SSMs.

However, this approach is unlikely to work on complex scenarios where tracked subjects have similar motion. In particular, the assumption regarding the stability of the self-similarity matrix structure may not be verified. This may not be a critical issue if the subjects have well differentiated motion, but in the general context of tracking, subjects are often performing the same action (walking, running, etc.). To counterbalance this, we use all the views and all the objects in the decision process. More precisely we are not any more trying to just match object tracks in two views, but we are interested in matching all the objects across all the views. First, we note by m the number of objects we are tracking. We assume that each object is correctly tracked in each one of the n view. Then, we define the labelling L as the tuple $(L_1, \ldots, L_n)$ that orders

tracking hypotheses in the view i as $L_i = (l_i^1, \ldots, l_i^m)$, associating to each object k, the tracking hypotheses $l_1^k, \ldots, l_n^k$, in the views $1, \ldots, n$ respectively. Then, matching tracking hypotheses between the views comes down to finding the labelling L^* so that

$$\begin{aligned} L^* = \arg\min_{L} & \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{m} \| H_{\text{SSM}}(l_i^k, I) - H_{\text{SSM}}(l_j^k, I) \|_2, \\ \text{with } & l_i^k \in \mathcal{H}_i \quad \text{and} \quad l_j^k \in \mathcal{H}_j \\ \text{s.t } & \forall i, l_i^k \neq l_i^{k'} \end{aligned} \tag{6.4}$$

We note that to avoid the $m!$ equivalent solutions, we fix the label values in the first view, i.e. $\forall k, l_1^k = k$. Since we are considering relatively simple scenarios with few objects, we adopt an exhaustive search to solve Eq. (6.4).

The first obvious advantage is the constraint of using each hypothesis once. The second advantage is the matching over all pairs of views. Intuitively, by taking into account more pairs of views, we increase the chance to have a high similarity between self-similarity matrices on some of them. The objective is to achieve multi-view information propagation from these views.

The assumptions made here correspond to the simplest scenarios, but the proposed model contains the core idea of multi-view matching with SSMs. In the next section, we build on the notions presented here to propose a model addressing more complex tracking scenarios.

6.4.2 Multi-view tracking association

We explain here how the ideas developed in the previous section extend to more general situations. From now on, we only assume to have tracking with possibly overlaps. Image descriptors are not computed during occlusion intervals and this results in SSMs with zero values at the corresponding coordinates. Such a situation can be observed in Fig. 6.3 from the occlusions between the different tracking hypotheses.

The self-similarity matrices must be compared on the entire video sequence. To compare tracking hypothesis matrices, the missing values must be taken into account. To this end, we will not use basic Euclidean distance between SSM descriptors any more, instead we define a new distance function between two self-similarity matrices. First, the SSM descriptors $H_{\text{SSM}}(l_i^k, I)$ are considered as a vector where each component c has the value $H_{\text{SSM}}^c(l_i^k, I)$ and corresponds to a value associated with one of the HOG or log-polar histogram bins.

This new distance $\text{Dist}(l_i^k, l_j^{k'})$ between the two tracking hypotheses l_i^k and $l_j^{k'}$ is defined as follows

$$\text{Dist}(l_i^k, l_j^{k'}) = \sqrt{\sum_c \text{Dist}_c^2(l_i^k, l_j^{k'})} \tag{6.5}$$

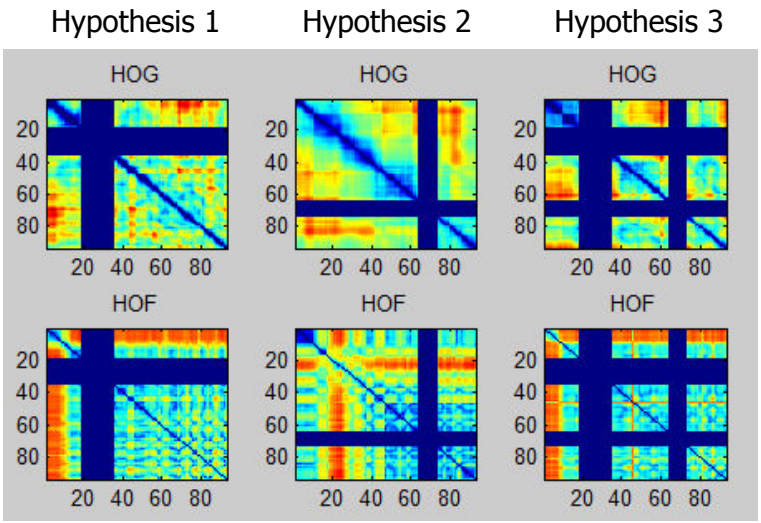


FIGURE 6.3: SSM computed using HOG and HOF descriptors with zero values (in blue) during occlusions. We can see how hypothesis 3 corresponds to a tracking hypothesis that has overlaps with the two other hypotheses.

where

$$\text{Dist}_c(l_i^k, l_j^{k'}) = \begin{cases} |H^c_{\text{SSM}}(l_i^k, I) - H^c_{\text{SSM}}(l_j^{k'}, I)|, & \text{if } H^c_{\text{SSM}}(l_i^k, I) \text{ and } H^c_{\text{SSM}}(l_j^{k'}, I) \text{ are defined} \\ \epsilon_c(i,j) & \text{otherwise.} \end{cases} \tag{6.6}$$

This means that if the values for the component c are properly estimated for both tracking hypotheses l_i^k and $l_j^{k'}$, then the descriptor distance uses absolute differences between component values. If the descriptor component is not defined (i.e., the descriptor component represents an SSM region where an occlusion occurs), the constant value $\epsilon_c(i,j)$ is used. This constant represents the minimal distance between tracking hypotheses of views i and j for component c. The idea here, is that when there is an occlusion, the SSM is non-informative and therefore we use this minimum distance $\epsilon_c(i,j)$ to penalize this hypothesis association as little as possible.

In this more complex scenario, many elements can influence the matching process described in the previous section: A hypothesis that has many occlusions can falsely have a shorter distance to other views hypotheses due to the compensation constants ϵ_c. Another source of errors can come from views where the similarity between self-similarity matrices describing the same action is not fully verified.

To take all these elements into account, we propose an energy based model where rather than minimizing a global distance between pairs of hypothesis associations, we will minimize a global energy function and each association of hypotheses will contribute by an energy term. The proposed energy should express the following principles:

Number of frames used to compare hypotheses. Two tracking hypotheses are compared on the entire sequence and depending on the trajectories and camera positions, this can lead to several occlusions and thus less information in the SSM. In general, the distance obtained by

comparing SSMs having fewer missing values, contains more information and should be more trusted.

Relative distance between tracking hypotheses for two views For a given pair of views, comparing two tracking hypotheses should take into account the smallest distance between all possible trajectory associations. The idea is that this smallest distance gives a good prior on the distance to expect for matching hypotheses. In this case, one can also expect to model "how much the cross-view stability of the SSM is verified". Indeed, the distances for matching hypotheses should be grouped near this smallest distance, and those corresponding to false hypotheses associations distributed farther away.

To help express these ideas, we note by

- $\text{Fr}(l_i^k, l_j^{k'})$, the number of shared frames for two tracking hypotheses l_i^k and $l^{k'_j}$, counting only frames where both objects are visible and do not overlap with other tracked objects.
- $\text{Fr}_{i,j}$, the maximum number of shared frames between all possible tracking hypotheses $l_i^k$and $l_j^{k'}$.

The smallest distance between all possible hypotheses associations for two views i and j is

$$\text{Dist}_{i,j} = \min_{l_i^k, l_j^{k'}} \text{Dist}(l_i^k, l_j^{k'}), \tag{6.7}$$

and the standard deviation from this smallest distance is

$$\sigma_{i,j}^2 = \frac{1}{|\mathcal{H}_{ij}|} \sum_{l_i^k, l_j^{k'}} \Big(\text{Dist}_{i,j} - \text{Dist}(l_i^k, l_j^{k'})) \Big)^2, \quad \text{with } |\mathcal{H}_{ij}| \text{ the total number of hypotheses combinations for views } i \text{ and } j \tag{6.8}$$

To solve the multi-view tracking association problem we use the following energy for a given labelling proposition:

$$E(L) = \sum_{i=1}^{n} \sum_{j=i+1}^{n} \sum_{k=1}^{m} \exp\big(1 - \frac{\text{Fr}(l_i^k, l_j^k)}{\text{Fr}_{i,j}}\big) \frac{\big(\text{Dist}_{i,j} - \text{Dist}(l_i^k, l_j^k)\big)^2}{2\sigma_{i,j}} \tag{6.9}$$

To solve the multi-view tracking association problem, we find the labelling L^* that minimizes the energy function E:

$$L^* = \arg\min_{L} E(L), \tag{6.10}$$

by an exhaustive search over all combinations. This is possible because the number of hypotheses is relatively small. In the case of larger hypothesis sets, a heuristic must be adopted to reduce the search space.

6.4.3 Multi-view tracking hypotheses selection

Monocular tracking approaches are extremely sensitive to occlusion and in a multi-view approach one can use the redundancy of information in different views to improve tracking results. Recent methods use geometric information to do so through occupancy maps [234] or a 3D reconstruction of the trajectories. The models described in the previous section can be used to improve tracking results without requiring any calibration information or appearance similarity.

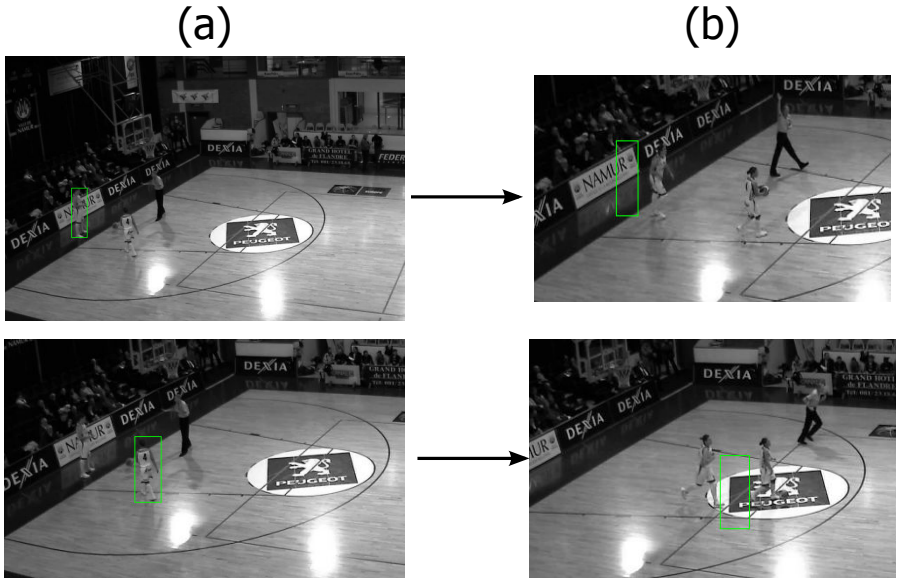


FIGURE 6.4: Tracking errors using a recent tracking algorithm from the state of art (STRUCK [155]). We first show (a) the bounding box used in the first frame then (b) the tracking failure even in the very simple scenario of player tracking.

Monocular tracking algorithms usually depend on a set of parameters, the most common one being the size of the search window between two consecutive frames. A good tuning of these parameters can greatly improve tracking results but there is no way to know *a priori* which values are best suited to each dataset. Monocular tracking methods are also very sensitive to occlusion which can induce identity switches. These errors are not limited to interaction between different tracked objects, they can also occur when there is a strong resemblance with background regions (see Fig. 6.4).

If we assume that views are not affected by these issues in the same instants, then tracking can be greatly improved by means of a hypotheses validation framework. In each view and for each object, a set of hypothetical tracks are generated. These tracking hypotheses can be obtained using different monocular trackers with different initializations or tracking parameters (maximum displacement distance, etc.). They can also be obtained using trained detectors (like a pedestrian detector). This can be directly translated into the energy based framework, resulting only in an increased number of hypotheses in each view. We will not only decide which hypotheses should be matched but we will also eliminate the inconsistent ones.

6.5 Preliminary results

In this section we present the results in two different datasets. In the first case we have 3 GoPro cameras placed in an office capturing a dynamic scene from three distant viewpoints. The tracking is easily done in this scenario since the subjects keep about the same position. However the persons are performing different actions. Using self-similarity matrices on the tracking boxes (see Fig. 6.5), we are able to match the different tracking hypotheses just by using the direct object identification framework (Eq. 6.4). In this case, direct matching is possible because there is no occlusion (and overlap is limited) between the different subjects. The comparison can be performed on the entire video sequence.

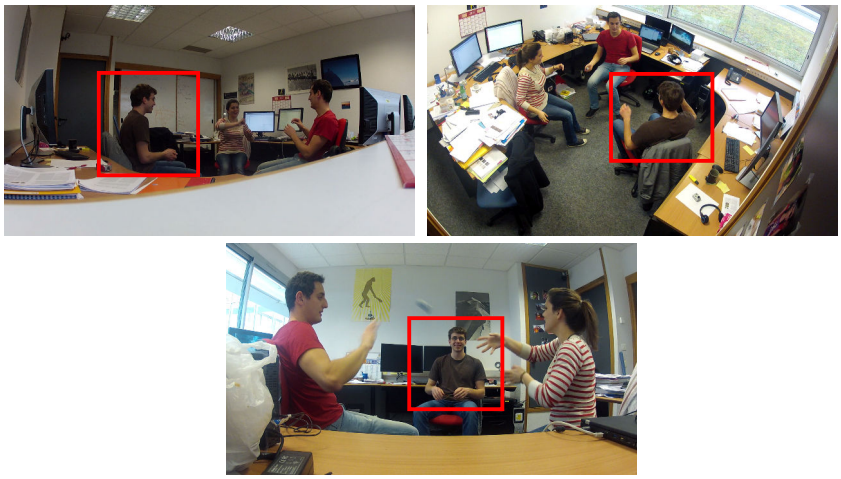


FIGURE 6.5: In this multi-view video sequence, 3 persons are playing while keeping a relatively stable position in space. The red rectangle shows the box used to track the subjects. Since there is no occlusion between the different persons, the direct matching method is able to match the different tracking hypotheses despite the large discrepancy between the viewpoints.

In the second scenario, we consider a basketball dataset[1]. In this dataset, we have different viewpoints and occlusions occurring at various instants depending on the viewpoint. In each case, we use ground truth labelling to generate the different tracking hypotheses, as shown by the colored paths in Fig. 6.6. This dataset illustrates the typical scenario for which the multi-view tracking framework is designed: large discrepancy between the viewpoints, occlusion between the objects that leads to different tracking hypotheses. Here the top view (first image in Fig. 6.6) has no occlusion between the players. It will be used as our ground truth against which hypotheses selection is performed. We also incorporate the fact that some tracking hypotheses are incompatible (two paths for the same object), thereby reducing the search space. To compare SSMs we use a combination of descriptors defined for three different sizes (10, 20 and 30 frames). To achieve correct matching and hypotheses selection, both oriented gradient and optical flow SSMs are used. Using only one of them leads to mismatches in certain views.

The results presented in this section are not sufficient to properly validate the proposed methods, and further experiments should be conducted. However, this shows the potential of using motion

[1] http://www.apidis.org/Dataset/

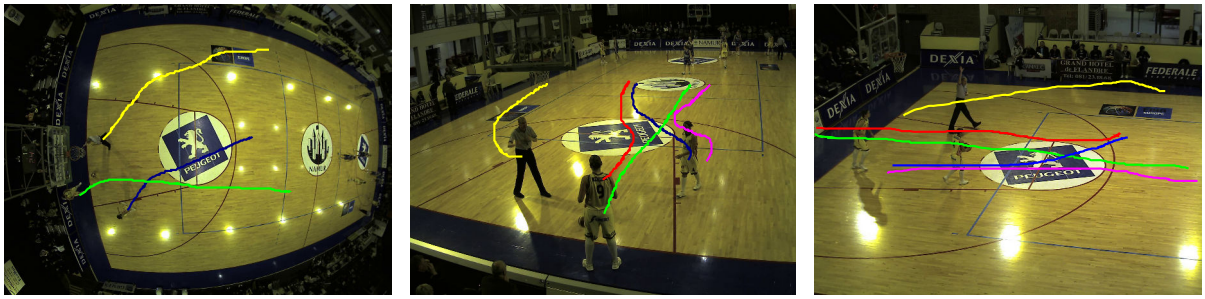


FIGURE 6.6: Example of a multi-view tracking scenario with hypotheses validation. In the leftmost image (corresponding to the top view) there is no ambiguity in the tracking, contrary to the other views. There is no occlusion between the players and this provides the ground truth trajectories that we try to match in the other views. In the middle and right-most views, occlusions occur at various instants. If we consider a monocular tracker, these occlusions can be the source of confusion between the objects. To avoid this, we consider all the possibilities, resulting in different tracking hypotheses. However, since the challenging events do not occur at the same instant in all the views, our framework is able to successfully select the appropriate tracks and match them across the cameras.

as way to address the multi-object tracking in the challenging scenario of uncalibrated multi-camera setups. Using a representation as simple as self-similarity matrices, we show that it is possible to match various actions in very different views.

We also note that in the case of moving cameras, the accumulation of errors on the estimation of the relative optical flow can be detrimental. To avoid this, motion boundary histograms [233] can be advantageously used to compute self-similarity matrices. Recent developments in action recognition [235] using descriptors based on motion boundary histograms achieve interesting results even with a significant amount of camera motion.

6.6 Conclusion

This thesis has been mainly concerned with calibrated multi-view scenarios, but in the course of this work, the necessity of addressing the non-calibrated setups has appeared, despite the apparent difficulty of performing object segmentation and tracking in this case. A state of the art review for both tracking (monocular, multi-view, multi-object) and action representation and recognition (including gait) showed the limits of existing methods. Particularly in the field of view-point independent representation of actions, a problem addressed in a very limited number of works. It is also interesting to see that despite the important work in the domain, tracking is still a very hard problem and even most recent algorithms struggle to address situations that may seem “simple”. In this chapter we have explored the usage of self-similarity matrices in the multi-view hypotheses validation framework for tracking and we have seen that despite its relative simplicity, this representation method achieves promising results.

Chapter 7

Conclusion

In this thesis we explored novel techniques to produce foreground/background segmentations in multi-camera setups. Our work was motivated by the objective of proposing new segmentation approaches to better address the growing scenario of multi-camera capture.

In Chapter 3, we show that geometric consistency constraints can be enforced using a sparse set of 3D samples. These samples are used in a generative model and foreground/background color models are estimated using a MAP approach. The method shows that it is possible to achieve good segmentation results without resorting to 3D reconstruction, depth estimation or epipolar geometry. The resulting approach is extensively tested on various indoor and outdoor datasets.

The proposed framework is then extended to other color models and modalities (Chapter 4). The contributions at this level are twofold. First, we derive MAP estimation algorithms in two new cases using different color models based on variational GMMs and depth cameras. This shows the generality of the sparse sampling framework. Second, the different tests performed allow us a better understanding of the multi-view segmentation problem in general, and a clearer framing of the limitations of our probabilistic approach.

In Chapter 5, we take advantage of the experience built up during the first part of this work, to propose a new method for the multi-view segmentation problem. Segmentation is cast as a superpixel labelling problem over time space, and solved with graph-cuts. In the proposed graph-construction 3D samples still play an important role as a mean to enforce geometric consistency. Using superpixels allows us to directly reason on coherent image regions with a richer description (using color and texture). Thanks to this new model, more constraints of the segmentation problem are expressed. The method is also naturally extended to the time domain and tested on multi-view video datasets.

In Chapter 6, we explore the possibilities of tracking in uncalibrated multi-view scenarios. The main idea is that it is possible to identify and match actions and motions in different points of view even if the camera calibration is unknown. A model using self-similarity matrices is proposed and preliminary tests show promising results.

7.1 Perspectives

The perspectives regarding the work presented in this thesis can be divided in short term and long term objectives. In the short term, many improvements can be considered:

- Using motion in the context of multi-camera segmentation can be further explored. In this work, we have presented a limited usage of such information but it clearly has the potential to overcome situations where using appearance only is extremely challenging.
- Another direction that is also mentioned, is to identify the appropriate appearance descriptors to use for multi-view segmentation. In particular, it is not clear if oriented texture descriptors (using Gabor filters for example) would improve segmentation results.
- Further testing of the proposed validation framework for multi-view tracking must be performed. A crucial point at this level is to identify the best feature to be used for the frequential analysis of the movements.

In the long term, there are more challenging objectives that naturally derive from this work:

- One clear direction still left to explore is the multi-object segmentation. This problem has many challenges, in particular the difficulty to handle multi-object visibility between different viewpoints. This problem can be even more challenging if we consider situations where object appearance can strongly vary between the viewpoints. The 3D sampling approaches can be interestingly applied to this context but a more complex model must be developed.
- Multi-view segmentation in uncalibrated scenarios is also among the main directions that are still to be explored. In this case, the main challenge is to identify the appropriate way to use motion information to relate object regions between the views. Many challenges exist, in particular if we consider the general case where all the cameras are moving.

Appendix A

Derivation of histogram update equations

A.1 Scope

This appendix details the derivation of the update equations for the mixture weights π_k and the histograms H_b^F and $H_{i,b}$ representing the color models of the foreground and background in the views. These equations are numbered (3.26), (3.27) and (3.28).

The maximization of the Q-function, as defined by equation (3.25) of the submission, can be performed independently on the following terms:

$$P = \sum_{s,k} p_s^k \log \pi_k, \tag{A.1}$$

$$F = \sum_{i,s} p_s^{\mathrm{f}} \log p(I_i^s | \Theta_i, k_s = \mathrm{f}) \tag{A.2}$$

and

$$\begin{aligned} G \quad &= \sum_i \Bigg[\sum_s p_s^{\mathrm{b}_i} \log p(I_i^s | \Theta_i, k_s = \mathrm{b}_i) \\ &+ \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} \log(p(I_i^p | \Theta_i^B) \Bigg]. \end{aligned} \tag{A.3}$$

Each term will be maximized using a Lagrange multiplier.

A.2 Maximizing P

We want to find the set of parameters $\pi = \{\pi_k\}_{k \in \mathcal{K}}$ that maximizes the term P subject to

$$\sum_k \pi_k = 1. \tag{A.4}$$

If we define the Lagrange function as

$$\Lambda_P(\pi_1, \pi_2, ..., \lambda) = P + \lambda(\sum_k \pi_k - 1) \tag{A.5}$$

and solve the equations

$$\frac{\partial}{\partial \pi_k} \Lambda_P(\pi_1, \pi_2, ..., \lambda) = 0 \tag{A.6}$$

we obtain

$$\frac{\partial}{\partial \pi_k}(P + \lambda(\sum_k \pi_k - 1)) = 0$$

$$\sum_s \frac{p_s^k}{\pi_k} + \lambda = 0 \tag{A.7}$$

$$\sum_s p_s^k + \lambda \pi_k = 0.$$

Summing over k we get

$$\lambda = -\sum_{s,k} p_s^k = -\sum_s 1 = -N \quad (N \text{ number of samples}) \tag{A.8}$$

and replacing λ by its value we obtain

$$\pi_k = \frac{1}{N} \sum_s p_s^k. \tag{A.9}$$

A.3 Maximizing F

The foreground color model common to all views is represented by a normalized histogram H^F. Denoting by H_b^F the bin I_i^s falls into, we have:

$$p(I_i^s | \Theta_i, k_s = \mathrm{f}) = H_b^F \tag{A.10}$$

hence

$$F = \sum_{b,i} \sum_{s \in S : I_i^s \in b} p_s^{\mathrm{f}} log(H_b^F). \tag{A.11}$$

Using a Lagrange multiplier we look for the new parametrization of the foreground color model (H_b^F values) that maximizes F under the constraint $\sum_b H_b^F = 1$.

We define the Lagrange function

$$\Lambda_F = \sum_{b,i} \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{f}} log(H_b^F) + \lambda(\sum_b H_b^F - 1). \tag{A.12}$$

Solving the equations

$$\frac{\partial}{\partial H_b^F} \Lambda_F = 0 \tag{A.13}$$

we obtain

$$\frac{\partial}{\partial H_b^F}(F + \lambda(\sum_b H_b^F - 1)) = 0$$

$$\sum_i \sum_{s \in S: I_i^s \in b} \frac{p_s^f}{H_b^F} + \lambda = 0 \tag{A.14}$$

$$\sum_i \sum_{s \in S: I_i^s \in b} p_s^f + \lambda H_b^F = 0.$$

Summing over the histogram bins b, we get:

$$\lambda = -\sum_b \sum_i \sum_{s \in S: I_i^s \in b} p_s^f \tag{A.15}$$

and replacing λ by its value, we obtain the new parametrization of the foreground histogram

$$H_b^F = \frac{\sum_i \sum_{s \in \mathcal{S}: I_i^s \in b} p_s^{\mathrm{f}}}{\sum_{b'} \sum_i \sum_{s \in S: I_i^s \in b'} p_s^{\mathrm{f}}}. \tag{A.16}$$

A.4 Maximizing G

The background color model for view i is represented by a normalized histogram $H_{i,b}$. Similarly to the previous section, we denote by $H_{i,b}$ the bin I_i^s falls into:

$$p(I_i^s | \Theta_i, k_s = \mathrm{b}_i) = H_{i,b}. \tag{A.17}$$

Hence,

$$G = \sum_i \sum_b \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{b}_i} log(H_{i,b}) + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} \log(H_{i,b}) \tag{A.18}$$

In this equation, the per-view terms

$$G_i = \sum_b \sum_{s \in S: I_i^s \in b} p_s^{\mathrm{b}_i} log(H_{i,b}) + \lambda_i \sum_{p \in R_i^{\mathrm{Ext}}} \log(H_{i,b}) \tag{A.19}$$

can be maximized independently. If we denote by $N_{i,b}^{\text{Ext}}$ the number of pixels from R_i^{Ext} inside histogram bin b for view i, then

$$G_i = \sum_b \sum_{s \in S : I_i^s \in b} p_s^{\text{b}_i} log(H_{i,b}) + \lambda_i N_{i,b}^{\text{Ext}} log(H_{i,b}). \tag{A.20}$$

Using a Lagrange multiplier we look for the new parametrization of the background color model ($H_{i,b}$ values) for each view i. The new models maximize G_i under the constraint $\sum_b H_{i,b} = 1$.

We define the Lagrange function

$$\begin{aligned} \Lambda_{G_i} = & \sum_b \sum_{s \in S : I_i^s \in b} p_s^{\text{b}_i} log(H_{i,b}) + \lambda_i N_{i,b}^{\text{Ext}} log(H_{i,b}) \\ & + \lambda (\sum_b H_{i,b} - 1) \end{aligned} \tag{A.21}$$

and solving the equations

$$\frac{\partial}{\partial H_{i,b}} \Lambda_{G_i} = 0 \tag{A.22}$$

we obtain

$$\frac{\partial}{\partial H_{i,b}} (G_i + \lambda (\sum_b H_{i,b} - 1)) = 0$$

$$\begin{aligned} & \frac{\sum_{s \in S : I_i^s \in b} p_s^{\text{b}_i} + \lambda_i N_{i,b}^{\text{Ext}}}{H_{i,b}} + \lambda = 0 \\ & \sum_{s \in S : I_i^s \in b} p_s^{\text{b}_i} + \lambda_i N_{i,b}^{\text{Ext}} + \lambda H_{i,b} = 0. \end{aligned} \tag{A.23}$$

Summing over the histogram bins b

$$\lambda = - \sum_b \sum_{s \in S : I_i^s \in b} p_s^{\text{b}_i} + \lambda_i N_{i,b}^{\text{Ext}} \tag{A.24}$$

and replacing λ by its value, we get the new parametrization for the background histograms

$$H_{i,b} = \frac{\sum_{s \in S : I_i^s \in b} \left(p_s^{\text{b}_i} + \lambda_i N_{i,b}^{\text{Ext}} \right)}{\sum_{b'=1}^{B} \sum_{s \in S : I_i^s \in b'} \left(p_s^{\text{b}_i} + \lambda_i N_{i,b'}^{\text{Ext}} \right)}. \tag{A.25}$$

Bibliography

[1] Adarsh Kowdle, Sudipta N. Sinha, and Richard Szeliski. Multiple view object cosegmentation using appearance and stereo cues. In *ECCV*, 2012.

[2] Kalin Kolev, Thomas Brox, and Daniel Cremers. Fast joint estimation of silhouettes and dense 3d geometry from multiple images. *IEEE Trans. PAMI*, 34(3):493–505, 2011.

[3] Carsten Rother, Vladimir Kolmogorov, and Andrew Blake. "grabcut": interactive foreground extraction using iterated graph cuts. In *ACM SIGGRAPH*, 2004.

[4] Claude R. Brice and Claude L. Fennema. Scene analysis using regions. *Artificial Intelligence*, 1(3–4):205 – 226, 1970. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/0004-3702(70)90008-1. URL http://www.sciencedirect.com/science/article/pii/0004370270900081.

[5] Theodosios Pavlidis. Structural pattern recognition. 1977.

[6] Xue Bai, Jue Wang, David Simons, and Guillermo Sapiro. Video snapcut: Robust video object cutout using localized classifiers. *ACM Trans. Graph.*, 28(3):70:1–70:11, July 2009. ISSN 0730-0301. doi: 10.1145/1531326.1531376. URL http://doi.acm.org/10.1145/1531326.1531376.

[7] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *Int. J. Comput. Vision*, 80(2):189–210, November 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0107-3. URL http://dx.doi.org/10.1007/s11263-007-0107-3.

[8] Changchang Wu. Towards linear-time incremental structure from motion. In *3D Vision - 3DV 2013, 2013 International Conference on*, pages 127–134, June 2013. doi: 10.1109/3DV.2013.25.

[9] M. Granados, J. Tompkin, K. Kim, O. Grau, J. Kautz, and C. Theobalt. How not to be seen — object removal from videos of crowded scenes. *Computer Graphics Forum*, 31 (2pt1):219–228, 2012. ISSN 1467-8659. doi: 10.1111/j.1467-8659.2012.03000.x. URL http://dx.doi.org/10.1111/j.1467-8659.2012.03000.x.

[10] Alasdair Newson, Andrés Almansa, Matthieu Fradet, Yann Gousseau, and Patrick Pérez. Towards fast, generic video inpainting. In *Proceedings of the 10th European Conference on Visual Media Production*, CVMP '13, pages 7:1–7:8, New York, NY, USA, 2013. ACM. ISBN 978-1-4503-2589-9. doi: 10.1145/2534008.2534019. URL http://doi.acm.org/10.1145/2534008.2534019.

[11] Yong Jae Lee, Jaechul Kim, and Kristen Grauman. Key-segments for video object segmentation. In *In ICCV*, 2011.

[12] V.I Morariu and O.I Camps. Modeling correspondences for multi-camera tracking using nonlinear manifold learning and target dynamics. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 545–552, June 2006. doi: 10.1109/CVPR.2006.189.

[13] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. P.: Multi-commodity network flow for tracking multiple people. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2012.

[14] Cedric Cagniart. *Motion Capture of Deformable Surfaces in Multi-View Studios*. These, Université de Grenoble, July 2012. URL http://hal.inria.fr/tel-00771536.

[15] D. Weinland, E. Boyer, and R. Ronfard. Action recognition from arbitrary views using 3d exemplars. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7, Oct 2007. doi: 10.1109/ICCV.2007.4408849.

[16] Pingkun Yan, S.M. Khan, and M. Shah. Learning 4d action feature models for arbitrary view action recognition. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–7, June 2008. doi: 10.1109/CVPR.2008.4587737.

[17] Jingen Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3209–3216, June 2011. doi: 10.1109/CVPR.2011.5995729.

[18] Cedric Cagniart, Edmond Boyer, and Slobodan Ilic. Probabilistic deformable surface tracking from multiple videos. In *ECCV (4)*, pages 326–339, 2010.

[19] Jean-Yves Guillemaut and Adrian Hilton. Joint multi-layer segmentation and reconstruction for free-viewpoint video applications. *IJCV*, 93(1):73–100, 2011.

[20] Jian Sun, Weiwei Zhang, Xiaoou Tang, and Heung-Yeung Shum. Background cut. Association for Computing Machinery, Inc., March 2006. URL http://research.microsoft.com/apps/pubs/default.aspx?id=69408.

[21] Jean-Sébastien Franco and Edmond Boyer. Exact polyhedral visual hulls. In *British Machine Vision Conference (BMVC'03)*, volume 1, pages 329–338, Norwich, Royaume-Uni, 2003. URL http://hal.inria.fr/inria-00349075.

[22] Zheng Wu, A Thangali, S. Sclaroff, and M. Betke. Coupling detection and data association for multiple object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1948–1955, June 2012. doi: 10.1109/CVPR.2012.6247896.

[23] Estelle Duveau, Simon Courtemanche, Lionel Reveret, and Edmond Boyer. Cage-based Motion Recovery using Manifold Learning. In *3DIMPVT 2012 - Second International Conference on 3D Imaging, Modeling, Processing, Visualization and Transmission*, pages 206–213, Zürich, Switzerland, October 2012. IEEE. doi: 10.1109/3DIMPVT.2012.29.

[24] Xinxiao Wu and Yunde Jia. View-invariant action recognition using latent kernelized structural svm. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7576 of *Lecture Notes in Computer Science*, pages 411–424. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33714-7. doi: 10.1007/978-3-642-33715-4_30. URL http://dx.doi.org/10.1007/978-3-642-33715-4_30.

[25] Tae Hoon Kim, Kyoung-Mu Lee, and Sang-Uk Lee. Nonparametric higher-order learning for interactive segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3201–3208, June 2010. doi: 10.1109/CVPR.2010.5540078.

[26] Michael Kass, Andrew Witkin, and Demetri Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision*, pages 321–331, 1988.

[27] Yuri Boykov and Marie-Pierre Jolly. Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images. In *ICCV*, 2001.

[28] R. Malladi, J.A Sethian, and B.C. Vemuri. Shape modeling with front propagation: a level set approach. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 17 (2):158–175, 1995.

[29] Jianbo Shi and Jitendra Malik. Normalized cuts and image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 22(8):888–905, August 2000. ISSN 0162-8828.

[30] S. Alpert, M. Galun, A Brandt, and R. Basri. Image segmentation by probabilistic bottom-up aggregation and cue integration. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(2):315–327, Feb 2012.

[31] Pablo Arbelaez, Michael Maire, Charless Fowlkes, and Jitendra Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5): 898–916, May 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.161. URL http://dx.doi.org/10.1109/TPAMI.2010.161.

[32] L. Vincent and P. Soille. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 13(6):583–598, Jun 1991. ISSN 0162-8828.

[33] Gang Zeng and Long Quan. Silhouette extraction from multiple images of an unknown background. In *ACCV*, 2004.

[34] Wonwoo Lee, Woontack Woo, and Edmond Boyer. Identifying foreground from multiple images. In *ACCV (2)*, pages 580–589, 2007.

[35] Wonwoo Lee, Woontack Woo, and Edmond Boyer. Silhouette Segmentation in Multiple Views. *IEEE Trans. PAMI*, 33(7):1429–1441, 2010.

[36] N. D. F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image Vision Comput.*, 28 (1):4–25, 2010.

[37] N.D.F. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Automatic object segmentation from calibrated images. In *CVMP*, 2011.

[38] Carsten Rother, Tom Minka, Andrew Blake, and Vladimir Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. In *CVPR*, 2006.

[39] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. In *CVPR*, 2011.

[40] L. Mukherjee, V. Singh, and Jiming Peng. Scale invariant cosegmentation for image groups. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1881–1888, June 2011.

[41] J.C. Rubio, J. Serrat, A. Lopez, and N. Paragios. Unsupervised co-segmentation through region matching. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 749–756, June 2012.

[42] Robert M Haralick and Linda G Shapiro. Image segmentation techniques. In *1985 Technical Symposium East*, pages 2–9. International Society for Optics and Photonics, 1985.

[43] Daniel Cremers, Mikael Rousson, and Rachid Deriche. A review of statistical approaches to level set segmentation: Integrating color, texture, motion and shape. *International Journal of Computer Vision*, 72(2):195–215, 2007. ISSN 0920-5691.

[44] Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In *Proceedings of the 22Nd Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '95, pages 191–198. ACM, 1995.

[45] Tim McInerney and Demetri Terzopoulos. T-snakes: Topology adaptive snakes. *Medical image analysis*, 4(2):73–91, 2000.

[46] P. Perez, A Blake, and M. Gangnet. Jetstream: probabilistic contour extraction with particles. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 2, pages 524–531 vol.2, 2001. doi: 10.1109/ICCV.2001.937670.

[47] Y.Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary amp; region segmentation of objects in n-d images. In *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, volume 1, pages 105–112 vol.1, 2001. doi: 10.1109/ICCV.2001.937505.

[48] D. Greig, B. Porteous, and A. Seheult. Exact maximum a posteriori estimation for binary images. *Royal Journal on Statistical Society*, 51(2):271–279, 1989.

[49] S. Vicente, V. Kolmogorov, and C. Rother. Graph cut based image segmentation with connectivity priors. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587440.

[50] V. Lempitsky and Y. Boykov. Global optimization for shape fitting. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383293.

[51] AK. Sinop and L. Grady. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408927.

[52] Leo Grady and Christopher V. Alvino. Reformulating and optimizing the mumford-shah functional on a graph - a faster, lower energy solution. In *ECCV*, pages 248–261, 2008.

[53] C. Couprie, L. Grady, L. Najman, and H. Talbot. Power watershed: A unifying graph-based optimization framework. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(7):1384–1399, July 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.200.

[54] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(11):1222–1239, Nov 2001. ISSN 0162-8828. doi: 10.1109/34.969114.

[55] Andrew Delong, Anton Osokin, HossamN. Isack, and Yuri Boykov. Fast approximate energy minimization with label costs. *International Journal of Computer Vision*, 96(1): 1–27, 2012. ISSN 0920-5691. doi: 10.1007/s11263-011-0437-z. URL http://dx.doi.org/10.1007/s11263-011-0437-z.

[56] M. Klodt and D. Cremers. A convex framework for image segmentation with moment constraints. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 2236–2243, Nov 2011.

[57] Claudia Nieuwenhuis and Daniel Cremers. Spatially varying color distributions for interactive multilabel segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 35(5): 1234–1247, May 2013. ISSN 0162-8828. doi: 10.1109/TPAMI.2012.183. URL http://dx.doi.org/10.1109/TPAMI.2012.183.

[58] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *Int. J. Comput. Vision*, 59(2):167–181, September 2004.

[59] Christopher M. Bishop. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. 2006.

[60] Yi Ma, Senior Member, Harm Derksen, Wei Hong, John Wright, and Student Member. Segmentation of multivariate mixed data via lossy coding and compression. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 3, 2007.

[61] D. Comaniciu and P. Meer. Mean shift: a robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, May 2002. ISSN 0162-8828. doi: 10.1109/34.1000236.

[62] S. Paris and F. Durand. A topological approach to hierarchical segmentation using mean shift. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007.

[63] E. Sharon, M. Galun, D. Sharon, R. Basri, and A. Brandt. Hierarchy and adaptivity in segmenting visual scenes. *Nature*, 442:810–813, March 2006. doi: 10.1038/nature04977.

[64] L. Itti, C. Koch, and E. Niebur. A model of saliency-based visual attention for rapid scene analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 20 (11):1254–1259, Nov 1998. ISSN 0162-8828. doi: 10.1109/34.730558.

[65] Stas Goferman, Lihi Zelnik-Manor, and Ayellet Tal. Context-aware saliency detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(10):1915–1926, Oct 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.272.

[66] Tie Liu, Zejian Yuan, Jian Sun, Jingdong Wang, Nanning Zheng, Xiaoou Tang, and Heung-Yeung Shum. Learning to detect a salient object. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):353–367, Feb 2011. ISSN 0162-8828.

[67] I Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 36(2):222–234, Feb 2014. ISSN 0162-8828.

[68] Yong Jae Lee, Jaechul Kim, and K. Grauman. Key-segments for video object segmentation. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1995–2002, Nov 2011. doi: 10.1109/ICCV.2011.6126471.

[69] Christopher Wen, Ali Azarbayejani, Trevor Darrell, and Alex Pentland. Pfinder: Real-time tracking of the human body. *IEEE Trans. PAMI*, 19(7):780–785, 1997.

[70] Chris Stauffer and Eric Grimson. Adaptive background mixture models for real-time tracking. In *CVPR*, 1999.

[71] K Toyama, J Krumm, B Brumitt, and B Meyers. Wallflower: principles and practice of background maintenance. In *ICCV*, 1999.

[72] J. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, Aug 2000. ISSN 0162-8828. doi: 10.1109/34.868688.

[73] Mukund Narasimhan and Jeff A. Bilmes. A submodular-supermodular procedure with applications to discriminative structure learning. In *UAI*, pages 404–412, 2005.

[74] L. Mukherjee, V. Singh, and C.R. Dyer. Half-integrality based algorithms for cosegmentation of images. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2028–2035, June 2009. doi: 10.1109/CVPR.2009.5206652.

[75] Dorit S. Hochbaum and Vikas Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009.

[76] Sara Vicente, Vladimir Kolmogorov, and Carsten Rother. Cosegmentation revisited: Models and optimization. In *Proceedings of the 11th European Conference on Computer Vision: Part II*, ECCV'10, 2010.

[77] Kai-Yueh Chang, Tyng-Luh Liu, and Shang-Hong Lai. From co-saliency to co-segmentation: An efficient and fully unsupervised energy minimization model. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2129–2136, June 2011. doi: 10.1109/CVPR.2011.5995415.

[78] Armand Joulin, Francis R. Bach, and Jean Ponce. Discriminative clustering for image co-segmentation. In *CVPR*, 2010.

[79] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. icoseg: Interactive co-segmentation with intelligent scribble guidance. In *CVPR*, 2010.

[80] Adarsh Kowdle, Dhruv Batra, Wen-Chao Chen, and Tsuhan Chen. imodel: Interactive co-segmentation for object of interest 3d modeling. In *Trends and Topics in Computer Vision*. 2012.

[81] A.C. Gallagher and Tsuhan Chen. Clothing cosegmentation for recognizing people. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008.

[82] Dhruv Batra, Adarsh Kowdle, Devi Parikh, Jiebo Luo, and Tsuhan Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *Int. J. Comput. Vision*, 93(3), 2011.

[83] John M. Winn and Nebojsa Jojic. Locus: Learning object classes with unsupervised segmentation. In *ICCV*, 2005.

[84] J. Carreira and C. Sminchisescu. Constrained parametric min-cuts for automatic object segmentation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010.

[85] B. Alexe, T. Deselaers, and V. Ferrari. What is an object? In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 73–80, June 2010. doi: 10.1109/CVPR.2010.5540226.

[86] E. Kim, Hongsheng Li, and Xiaolei Huang. A hierarchical image clustering cosegmentation framework. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 686–693, June 2012. doi: 10.1109/CVPR.2012.6247737.

[87] Jifeng Dai, Ying Nian Wu, Jie Zhou, and Song-Chun Zhu. Cosegmentation and cosketch by unsupervised learning. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV'13, 2013.

[88] A Faktor and M. Irani. Co-segmentation by composition. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1297–1304, Dec 2013. doi: 10.1109/ICCV.2013.164.

[89] Alon Faktor and Michal Irani. "clustering by composition": Unsupervised discovery of image categories. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VII*, ECCV'12, pages 474–487, Berlin, Heidelberg, 2012. Springer-Verlag. ISBN 978-3-642-33785-7. doi: 10.1007/978-3-642-33786-4_35. URL http://dx.doi.org/10.1007/978-3-642-33786-4_35.

[90] Oren Boiman and Michal Irani. Similarity by composition. In *NIPS*, pages 177–184, 2006.

[91] Tammy Riklin-Raviv, Nir Sochen, and Nahum Kiryati. Shape-based mutual segmentation. *International Journal of Computer Vision*, 79(3):231–245, 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0115-3. URL http://dx.doi.org/10.1007/s11263-007-0115-3.

[92] Claudia Nieuwenhuis, Evgeny Strekalovskiy, and Daniel Cremers. Proportion priors for image sequence segmentation. In *ICCV*, pages 2328–2335, 2013.

[93] Jean-Sébastien Franco and Edmond Boyer. Fusion of Multi-View Silhouette Cues Using a Space Occupancy Grid. In *ICCV* , 2005.

[94] Dan Snow, Paul Viola, and Ramin Zabih. Exact voxel occupancy with graph cuts. In *CVPR*, 2000.

[95] K.N. Kutulakos and S.M. Seitz. A theory of shape by space carving. In *ICCV*, 1999.

[96] C. Reinbacher, M. Rüther, and H. Bischof. Fast variational multi-view segmentation through backprojection of spatial constraints. *Image and Vision Computing*, 30(11):797–807, 2012.

[97] N.D.F. Campbell, G. Vogiatzis, C. Hernández, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1):14 – 25, 2010. ISSN 0262-8856. doi: http://dx.doi.org/10.1016/j.imavis.2008.09.005. URL http://www.sciencedirect.com/science/article/pii/S026288560800200X.

[98] Tobias Feldmann, Lars Dießelberg, and Annika Wörner. Adaptive foreground/background segmentation using multiview silhouette fusion. In *DAGM-Symposium*, 2009.

[99] J. Gallego, J. Salvador, J.R. Casas, and M. Pardàs. Joint multi-view foreground segmentation and 3d reconstruction with tolerance loop. In *ICIP*, 2011.

[100] Edmond Boyer. On using silhouettes for camera calibration. In *In Proceedings of the 7th Asian Conference on Computer Vision (ACCV*, pages 1–10. Springer, 2006.

[101] M. Sormann, C. Zach, and K. Karner. Graph cut based multiple view segmentation for 3d reconstruction. In *3DPVT*, 2006.

[102] Muhammad Sarim, Adrian Hilton, Jean-Yves Guillemaut, Hansung Kim, and Takeshi Takai. Wide-baseline multi-view video segmentation for 3d reconstruction. In *3DVP*, 2010.

[103] A Levin, D. Lischinski, and Y. Weiss. A closed-form solution to natural image matting. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):228–242, Feb 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1177.

[104] J. Y Guillemaut, A Hilton, J. Starck, J. Kilner, and O. Grau. A bayesian framework for simultaneous matting and 3d reconstruction. In *3-D Digital Imaging and Modeling, 2007. 3DIM '07. Sixth International Conference on*, pages 167–176, Aug 2007. doi: 10.1109/3DIM.2007.3.

[105] J. Y Guillemaut, J. Kilner, and A Hilton. Robust graph-cut scene segmentation and reconstruction for free-viewpoint video of complex dynamic scenes. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 809–816, Sept 2009. doi: 10.1109/ICCV.2009.5459299.

[106] Jean-Sébastien Franco, Edmond Boyer, et al. Exact polyhedral visual hulls. In *British Machine Vision Conference (BMVC'03)*, volume 1, pages 329–338, 2003.

[107] Jeff Bilmes. A gentle tutorial of the EM algorithm and its application to parameter estimation for Gaussian mixture and hidden Markov models. Technical report, ICSI, 1997.

[108] A. Djelouah, J-S. Franco, E. Boyer, F. Leclerc, and P. Pérez. N-Tuple Color Segmentation for Multi-View Silhouette Extraction. In *ECCV*, 2012.

[109] M. Lhuillier and Long Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *IEEE Trans. PAMI*, 27(3):418–433, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.44.

[110] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000.

[111] Viet Quoc Pham, Keita Takahashi, and Takeshi Naemura. Foreground-background segmentation using iterated distribution matching. In *CVPR*, 2011.

[112] Viet-Quoc Pham, K. Takahashi, and T. Naemura. Foreground-background segmentation using iterated distribution matching. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2113–2120, June 2011. doi: 10.1109/CVPR.2011.5995356.

[113] Andrew Blake, Carsten Rother, Matthew Brown, Patrick Perez, and Philip Torr. Interactive image segmentation using an adaptive gmmrf model. In *Proc. European Conference in Computer Vision (ECCV)*. Springer-Verlag, May 2004. URL http://research.microsoft.com/apps/pubs/default.aspx?id=67898.

[114] C. Biernacki, G. Celeux, and G. Govaert. Assessing a mixture model for clustering with the integrated completed likelihood. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(7):719–725, Jul 2000. ISSN 0162-8828. doi: 10.1109/34.865189.

[115] Mario A T Figueiredo and AK. Jain. Unsupervised learning of finite mixture models. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(3):381–396, March 2002. ISSN 0162-8828. doi: 10.1109/34.990138.

[116] A. Corduneanu and Christopher M. Bishop. Variational bayesian model selection for mixture distributions. In *Proceedings Eighth International Conference on Artificial Intelligence and Statistics*, page 27–34. Morgan Kaufmann, January 2001. URL http://research.microsoft.com/apps/pubs/default.aspx?id=67239.

[117] Vladimir Kolmogorov, Antonio Criminisi, Andrew Blake, Geoffrey Cross, and Carsten Rother. Bi-layer segmentation of binocular stereo video. In *CVPR*, 2005.

[118] L. Wang, C. Zhang, R. Yang, and C. Zhang. Tofcut: Towards robust real-time foreground extraction using a time-of-flight camera. In *3DPVT*, 2010.

[119] Li Guan, Jean-Sébastien Franco, and Marc Pollefeys. 3D Object Reconstruction with Heterogeneous Sensor Data. In *3DPVT*, 2008.

[120] Y. M. Kim, Christian Theobalt, J. Diebel, J. Kosecka, B. Micusik, and S. Thrun. Multi-view image and tof sensor fusion for dense 3d reconstruction. In *3DIM*, 2009.

[121] Ryan Crabb, Colin Tracey, Akshaya Puranik, and James Davis. Real-time foreground segmentation via range and color imaging. *CVPR Workshop*, 2008.

[122] Jeroen van Baar, Paul A. Beardsley, Marc Pollefeys, and Markus H. Gross. Interactive video segmentation supported by multiple modalities, with an application to depth maps. In *3DTV-Conference*, pages 1–4, 2012.

[123] Jitendra Malik, Serge Belongie, Thomas Leung, and Jianbo Shi. Contour and texture analysis for image segmentation. *Int. J. Comput. Vision*, 43(1):7–27, June 2001. ISSN 0920-5691. doi: 10.1023/A:1011174803800. URL http://dx.doi.org/10.1023/A:1011174803800.

[124] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurélien Lucchi, Pascal Fua, and Sabine Susstrunk. SLIC Superpixels Compared to State-of-the-art Superpixel Methods. *IEEE PAMI*, 2012.

[125] Michael Van den Bergh, Xavier Boix, Gemma Roig, Benjamin de Capitani, and Luc Van Gool. Seeds: Superpixels extracted via energy-driven sampling. In Andrew Fitzgibbon, Svetlana Lazebnik, Pietro Perona, Yoichi Sato, and Cordelia Schmid, editors, *Computer Vision – ECCV 2012*, volume 7578 of *Lecture Notes in Computer Science*, pages 13–26. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-33785-7. doi: 10.1007/978-3-642-33786-4_2. URL http://dx.doi.org/10.1007/978-3-642-33786-4_2.

[126] A Rabinovich, S. Belongie, T. Lange, and J.M. Buhmann. Model order selection and cue combination for image segmentation. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 1130–1137, June 2006. doi: 10.1109/CVPR.2006.186.

[127] Xuming He, Richard S. Zemel, and Debajyoti Ray. Learning and incorporating top-down cues in image segmentation. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part I*, ECCV'06, pages 338–351, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33832-2, 978-3-540-33832-1. doi: 10.1007/11744023_27. URL http://dx.doi.org/10.1007/11744023_27.

[128] P. Arbelaez and L. Cohen. Constrained image segmentation from hierarchical boundaries. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587492.

[129] Tinghuai Wang and J. Collomosse. Probabilistic motion diffusion of labeling priors for coherent video segmentation. *Multimedia, IEEE Transactions on*, 14(2):389–400, April 2012. ISSN 1520-9210. doi: 10.1109/TMM.2011.2177078.

[130] Vijay Badrinarayanan, Ignas Budvytis, and Roberto Cipolla. Mixture of trees probabilistic graphical model for video segmentation. *International Journal of Computer Vision*, pages 1–16, 2013. ISSN 0920-5691. doi: 10.1007/s11263-013-0673-5. URL http://dx.doi.org/10.1007/s11263-013-0673-5.

[131] Andrew Blake and M. Isard. *Active Contours: The Application of Techniques from Graphics,Vision,Control Theory and Statistics to Visual Tracking of Shapes in Motion*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1st edition, 1998. ISBN 3540762175.

[132] Yung-Yu Chuang, Aseem Agarwala, Brian Curless, David H. Salesin, and Richard Szeliski. Video matting of complex scenes. *ACM Transactions on Graphics*, 21(3): 243–248, July 2002. Sepcial Issue of the SIGGRAPH 2002 Proceedings.

[133] Aseem Agarwala, Aaron Hertzmann, David H. Salesin, and Steven M. Seitz. Keyframe-based tracking for rotoscoping and animation. *ACM Trans. Graph.*, 23(3):584–591, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015764. URL http://doi.acm.org/10.1145/1015706.1015764.

[134] Yin Li, Jian Sun, and Heung-Yeung Shum. Video object cut and paste. *ACM Trans. Graph.*, 24(3):595–600, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073234. URL http://doi.acm.org/10.1145/1073204.1073234.

[135] Jue Wang, Pravin Bhat, R. Alex Colburn, Maneesh Agrawala, and Michael F. Cohen. Interactive video cutout. *ACM Trans. Graph.*, 24(3):585–594, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073233. URL http://doi.acm.org/10.1145/1073204.1073233.

[136] Xue Bai and G. Sapiro. A geodesic framework for fast interactive image and video segmentation and matting. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, Oct 2007. doi: 10.1109/ICCV.2007.4408931.

[137] Christopher J. Armstrong, Brian L. Price, and William A. Barrett. Interactive segmentation of image volumes with live surface, 2007.

[138] Yanchao Yang and G. Sundaramoorthi. Modeling self-occlusions in dynamic shape and appearance tracking. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 201–208, Dec 2013. doi: 10.1109/ICCV.2013.32.

[139] C.R. Wren, A Azarbayejani, T. Darrell, and AP. Pentland. Pfinder: real-time tracking of the human body. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19 (7):780–785, Jul 1997. ISSN 0162-8828. doi: 10.1109/34.598236.

[140] K. Toyama, J. Krumm, B. Brumitt, and B. Meyers. Wallflower: principles and practice of background maintenance. In *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, volume 1, pages 255–261 vol.1, 1999. doi: 10.1109/ICCV.1999.791228.

[141] Zoran Zivkovic and Ferdinand van der Heijden. Efficient adaptive density estimation per image pixel for the task of background subtraction. *Pattern Recogn. Lett.*, 27(7): 773–780, May 2006. ISSN 0167-8655. doi: 10.1016/j.patrec.2005.11.005. URL http://dx.doi.org/10.1016/j.patrec.2005.11.005.

[142] A Criminisi, G. Cross, A Blake, and V. Kolmogorov. Bilayer segmentation of live video. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 53–60, June 2006. doi: 10.1109/CVPR.2006.69.

[143] Julien Pilet, Christoph Strecha, and Pascal Fua. Making background subtraction robust to sudden illumination changes. In *In Proc. European Conf. on Computer Vision*, 2008.

[144] M. Reso, J. Jachalsky, B. Rosenhahn, and J. Ostermann. Temporally consistent superpixels. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 385–392, Dec 2013. doi: 10.1109/ICCV.2013.55.

[145] M. Van den Bergh, G. Roig, X. Boix, S. Manen, and L. Van Gool. Online video seeds for temporal window objectness. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 377–384, Dec 2013. doi: 10.1109/ICCV.2013.54.

[146] Matthias Grundmann, Vivek Kwatra, Mei Han, and Irfan Essa. Efficient hierarchical graph based video segmentation. *IEEE CVPR*, 2010.

[147] Ignas Budvytis, Vijay Badrinarayanan, and Roberto Cipolla. Mot - mixture of trees probabilistic graphical model for video segmentation. In *Proceedings of the British Machine Vision Conference*, pages 72.1–72.11. BMVA Press, 2012. ISBN 1-901725-46-4. doi: http://dx.doi.org/10.5244/C.26.72.

[148] Wei-Te Li, Haw-Shiuan Chang, Kuo-Chin Lien, Hui-Tang Chang, and Y.F. Wang. Exploring visual and motion saliency for automatic video object extraction. *Image Processing, IEEE Transactions on*, 22(7):2600–2610, July 2013. ISSN 1057-7149. doi: 10.1109/TIP.2013.2253483.

[149] Thomas Brox and Jitendra Malik. Object segmentation by long term analysis of point trajectories. In *Proceedings of the 11th European Conference on Computer Vision: Part V*, ECCV'10, pages 282–295, Berlin, Heidelberg, 2010. Springer-Verlag. ISBN 3-642-15554-5, 978-3-642-15554-3. URL http://dl.acm.org/citation.cfm?id=1888150.1888173.

[150] Wei-Chen Chiu and Mario Fritz. Multi-class video co-segmentation with a generative multi-video model. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013.

[151] Yuri Boykov and Vladimir Kolmogorov. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision. *IEEE PAMI*, 2004.

[152] Abdelaziz Djelouah, Jean-Sébastien Franco, Edmond Boyer, Francois Le Clerc, and Patrick Pérez. Multi-View Object Segmentation in Space and Time. In *ICCV'13 - International Conference On Computer Vision*, December 2013. URL http://hal.inria.fr/hal-00873544.

[153] J. Winn, A Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 2, pages 1800–1807 Vol. 2, Oct 2005. doi: 10.1109/ICCV.2005.171.

[154] N. Hasler, B. Rosenhahn, T. Thormählen, M. Wand, J. Gall, and H. P. Seidel. Markerless motion capture with unsynchronized moving cameras. In *CVPR*, 2009.

[155] S. Hare, A Saffari, and P. H S Torr. Struck: Structured output tracking with kernels. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 263–270, Nov 2011. doi: 10.1109/ICCV.2011.6126251.

[156] Dapeng Chen, Zejian Yuan, Yang Wu, Geng Zhang, and Nanning Zheng. Constructing adaptive complex cells for robust visual tracking. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 1113–1120, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.142. URL http://dx.doi.org/10.1109/ICCV.2013.142.

[157] Helmut Grabner, Christian Leistner, and Horst Bischof. Semi-supervised on-line boosting for robust tracking. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 234–247, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_19. URL http://dx.doi.org/10.1007/978-3-540-88682-2_19.

[158] Kaihua Zhang and Huihui Song. Real-time visual tracking via online weighted multiple instance learning. *Pattern Recogn.*, 46(1):397–411, January 2013. ISSN 0031-3203. doi: 10.1016/j.patcog.2012.07.013. URL http://dx.doi.org/10.1016/j.patcog.2012.07.013.

[159] B. Babenko, Ming-Hsuan Yang, and S. Belongie. Visual tracking with online multiple instance learning. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 983–990, June 2009. doi: 10.1109/CVPR.2009.5206737.

[160] L. Leal-Taixe, G. Pons-Moll, and B. Rosenhahn. Branch-and-price global optimization for multi-view multi-target tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1987–1994, June 2012.

[161] M. Hofmann, D. Wolf, and G. Rigoll. Hypergraphs for joint multi-view reconstruction and multi-object tracking. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3650–3657, June 2013. doi: 10.1109/CVPR.2013.468.

[162] F. Fleuret, J. Berclaz, R. Lengagne, and P. Fua. Multicamera people tracking with a probabilistic occupancy map. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 30(2):267–282, Feb 2008. ISSN 0162-8828. doi: 10.1109/TPAMI.2007.1174.

[163] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, Sept 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.21.

[164] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(1):172–185, Jan 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2010.68.

[165] Bruce D. Lucas and Takeo Kanade. An iterative image registration technique with an application to stereo vision. In *Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2*, IJCAI'81, pages 674–679, San Francisco, CA, USA, 1981. Morgan Kaufmann Publishers Inc. URL http://dl.acm.org/citation.cfm?id=1623264.1623280.

[166] I Matthews, T. Ishikawa, and S. Baker. The template update problem. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 26(6):810–815, June 2004. ISSN 0162-8828. doi: 10.1109/TPAMI.2004.16.

[167] Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *Int. J. Comput. Vision*, 56(3):221–255, February 2004. ISSN 0920-5691. doi: 10.1023/B:VISI.0000011205.11775.fd. URL http://dx.doi.org/10.1023/B:VISI.0000011205.11775.fd.

[168] N. Alt, S. Hinterstoisser, and N. Navab. Rapid selection of reliable templates for visual tracking. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1355–1362, June 2010. doi: 10.1109/CVPR.2010.5539812.

[169] Xue Mei and Haibin Ling. Robust visual tracking using l1 minimization. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1436–1443, Sept 2009. doi: 10.1109/ICCV.2009.5459292.

[170] Xue Mei, Haibin Ling, Yi Wu, E. Blasch, and Li Bai. Minimum error bounded efficient l1 tracker with occlusion detection. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1257–1264, June 2011. doi: 10.1109/CVPR.2011.5995421.

[171] Yi Wu, Haibin Ling, Jingyi Yu, Feng Li, Xue Mei, and Erkang Cheng. Blurred target tracking by blur-driven tracker. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1100–1107, Nov 2011.

[172] D.M. Gavrila. Pedestrian detection from a moving vehicle. In David Vernon, editor, *Computer Vision — ECCV 2000*, volume 1843 of *Lecture Notes in Computer Science*, pages 37–49. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67686-7. doi: 10.1007/3-540-45053-X_3. URL http://dx.doi.org/10.1007/3-540-45053-X_3.

[173] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In Aleš Leonardis, Horst Bischof, and Axel Pinz, editors, *Computer Vision – ECCV 2006*, volume 3951 of *Lecture Notes in Computer Science*, pages 404–417. Springer Berlin Heidelberg, 2006. ISBN 978-3-540-33832-1. doi: 10.1007/11744023_32. URL http://dx.doi.org/10.1007/11744023_32.

[174] Vittorio Ferrari, Tinne Tuytelaars, and Luc Van Gool. Object detection by contour segment networks. In *Proceeding of the European Conference on Computer Vision*, volume 3953 of *LNCS*, pages 14–28. Elsevier, June 2006.

[175] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. In Cordelia Schmid, Stefano Soatto, and Carlo Tomasi, editors, *International Conference on Computer Vision & Pattern Recognition*, volume 2, pages 886–893, INRIA Rhône-Alpes, ZIRST-655, av. de l'Europe, Montbonnot-38334, June 2005. URL http://lear.inrialpes.fr/pubs/2005/DT05.

[176] Theo Gevers, Joost Van De Weijer, and Harro Stokman. Color feature detection. In Rastislav Lukac and Konstantinos N. Plataniotis, editors, *Color image processing: methods and applications*, volume 9, pages 203–226. CRC press, October 2006. ISBN 978-0-8493-9774-5. doi: 10.1201/9781420009781. URL http://hal.inria.fr/inria-00548685. GVS06 GVS06.

[177] AE. Abdel-Hakim and AA Farag. Csift: A sift descriptor with color invariant characteristics. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 1978–1983, 2006. doi: 10.1109/CVPR.2006.95.

[178] B.S. Manjunath and W.Y. Ma. Texture features for browsing and retrieval of image data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 18(8):837–842, Aug 1996. ISSN 0162-8828. doi: 10.1109/34.531803.

[179] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(7):971–987, Jul 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1017623.

[180] I Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587756.

[181] Paul Scovanner, Saad Ali, and Mubarak Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proceedings of the 15th International Conference on Multimedia*, MULTIMEDIA '07, pages 357–360, New York, NY, USA, 2007. ACM. ISBN 978-1-59593-702-5. doi: 10.1145/1291233.1291311. URL http://doi.acm.org/10.1145/1291233.1291311.

[182] Shu Wang, Huchuan Lu, Fan Yang, and Ming-Hsuan Yang. Superpixel tracking. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 1323–1330, Nov 2011. doi: 10.1109/ICCV.2011.6126385.

[183] A Adam, E. Rivlin, and I Shimshoni. Robust fragments-based tracking using the integral histogram. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 798–805, June 2006. doi: 10.1109/CVPR.2006.256.

[184] Wei He, T. Yamashita, Hongtao Lu, and Shihong Lao. Surf tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 1586–1592, Sept 2009. doi: 10.1109/ICCV.2009.5459360.

[185] David A. Ross, Jongwoo Lim, Ruei-Sung Lin, and Ming-Hsuan Yang. Incremental learning for robust visual tracking. *Int. J. Comput. Vision*, 77(1-3):125–141, May 2008. ISSN 0920-5691. doi: 10.1007/s11263-007-0075-7. URL http://dx.doi.org/10.1007/s11263-007-0075-7.

[186] Zdenek Kalal, Krystian Mikolajczyk, and Jiri Matas. Tracking-learning-detection. *IEEE Trans. Pattern Anal. Mach. Intell.*, 34(7):1409–1422, July 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.239. URL http://dx.doi.org/10.1109/TPAMI.2011.239.

[187] Ming Yang, Ying Wu, and Gang Hua. Context-aware visual tracking. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(7):1195–1209, July 2009. ISSN 0162-8828. doi: 10.1109/TPAMI.2008.146.

[188] H. Grabner, J. Matas, L. Van Gool, and P. Cattin. Tracking the invisible: Learning where the object might be. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1285–1292, June 2010. doi: 10.1109/CVPR.2010.5539819.

[189] Thang Ba Dinh, Nam Vo, and G. Medioni. Context tracker: Exploring supporters and distracters in unconstrained environments. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1177–1184, June 2011. doi: 10.1109/CVPR.2011.5995733.

[190] Chang Huang, Bo Wu, and Ramakant Nevatia. Robust object tracking by hierarchical association of detection responses. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, pages 788–801, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88685-3. doi: 10.1007/978-3-540-88688-4_58. URL http://dx.doi.org/10.1007/978-3-540-88688-4_58.

[191] AG.A Perera, C. Srinivas, A Hoogs, G. Brooksby, and Wensheng Hu. Multi-object tracking through simultaneous long occlusions and split-merge conditions. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 1, pages 666–673, June 2006. doi: 10.1109/CVPR.2006.195.

[192] Hao Jiang, S. Fels, and J.J. Little. A linear programming approach for multiple object tracking. In *Computer Vision and Pattern Recognition, 2007. CVPR '07. IEEE Conference on*, pages 1–8, June 2007. doi: 10.1109/CVPR.2007.383180.

[193] Li Zhang, Yuan Li, and R. Nevatia. Global data association for multi-object tracking using network flows. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8, June 2008. doi: 10.1109/CVPR.2008.4587584.

[194] Stefan Roth. Discrete-continuous optimization for multi-target tracking. In *Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, CVPR '12, pages 1926–1933, Washington, DC, USA, 2012. IEEE Computer Society. ISBN 978-1-4673-1226-4. URL http://dl.acm.org/citation.cfm?id=2354409.2354941.

[195] R.T. Collins. Multitarget data association with higher-order motion models. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1744–1751, June 2012. doi: 10.1109/CVPR.2012.6247870.

[196] Chris J. Needham and Roger D. Boyle. Tracking multiple sports players through occlusion, congestion and scale. In *British Machine Vision Conference*, pages 93–102, 2001.

[197] Yizheng Cai, Nando de Freitas, and James J. Little. Robust visual tracking for multiple targets. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part IV*, ECCV'06, pages 107–118, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33838-1, 978-3-540-33838-3. doi: 10.1007/11744085_9. URL http://dx.doi.org/10.1007/11744085_9.

[198] Wei-Lwun Lu, Jo-Anne Ting, K.P. Murphy, and J.J. Little. Identifying players in broadcast sports videos using conditional random fields. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 3249–3256, June 2011.

[199] Junliang Xing, Haizhou Ai, Liwei Liu, and Shihong Lao. Multiple player tracking in sports video: A dual-mode two-way bayesian inference approach with progressive observation modeling. *Image Processing, IEEE Transactions on*, 20(6):1652–1667, June 2011. ISSN 1057-7149. doi: 10.1109/TIP.2010.2102045.

[200] S. Pellegrini, A Ess, K. Schindler, and L. Van Gool. You'll never walk alone: Modeling social behavior for multi-target tracking. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 261–268, Sept 2009. doi: 10.1109/ICCV.2009.5459260.

[201] W. Brendel, M. Amer, and S. Todorovic. Multiobject tracking as maximum weight independent set. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 1273–1280, June 2011. doi: 10.1109/CVPR.2011.5995395.

[202] Jingchen Liu, P. Carr, R.T. Collins, and Yanxi Liu. Tracking sports players with context-conditioned motion models. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1830–1837, June 2013. doi: 10.1109/CVPR.2013.239.

[203] J. Krumm, S. Harris, B. Meyers, B. Brumitt, M. Hale, and S. Shafer. Multi-camera multi-person tracking for easyliving. In *Visual Surveillance, 2000. Proceedings. Third IEEE International Workshop on*, pages 3–10, 2000. doi: 10.1109/VS.2000.856852.

[204] Ivana Mikic, Simone Santini, and Ramesh Jain. Video processing and integration from multiple cameras. In *In Proceedings of the 1998 Image Understanding Workshop, Morgan-Kaufman*, pages 183–187. Cambridge University Press, 1998.

[205] J. Black, T. Ellis, and P. Rosin. Multi view image surveillance and tracking. In *Motion and Video Computing, 2002. Proceedings. Workshop on*, pages 169–174, Dec 2002. doi: 10.1109/MOTION.2002.1182230.

[206] K. Otsuka and N. Mukawa. Multiview occlusion analysis for tracking densely populated objects based on 2-d visual angles. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 1, pages I–90–I–97 Vol.1, June 2004. doi: 10.1109/CVPR.2004.1315018.

[207] Anurag Mittal and Larry S. Davis. M2tracker: A multi-view approach to segmenting and tracking people in a cluttered scene using region-based stereo. In *Proceedings of the 7th European Conference on Computer Vision-Part I*, ECCV '02, pages 18–36, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-43745-2. URL http://dl.acm.org/citation.cfm?id=645315.649473.

[208] D. Beymer. Person counting using stereo. In *Human Motion, 2000. Proceedings. Workshop on*, pages 127–133, 2000. doi: 10.1109/HUMO.2000.897382.

[209] D.B. Yang, H.H. Gonzalez-Banos, and L.J. Guibas. Counting people in crowds with a real-time network of simple image sensors. In *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*, pages 122–129 vol.1, Oct 2003. doi: 10.1109/ICCV.2003.1238325.

[210] J. Berclaz, F. Fleuret, E. Turetken, and P. Fua. Multiple object tracking using k-shortest paths optimization. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(9):1806–1819, Sept 2011. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.21.

[211] C.J. Veenman, M. J T Reinders, and E. Backer. Resolving motion correspondence for densely moving points. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 23(1):54–72, Jan 2001. ISSN 0162-8828.

[212] K. Shafique and M. Shah. A noniterative greedy algorithm for multiframe point correspondence. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(1): 51–65, Jan 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.1.

[213] R. Hamid, R.K. Kumar, M. Grundmann, Kihwan Kim, I Essa, and J. Hodgins. Player localization using multiple static cameras for sports visualization. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 731–738, June 2010. doi: 10.1109/CVPR.2010.5540142.

[214] Cen Rao, Alper Yilmaz, and Mubarak Shah. View-invariant representation and recognition of actions. *Int. J. Comput. Vision*, 50(2):203–226, November 2002. ISSN 0920-5691. doi: 10.1023/A:1020350100748. URL http://dx.doi.org/10.1023/A:1020350100748.

[215] T. Syeda-Mahmood, A Vasilescu, and S. Sethi. Recognizing action events from multiple viewpoints. In *Detection and Recognition of Events in Video, 2001. Proceedings. IEEE Workshop on*, pages 64–72, 2001. doi: 10.1109/EVENT.2001.938868.

[216] Yuping Shen and H. Foroosh. View-invariant action recognition using fundamental ratios. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–6, June 2008. doi: 10.1109/CVPR.2008.4587755.

[217] Ali Farhadi and Mostafa Kamali Tabrizi. Learning to recognize activities from the wrong view point. In *Proceedings of the 10th European Conference on Computer Vision: Part I*, ECCV '08, pages 154–166, Berlin, Heidelberg, 2008. Springer-Verlag. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_13. URL http://dx.doi.org/10.1007/978-3-540-88682-2_13.

[218] Jingen Liu, M. Shah, B. Kuipers, and S. Savarese. Cross-view action recognition via view knowledge transfer. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 3209–3216, Washington, DC, USA, 2011. IEEE Computer Society. ISBN 978-1-4577-0394-2. doi: 10.1109/CVPR.2011.5995729. URL http://dx.doi.org/10.1109/CVPR.2011.5995729.

[219] R. Cutler and L.S. Davis. Robust real-time periodic motion detection, analysis, and applications. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8): 781–796, Aug 2000. ISSN 0162-8828. doi: 10.1109/34.868681.

[220] Chiraz BenAbdelkader, Ross G. Cutler, and Larry S. Davis. Gait recognition using image self-similarity. *EURASIP J. Appl. Signal Process.*, 2004:572–585, January 2004. ISSN 1110-8657. doi: 10.1155/S1110865704309236. URL http://dx.doi.org/10.1155/S1110865704309236.

[221] Gunnar Johansson. Visual perception of biological motion and a model for its analysis. *Perception & Psychophysics*, 14(2):201–211, 1973. ISSN 0031-5117. doi: 10.3758/BF03212378. URL http://dx.doi.org/10.3758/BF03212378.

[222] S. V. Stevenage, M. S. Nixon, and K. Vince. Visual analysis of gait as a cue to identity. *Applied Cognitive Psychology*, 13(6), 1999.

[223] Chewyean Yam, Mark S. Nixon, and John N. Carter. Gait recognition by walking and running: a model-based approach. In *In Asian Conference on Computer Vision*, pages 1–6, 2002.

[224] R. Urtasun and P. Fua. 3d tracking for gait characterization and recognition. In *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, pages 17–22, May 2004. doi: 10.1109/AFGR.2004.1301503.

[225] J. Han and B. Bhanu. Individual recognition using gait energy image. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 28(2):316–322, Feb 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.38.

[226] T. Kobayashi and N. Otsu. Action and simultaneous multiple-person identification using cubic higher-order local auto-correlation. In *Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on*, volume 4, pages 741–744 Vol.4, Aug 2004. doi: 10.1109/ICPR.2004.1333879.

[227] Yasushi Makihara, Ryusuke Sagawa, Yasuhiro Mukaigawa, Tomio Echigo, and Yasushi Yagi. Gait recognition using a view transformation model in the frequency domain. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part III*, ECCV'06, pages 151–163, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33836-5, 978-3-540-33836-9. doi: 10.1007/11744078_12. URL http://dx.doi.org/10.1007/11744078_12.

[228] Chen Wang, Junping Zhang, Liang Wang, Jian Pu, and Xiaoru Yuan. Human identification using temporal information preserving gait template. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 34(11):2164–2176, Nov 2012. ISSN 0162-8828. doi: 10.1109/TPAMI.2011.260.

[229] Khalid Bashir, Tao Xiang, and Shaogang Gong. Bashir et al.: Gait representation using flow fields 1 gait representation using flow fields, 2009.

[230] Gait flow image: A silhouette-based gait representation for human identification. *Pattern Recognition*, 44(4):973 – 987, 2011.

[231] Y. Makihara, B.S. Rossa, and Y. Yagi. Gait recognition using images of oriented smooth pseudo motion. In *Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on*, pages 1309–1314, Oct 2012. doi: 10.1109/ICSMC.2012.6377914.

[232] Stephen Lombardi, Ko Nishino, Yasushi Makihara, and Yasushi Yagi. Two-point gait: Decoupling gait from body shape. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 1041–1048, Washington, DC, USA, 2013. IEEE Computer Society. ISBN 978-1-4799-2840-8. doi: 10.1109/ICCV.2013.133. URL http://dx.doi.org/10.1109/ICCV.2013.133.

[233] Navneet Dalal, Bill Triggs, and Cordelia Schmid. Human detection using oriented histograms of flow and appearance. In *Proceedings of the 9th European Conference on Computer Vision - Volume Part II*, ECCV'06, pages 428–441, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-33834-9, 978-3-540-33834-5. doi: 10.1007/11744047_33. URL http://dx.doi.org/10.1007/11744047_33.

[234] Horesh Ben Shitrit, Jérôme Berclaz, François Fleuret, and Pascal Fua. Multi-Commodity Network Flow for Tracking Multiple People. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2013. ISSN 0162-8828.

[235] Heng Wang, Alexander Kläser, Cordelia Schmid, and Cheng-Lin Liu. Dense trajectories and motion boundary descriptors for action recognition. Research Report RR-8050, INRIA, August 2012. URL http://hal.inria.fr/hal-00725627.