UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Region-based particle filter leveraged with a hierarchical co-clustering

by

David Varas González

UNIVERSITAT POLITÈCNICA
DE CATALUNYA

Universitat Politècnica de Catalunya
Teoria del Senyal i Comunicacions

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

Region-based Particle Filter leveraged with a Hierarchical Co-Clustering

by David Varas González

Advisor: Prof. Ferran Marques Acosta
Barcelona, August 2016

Abstract

In this thesis, we exploit the hierarchical information associated with images to tackle two fundamental problems of computer vision: video object segmentation and video segmentation.

In the first part of the thesis, we present a *video object segmentation* approach that extends the well knonw particle filter algorithm to a region-based image representation. Image partition is considered part of the particle filter measurement, which enriches the available information and leads to a reformulation of the particle filter theory. We define particles as unions of regions in the current image partition and their propagation is computed through a single optimization process. During this propoagation, the prediction step is performed using a co-clustering between the previous image object partition and a partition of the current one, which allows us to tackle the evolution of non-rigid structures.

The second part of the thesis is devoted to the exploration of a co-clustering technique for *video segmentation*. This technique, given a collection of images and their associated hierarchies, clusters nodes from these hierarchies to obtain a coherent multiresolution representation of the image collection. We formalize the co-clustering as a Quadratic Semi-Assignment Problem and solve it with a linear programming relaxation approach that makes effective use of information from hierarchies. Initially, we address the problem of generating an optimal, coherent partition per image and, afterwards, we extend this method to a multiresolution framework. Finally, we particularize this framework to an iterative multiresolution video segmentation algorithm in sequences with small variations.

Finally, in the last part of the thesis we validate the presented techniques for object and video segmentation using the proposed algorithms as tools to tackle problems in a context for which they were not initially thought.

Acknowledgements

First of all, I would like to express my heartfelt gratitude to my supervisor and friend Prof. Ferran Marques, for our the endless hours devoted to talk about hierarchies and football, for believing in me, for his guidance and his confidence when the path was hard, for his help in the good and the bad moments and for being there, even in the last minute of our deadlines, with a confortable smile. This thesis would have not been possible without you.

An important part of this thesis has been developed with my friends at our office, D5-120. We have shared our little knowledge, baking competitions, lunches, exclusive talks and an endless number of (freak) moments. I sincerely thank you all for creating such a great place to work.

I also want to acknowledge Albert and Josep, for their patience with my ignorance, for teaching me so many things that have improved the quality of my research and for the weekly tactical football discussions.

A very special thanks goes to my parents, my brother and the rest of my family, for being always there, for their love, for helping me crossing rivers and mountains and making me a better person. There is no doubt that a huge part of this thesis is also yours.

Finally, I would like to say Xana *muito obrigado* for all the especial moments that we share every day, for her patience, for being by my side every single minute, for making my life better. You are the best finding of my research.

I also acknowledge that this thesis would not have been possible without the financial assistance of the Image Processing Group (GPI) at UPC and the Catalan Government through the FI grant and the projects CENIT-2007-1012 I3MEDIA, TEC2010-18094 and TEC2013-43935-R.

Acronyms

PDF	Probability Density Function
SIR	Sequential Importance Resampling
CNN	Convolutional Neural Network
HEVC	High Efficiency Video Coding
SIS	Sequential Importance Sampling
ASIR	Auxiliary Sampling Importance Resampling
RPF	Regularized Particle Filter
MC	Monte Carlo
QSAP	Quadratic Semi Assignment Problem
LP	Linear Programming
BPT	Binary Partition Tree
HOG	Histogram of Oriented Gradients
LFCO	Linear Fractional Combinatorial Optimization
MRHC	Multiresolution Hierarchical Co-clustering
UCM	Ultrametric Contour Map
BPR	Boundary Precision Recall
VPR	Volume Precision Recall

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Segmentation is one of the fundamental problems in computer vision with a large number of applications such as action recognition [5], 3D reconstruction [6], or video indexing [7]. In a context in which a single image is considered, many image segmentation methods exist (e.g. [8, 9, 10]) that produce robust results. Most of these methods agree on the use multiple types of similarities based on brightness, color and texture over local image patches to achieve best image segmentation performance. Moreover, in the area of segmentation, hierarchical techniques have proven to produce the best frameworks. These approaches aim at creating a hierarchy of partitions by sequentally fusing regions composed by one or more image pixels.

The goal of **Video Object Segmentation** is to delineate the boundaries of moving and/or static objects that appear in arbitrary videos. In general, objects are spatially connected, and characterized by locally smooth motion trajectories. In other words, their shape, which is usually assumed to be a connected component, does not define abrupt trajectory changes. However, in some cases, this shape may suffer strong deformations. Pixels that belong to objects to be segmented occupy regions within each video frame. Also, assuming relatively slow camera motion, the shape and location of these regions vary slowly from frame to frame. Thus, the video object segmentation task can be formulated as tracking regions across frames of a sequence, such that the resulting tracks are locally smooth. This results in a division of the spatiotemporal video volume into tubes that are coherent in space and time and represent the shape and trajectory of objects along the sequence.

Video Object Segmentation is a prerequisite step of a wide range of higher-level vision algorithms, including activity recognition [11], video summarization and retrieval [12], and video rendering [13]. Most prior work focuses on a simplified formulation of this problem: that of segmenting moving objects [14]. Typically, these methods require the number of moving objects or layers to be prespecified, and cannot handle long videos. Also, motion segmentation using optical flow rests on the assumption of brightness constancy, which is violated at boundaries that move, resulting in poor estimates of object contours [15].

Currently, the two predominant approaches to tackle the video object segmentation problem are tracking interest points, and perceptual grouping of pixels from all frames of the sequence [16]. However, there is not a consensus about which is the best way to face the problem.

Point-based approaches group the trajectories of keypoints with similar motion [12, 17]. However, point tracking approaches yield to a confidence map of the neighbohood of the objects instead of a segmentation. To improve robustness, multiple points that fall within a fixed size and shape window are jointly tracked along a pre-specified number of consecutive frames. These ad hoc choices increase complexity that is proportional to the product of scales and locations of the scanning windows. As interest points do not capture the spatial cohesiveness of objects, this approach usually suffers from multiple overlapping object detections. These are usually resolved by making heuristic assumptions about the number, sizes, and shapes of objects present in the video.

In the second approach, video object segmentation is formulated as an optimization employing motion and appearance constraints. These constraints are then used to propagate the segments to all frames. These methods require accurate object region annotation for the first frame and employ region tracking to segment the rest of frames into object and background regions. Both fully automatic methods and methods requiring manual initialization have been proposed for video object segmentation. In the latter class [18] need annotations of object segments in key frames for initialization.

Video segmentation is far less researched due to its computational complexity and the inherent difficulties of the problem such as camera-motion, occlusions, changes in scale, perspective, illumination and contrast, or non-rigid deformations. Image segmentation aims to group perceptually similar pixels into regions. Video segmentation generalizes this concept to the grouping of pixels into spatio-temporal regions that exhibit coherence in both appearance and motion. Such segmentation is useful for several higher-level vision tasks such as object tracking and segmentation, content-based retrieval, and visual enhancement.

In spite of its potential applications, relatively few works address the problem of video segmentation. The reviewed works show that two main approaches are used to solve the problem of video segmentation. On the one hand, a common approach is to extend single image segmentation techniques to multiple frames, exploiting the fact that there is redundancy along the time axis and that the motion field is smooth. Thus, for instance, Levinshtein et al. [1] extend superpixel grouping [2] to 3D voxels. Sundaram and Keutzer [3] apply spectral clustering to all the video sequence pixels with an affinity matrix given by the gPb 2D contour detection algorithm [4] which combines intensity, color and texture. On the other hand, several works tackle the problem as one of labeling using minimum energy optimization of a Markov Random Field where nodes are now voxels [5–8] or 2D regions [9], again an extension of a successful segmentation strategy in single images. Grundmann et al. [10] build their hierarchical algorithm for long sequences upon Felzenszwalb and Huttenlocher's [11] graph algorithm for 2D image segmentation. Likewise, Huang et al. adapt the graph-cut algorithm to run on 3D hypergraphs whose nodes are regions resulting from an oversegmentation of each frame.

In this thesis we combine concepts associated with both approaches in order to efficiently solve the problem of video segmentation. On the one hand, we use single image oversegmentations and their associated hierarchies to exploit the temporal redundancy of relevant contours in the sequence. On the other hand, the problem is tackled as an iterative optimization over the contours tacking into account information at different resolutions.

Intuitively, besides within-frame similarities used for image segmentation, video segmentation should also use between-frame similarities to connect and thus segment corresponding regions across multiple frames. While recent work on this field proposes a variety of such between-frame similarities [2, 10–13] there is no common agreement yet on which similarities are necessary for best performance. We also explore two ways in which similarities are computed depending on which region parts are taken into account to compute these similarities: region boundaries or their inner pixels.

We face both the video object segmentation and the video segmentation problems. At first sight, video object segmentation may be seen as a particularization of a video segmentation scenario, in which a single object should be delimited in the sequence. In this sense, these two fundamental problems of computer vision may be associated with a two-class (video object segmentation) and an N-class (video segmentation) labeling problems. However, the way to tackle these challenges is very different.

The problem of video object segmentation was first explored in this thesis. This analysis was naturally requested by the introduction of the information provided by a partition in a tracking scenario, which was one of the primary objectives of this work. In the context of tracking, we introduced regions in a particle filter tracker to explore its performance when not all the pixels of a geometrical shape were used to represent and update the object model. Due to the potential showed by this approach, we decided to merge these two concepts to develop a video object segmentation algorithm developing a new theory for the particle filter based on regions. This region-based approach for video object segmentation is presented in Chapter 2 and the main contributions of this thesis to this area are a novel formulation of the particle filter algorithm, a joint optimization to propagate particles and a refinement step based on the Bayes' rule to improve the quality of the final segmentation.

One of the key aspects of the process of estimating the shape of an object iteratively in a sequence is the propagation of this shape between consecutive video frames. In our region-based video object segmentation algorithm, we developed a co-clustering technique to propagate all the information associated with the object segmentation using a single optimization process.

While we were developing this co-clustering algorithm, we discovered that clustering regions from partitions was a complex problem with a large number of possible applications. Thus, we decided to explore a video segmentation scenario introducing a novel co-clustering technique that groups nodes of hierarchies associated with a set of images represented at different resolutions. This is the second main topic of this thesis, and it is presented in Chapter 3. The main contributions of this thesis to video segmentation are an optimization on hierarchies of video frames and an iterative approach that combines the information at different resolutions to perform video segmentation.

Finally, once our co-clustering technique proved to obtain robust segmentation results in the context of video segmentation, we introduced this knowledge in the propagation step of our previous region-based video object segmentation algorithm to improve the results. These results are analyzed in detail in Chapter 2.

Chapters 2 and 3, that correspond to video object segmentation and video segmentation respectively, are organized as follows: first, an introduction is presented to motivate the problem. Second, the most important works in the area of study are reviewed in a related work section. Third, the technique developed in this thesis is presented. Then, this technique is assessed with a set of experiments and finally some conclusions are drawn.

In Chapter 4, we present three different applications in which the techniques presented in previous chapters are used to solve computer vision problems in contexts for which they were not initially developed. Finally, in Chapter 5 we discuss some conclusions derived from the work presented in this Thesis and its applications.

Chapter 2

Region-based particle filters

2.1 Introduction

Many problems in science require estimation of the state of a system that changes over time. Usually, this estimation must be performed using a sequence of noisy measurements because the real state of the system cannot be directly accessed.

In the Bayesian approach to dynamic state estimation, the goal is to construct the posterior *probability density function* (pdf) of the state based on a set of observed noisy measurements. These measurements include a combination of the state that must be inferred with some noise that can be generated by different sources. In the inference process, information is usually represented by two vectors. The state vector contains all relevant information required to describe the system under investigation. For example, in tracking problems, this information could be related to the kinematic characteristics of the target. The measurement vector represents observations that are related with the state vector. In other words, the measurement vector contains the information of the problem that can be accessed in order to estimate the state vector.

For many problems, an estimate is required every time that a measurement is observed. In this case, a recursive filter is a convenient solution. In a recursive filtering approach observed data (measurement vector) can be processed sequentially and it is not necessary to store the complete data set in order to analize a new measurement when it becomes available. Otherwise, the problem rapidly becomes intractable. Such a filter consists of essentially two stages: prediction and update.

It can be proved that the Kalman Filter [19] is the optimal solution for this problem assuming that the noise and the initial state have Gaussian distributions and the functions that relate the present state and the observation to their previous values are linear functions. If these functions are nonlinear, they can be linearized using the Taylor series expansion to obtain the Extended Kalman Filter [20].

Similar to the Kalman filter, the Extended Kalman Filter assumes that the initial state is distributed by a Gaussian function. However, using this method, both the function that defines the present state and the observation can be non-linear functions.

One limitation of these filters is the assumption that the state variables are normally distributed. Thus, they will lead to poor estimations of those state variables that do not follow a Gaussian distribution. This limitation can be overcomed by using numerical methods such as particle filtering [21]. In the context of Bayesian tracking, Particle Filters provide a robust tracking framework as they are neither limited to linear systems nor require the noise to be Gaussian [22].

Several variants of the particle filter such as SIR, ASIR, and RPF are introduced within a generic framework of the *Sequential Importance Sampling* (SIS) algorithm in [23].

Object tracking is one of the fundamental issues that computer vision applications must deal with. Detection, tracking and analysis of object's behavior are common objectives in tasks such as automated surveillance, video indexing, human-computer interaction or motion-based recognition.

In image object tracking, difficulties such as fast object motion, changes of the object patterns or of the scene, nonrigid object structures, occlusions and camera motion are common problems that must be handled by the system [24]. In this context, an object trajectory is generated by estimating the object location in each video frame. Although these locations may be enought to track objects in video sequences, they do not provide an accurate estimation of their shapes.

Such an estimation has a crucial role in video editing, postprocessing and interactive applications in which the shape of the object should be considered. As an approximation, objects are usually represented by a geometrical shape (e.g.: an ellipse). However, a fixed shape may be a too simple representation of real objects and applications using object's shape to extract information about the scene cannot make use of these trackers

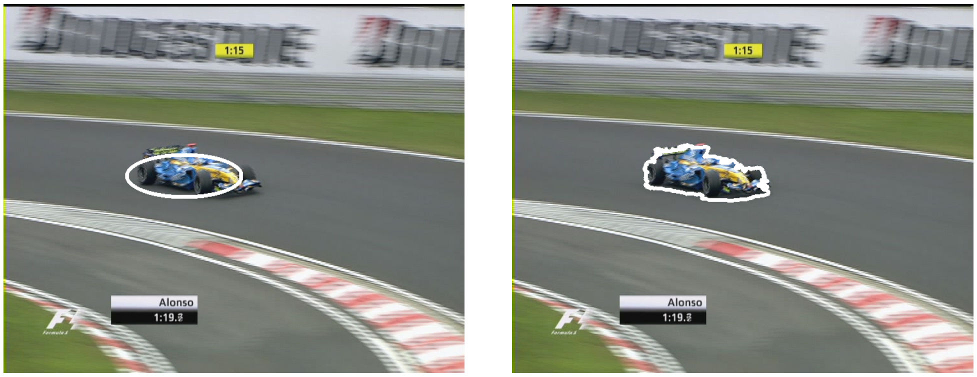


Figure 2.1: Extension of the classical particle filter using region information. Left: tracking of a car with a classical color-based particle filter that estimates the object with an elliptical shape. Right: result obtained with the region-based algorithm presented in this thesis.

and require video object segmentation (i.e.: gesture recognition). In Figure 2.1, the difference between a shape-based and our region-based approach in the context of F1 car tracking is presented. Moreover, since a fixed shape does not allow segmenting the object from the scene, an updated model of the target may be corrupted by those pixels that do not belong to the object and are included inside the estimated shape. Techniques such as [25] solve that problem by including an object segment on the loop. One key distinction between tracking and segmentation is that tracking systems are usually designed for real time purposes, while segmentation systems may work off-line as the importance of its applications relies on obtaining accurate segmentations ([26], [27]).

In this context, the main contributions of this thesis to the area of **video object tracking and segmentation**, which are described in this chapter, are:

A **novel formulation** of the particle filter algorithm using a region-based image representation. When regions from a given partition are used for object tracking, this partition should also be considered in the derivation of new mathematical expressions allowing object segmentation at each time instant.

A **joint optimization** method to propagate the complete set of particles with a single process. Co-clustering is used to perform this task. Using

this approach, the algorithm can robustly tackle the evolution of non-rigid structures.
A **refinement step** to both improve the particles quality and guide the randomness associated with the particle filter towards better object estimates based on the Bayes rule.

This chapter is organized as follows. In Section 2.2, the most relevant state-of-the-art techniques in video object segmentation are described. For each technique, the key differences with our work are briefly highlighted and discussed. In Section 2.3 we review the most commonly used particle filtering methods and the basic structure of video object tracking using particle filters. A first approach towards a particle filter implementation using regions to perform both tracking and segmentation is presented in Section 2.4. This method uses a particle filter algorithm combined with partitions information to extract the object shape at each time instant. Then, in Section 2.5 we explain, based on the previous structure, the extension of the particle filter theory to a region-based approach. The use of regions allows us to robustly tackle the evolution of non-rigid structures over time. The performance of the algorithm is tested using the Segtrack dataset [26], the Segtrackv2 dataset [28] and a set of sequences from the LabelMe Video database [29]. A set of experiments to assess these algorithms are performed in Section 2.6 and the numerical and graphical results of these tests are presented. Finally, some conclusions are drawn in Section 3.9.

2.2 State-of-the-art

The tracking process in video sequences involves an object or background representation such as color histograms or mixture models. Different approaches have been presented in the context of video object tracking for this purpose. In [30], the object model is generated by computing the mean color of all the pixels included inside rectangular windows. Although this system is capable of tracking multiple objects, it needs support of existing methods for making a periodic update of the object model.

A weighted histogram computed from a geometrical region is used in [31] to represent the object. In this work the authors use the mean-shift algorithm to locate the object in the scene. The shape of this region is not related with the object shape and thus, it cannot be segmented. Instead of modeling the object, [32] model each background pixel as a mixture of

Gaussians. This leads to a system that can effectively find foreground pixels associated with the tracked object even with abrupt light changes. Grouping similar pixels under a certain criterion provides high level information that can be effectively exploited to segment and characterize objects. Another approach in which color distributions are also used is the mean shift tracker [31]. The mean shift algorithm proved to be a very powerful tool for the analysis of complex multimodal feature spaces [31] and has been succesfully used to track objects in a wide variety of scenarios. In [33], a hybrid approach using particle filters and mean shift trackers is used to reduce the complexity of the algorithm while being able to deal with multimodal pdfs. In [34] an extension of the mean shift algorithm is presented for face tracking and segmentation reliying on the use of an image partition and explicit color models for object and background, which are updated through the tracking process. The information provided by regions of this partition is used to define with precision face contours, providing a mechanism to adapt the tracker to variations in object scale and to illumination and background changes.

The extension from the pixel model to a region-based model has already been considered for object tracking. A region-based tracker based on the mean shift algorithm is presented in [35]. In it, face tracking is performed using an elliptical kernel. Then, a segmentation is provided after a fitting of the ellipse in the image partition. Although results are promising, the system is not robust to occlusions during the tracking of the object. A set of patches are considered as regions in [36] to define the object which is tracked with a particle filter that uses the normalized cross-correlation to weight its particles. Targets are tracked in challenging situations, but their shape is not estimated.

Object tracking and segmentation is addressed in [37] using pixel-wise posteriors. In it, although good results are obtained over a large database, errors appear due to the lack of spatial information. We overcome this problem by considering the spatial information provided by the relations among regions (Section 2.5). Motion estimation is used to obtain video object segmentation in [38]. Besides, an appearance model with spatial constraints is considered. Despite their promising results, some parts of the objects are lost due to the importance assigned to each fragment during the tracking. In [39], motion, appearance and predicted-shape similarities are used to perform object extraction in video sequences. However, this work assumes that objects are spatially cohesive and characterized by locally smooth motion trajectories. Our approach substitutes the motion estimation step by a co-clustering (Section 2.5) to predict the position and the shape of the object. In [40], a system to segment foreground objects in video is presented using both static and dynamic cues. This strategy produces satisfactory estimations by discovering object-like key-segments, but it is not robust when the foreground and background are similar. We use both shape descriptors and a contour-based representation of the object to solve these errors (Section 2.6).

Object tracking is modeled as a Maximum Weight Cliques problem in [41] to perform object segmentation in all video frames simultaneously. In this approach, the shape of the object is not predicted in adjacent frames when region similarity is computed. Thus, the segmentation performance is degraded for fast moving objects. Our approach overcomes this problem combining a co-clustering with a tracking oriented adjacency graph.

In [27], objects are tracked by identifying stationary statistics of both appearance and shape over time. In it, occlusions and disocclusions are taken into account, obtaining accurate segmentations of the object in challenging sequences. However, further work is required to deal with occlusions caused by other objects and to improve the detection of self-disocclusions. A similar approach is used in [42] to identify static and moving objects in the scene.

In contrast with other tracking methods, particle filters can robustly deal with occlusions and track objects in clutter as they neither are limited to linear systems nor require the noise involved in the process to be Gaussian. In [43], a particle filter with edge-based features is proposed. This method has been widely used since it provides a robust framework for tracking curves in clutter. However, the space of possible deformations is limited and some transformations of the object shape may not be correctly estimated. We adapt this idea considering shape descriptors without any restriction in the space of possible deformations.

Image-based features for particle filters were introduced by [44]. In it, color histogram is used to robustly track objects in the scene. This feature has the advantages of being scale invariant and robust to partial occlusions and rotations. Moreover, it can be efficiently computed. In our work, we use the Diffusion distance [45] instead of the Bhattacharyya distance [46] for histogram comparison since it leads to better perceptual performance (Section 2.5). As the color of an object can vary through time, the target model is adapted during temporally stable image observations in [47]. Note that [44], [47] do not provide shape estimation.

We propose a region-based particle filter that allows tracking and segmenting objects in video sequences. The extension from the pixel model to a region-based model has already been considered for object tracking. For instance, a region-based tracker relying on the mean shift algorithm [31] is presented in [35]. In it, objects correctly modeled as given shapes are robustly tracked and segmented (e.g., faces modeled as ellipses). In our work, we overcome this situation as we do not consider any geometrical shape to represent the object (Section 2.5). In [36], a set of patches is considered as regions to define the object, which is tracked with a particle filter. Targets are tracked in challenging situations, but their shape is not estimated.

2.3 Particle Filters for object tracking

The tracking problem

Let us consider the problem of estimating the state of a system x_k, that defines the evolution of a target at time k given a set of measurements $z_{1:k} = \{z_j, j = 1, ..., k\}$ up to the same time instant:

$$x_k = f_k(x_{k-1}, v_{k-1}) \tag{2.1}$$

$$z_k = h_k(x_k, n_k) \tag{2.2}$$

where $f_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_v} \to \mathbb{R}^{n_x}$ and $h_k : \mathbb{R}^{n_x} \times \mathbb{R}^{n_n} \to \mathbb{R}^{n_z}$ are a priori unknown and possibly nonlinear functions that define the state and measurement evolution along time, $\{v_{k-1}, k \in \mathbb{N}\}$ and $\{n_k, k \in \mathbb{N}\}$ are independent and identically distributed $(i.i.d.)$ noise sequences, and n_x, n_z, n_v, n_n are the dimensions of the state, measurement, state vector and measurement noise vector, respectively.

As the set of measurements $z_{1:k}$ is available at time k to estimate the state x_k with a certain probability, Bayesian analysis proposes the study of the pdf $p(x_k|z_{1:k})$. This pdf can be recursively estimated in two stages: Prediction and Update. In Prediction, all the previous information is used to predict the state of the object at the current instant. Then, this prediction is corrected in the update step when the current measurement is available.

Let us suppose that the objective function $p(x_{k-1}|z_{1:k-1})$ at $k-1$ is available. The prior of the state at time k can be obtained using the Chapman-

Kolmogorov equation [48] in the prediction stage.

$$p(x_k|z_{1:k-1}) = \int p(x_k|x_{k-1})p(x_{k-1}|z_{1:k-1})dx_{k-1} \tag{2.3}$$

Note that in Equation 2.3, the equality $p(x_k|x_{k-1}, z_{1:k-1}) = p(x_k|x_{k-1})$ has been applied as Equation 2.1 defines a Markov Process of order one. The probabilistic model $p(x_k|x_{k-1})$ of the state evolution is defined by Equation 2.3 and the known statistics of v_{k-1}.

Then, when a measurement z_k becomes available at time k, Bayes rule may be used to update the prior in the update stage:

$$p(x_k|z_{1:k}) = \frac{p(z_k|x_k)p(x_k|z_{1:k-1})}{p(z_k|z_{1:k-1})} \tag{2.4}$$

with the normalizing constant:

$$p(z_k|z_{1:k-1}) = \int p(z_k|x_k)p(x_k|z_{1:k-1})dx_k \tag{2.5}$$

The relations between Equation 2.3 and Equation 2.4 are the basis of the optimal Bayesian solution. However, this recursive propagation of the posterior density is only a conceptual solution that in general cannot be determined analytically. Only in a restrictive set of cases a solution for this recursion can be evaluated, including the Kalman filter and grid-based filters. When the analytic solution is intractable, extended Kalman filters, approximate grid-based filters, and particle filters approximate the optimal Bayesian solution.

Particle filters

The Sequential Importance Sampling (SIS) algorithm is a recursive Monte Carlo (MC) method that forms the basis of the generic particle filter algorithm as for most sequential MC filters [49]. This approach is known as bootstrap filtering [21], the condensation algorithm [50], particle filtering [51], interacting particle approximations [52], and survival of the fittest [53]. Recursive Bayesian filtering may also be implemented using this technique. The key idea of this algorithm is to represent the objective posterior density function $p(x_k|z_{1:k})$ defined in the tracking problem (Section 2.3) by a set of random samples with their associated weights. These samples and weights may be further used to compute estimates of the objective function. As Monte Carlo methods are unbiased estimators, the expectation of the SIS filter is equal to the expectation of the real pdf. Moreover, as the number of samples becomes very large, this characterization becomes an equivalent representation of the posterior pdf, and the SIS filter approaches the optimal Bayesian estimate regardless of the dimension of the state vector. These properties make MC methods a good choice to be the basis of the SIS filter for tackling the tracking problem.

A common way to solve the tracking problem without imposing any constraint is the SIS particle filter. Let us consider a set of support points $\{x_{1:k}^{(i)}, i = 1, ..., N_s\} \in \Omega \leq \mathbb{R}^{D_x}$ with associated weights $\{w_k^{(i)}, i = 1, ..., N_s\}$, where D_x is the dimension of each support point, and Ω is denoted as the *solution space*. Let us define a set of *particles* $\{x_{1:k}^{(i)}, w_k^{(i)}\}_{i=1}^{N_s}$ that characterize the posterior $p(x_{1:k}|z_{1:k})$, where $x_{1:k} = \{x_j, j = 1, ..., k\}$ is the set of all states up to time k. Then, the posterior $p(x_{1:k}|z_{1:k})$ can be approximated as:

$$p(x_{1:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_{1:k} - x_{1:k}^{(i)}) \tag{2.6}$$

where the weights $w_k^{(i)}$ are chosen using *importance sampling* [49]. As this posterior is computed using a sequential procedure, weights can be expressed as [23]:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)} \tag{2.7}$$

where $q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)$ is called *importance density.* Note that particles are drawn from this function that is a priori not defined. Therefore, a convenient function may be selected depending on the problem.

The SIS algorithm performs the recursive propagation of the weights and support points as each measurement is received sequentially over time. A common problem with the SIS particle filter is that, after a few iterations, all but one particle will have negligible weight and one particle will concentrate all the probability mass. Note that this is in general an undesired situation, as a single particle will not correctly represent the function $p(x_{1:k}|z_{1:k})$ in almost any case.

In [49], it is shown that the variance of the importance weights can only increase over time. For this reason, it is impossible to avoid the degeneracy phenomenon when the importance weights are recursively propagated. This degeneracy implies that a large computational effort is devoted to updating particles whose contribution to the approximation of the objective function is almost zero. However, this degeneracy can be measured in order to mitigate its effects. A suitable measure to quantify the degeneracy of the algorithm is the Effective Sample Size (N_{eff}) introduced in [54] and [55]. This measure quantifies the number of particles that are contributing in an effective manner to the estimation of $p(x_{1:k}|z_{1:k})$ and it is defined as:

$$N_{eff} = \frac{N_s}{1 + var(w_k^{*(i)})} \tag{2.8}$$

where $w_k^{*(i)} = p(x_k^i|z_{1:k})/q(x_k^i|x_{k-1}^i, z_k)$ is the "true weight" of a particle. As this weight cannot be evaluated exactly because the pdf $p(x_k^i|z_{1:k})$ is the objective function to be estimated, an estimate of the Effective Sample Size can be obtained by:

$$\hat{N}_{eff} = (\sum_{i=1}^{N_s} w_k^{(i)})^{-1} \tag{2.9}$$

where $w_k^{(i)}$ is the normalized weight obtained using 2.7. Note that $N_{eff} \leq N_s$ and a small value of this measure indicates severe degeneracy as only a few particles are effectively contributing to the estimation of the function. Clearly, the degeneracy problem is an undesirable effect in particle filters. The brute force approach to reducing its effect is to use a very large number of particles. This is often impractical as the computational time of the algorithm increases exponentially with N_s. However, other methods can be used to reduce the effects of degeneracy. Two methods have been mainly proposed for this purpose: good choice of importance density and use of a resampling step. These methods are briefly described in this section.

Good Choice of Importance Density: The first method consists on choosing the importance density that minimizes $var(w_k^{*(i)})$ so that N_{eff} is maximized. The optimal importance density function that minimizes the variance of the true weights $w_k^{*(i)}$ conditioned on x_{k-1}^i and z_k has been shown [49] to be:

$$q(x_k|x_{k-1}^i, z_k)_{opt} = \frac{p(z_k|x_k, x_{k-1}^i)p(x_k|x_{k-1}^i)}{p(z_k|x_{k-1}^i)} \tag{2.10}$$

This choice of importance density is optimal since for a given state at the previous time instant (x_{k-1}^i), the importance weight w_k^i associated to its particle in the current approximation takes the same value, whatever sample is drawn from $q(x_k|x^i_{k-1}, z_k)_{opt}$. Hence, conditional on x^i_{k-1}, $var(w^{*(i)}_k) = 0$. This is the variance of the different w^i_k resulting from different sampled x^i_{k-1}.

This optimal importance density suffers from two major drawbacks. It requires the ability to sample from $p(x_k|x^i_{k-1}, z_k)$ and to evaluate an integral over the new state. In the general case, it may not be straightforward to do either of these things. There are two cases when using the optimal importance density is possible.

The first case is when is a member of a finite set. In such cases, the function presented in Equation 2.10 can be evaluated, and sampling from $p(x_k|x^i_{k-1}, z_k)$ is possible. An example of an application when is a member of a finite set is a Jump–Markov linear system for tracking maneuvering targets [56], whereby the discrete modal state (defining the maneuver index) is tracked using a particle filter, and (conditioned on the maneuver index) the continuous base state is tracked using a Kalman filter.

Analytic evaluation is possible for a second class of models for which $p(x_k|x_{k-1}, z_k)$ is Gaussian [57]. This can occur if the dynamics are nonlinear but the relation between state and measurement is linear.

For many other models, analytic evaluation is not possible. However, it is possible to construct suboptimal approximations to the optimal importance density $p(x_k|x^i_{k-1}, z_k)$ by using local linearization techniques [49]. Such linearizations use an importance density that is a Gaussian approximation to $p(x_k|x^i_{k-1}, z_k)$. Another approach is to estimate a Gaussian approximation to using the unscented transform [58].

Resampling: The second method by which the effects of degeneracy can be reduced is to use resampling whenever a significant degeneracy is observed (i.e., when N_{eff} falls below a certain threshold). The basic idea of resampling is to eliminate particles that have small weights and to concentrate on particles with large weights. However, a certain randomness is considered when selecting particles to be eliminated as diversity is a desired property of the algorithm. This is, not all particles with small weight should be systematically eliminated as they may be close to good solutions of the problem in subsequent time instants.

The resampling step involves generating a new set $\{x^{*i}_k\}^{N_s}_{i=1}$by resampling (with replacement) N_s times from an approximate discrete representation of $p(x_k|z_{1:k})$ given by Equation 2.6 so that $P(x_k^{*i} = x_k^j) = w_k^j$. The resulting sample is in fact an i.i.d. sample from the discrete density $p(x_k|z_{1:k})$. Therefore, the weights are now reset to $w_k^i = 1/N_s$. It is possible to implement this resampling procedure in $O(N_s)$ operations by sampling N_s ordered uniforms using an algorithm based on order statistics [51]. Note that other efficient (in terms of reduced MC variation) resampling schemes, such as stratified sampling and residual sampling [55], may be applied as alternatives to this algorithm. Systematic resampling [59] is the most common choice since it is simple to implement, takes time $O(N_s)$, and minimizes the MC variation.

Although the resampling step reduces the effects of the degeneracy problem, it introduces other practical problems. First, it limits the opportunity to parallelize since all the particles must be combined. Second, the particles that have high weights are statistically selected many times. This leads to a loss of diversity among the particles as the resultant sample will contain many repeated points. This problem, which is known as *sample impoverishment*, is severe in the case of small process noise. In fact, for the case of very small process noise, all particles will collapse to a single point within a few iterations. Third, since the diversity of the paths of the particles is reduced, any smoothed estimates based on the particles' paths degenerate. Schemes exist to counteract this effect. One approach considers the states for the particles to be predetermined by the forward filter and then obtains the smoothed estimates by recalculating the particles' weights via a recursion from the final to the first time step [60]. Another approach is MCMC [61].

Color-based particle filters

The color-based particle is presented in this section for further comparisons with our region-based approach. This technique estimates the state of an object in a video sequence using color cues. At each time instant, the algorithm receives an image (measurement) and the object is tracked using a parametrization of a geometrical shape and motion cues (state):

$$x_k = \{u, v, H_x, H_y, \dot{u}, \dot{v}\} \tag{2.11}$$

$$z_k = I_k \tag{2.12}$$

where (u, v) is the object position, H_x, H_y are the axis lengths of a geometrical shape (rectangle or ellipse) and $(\dot{u}, \dot{v})$ represent the object motion.

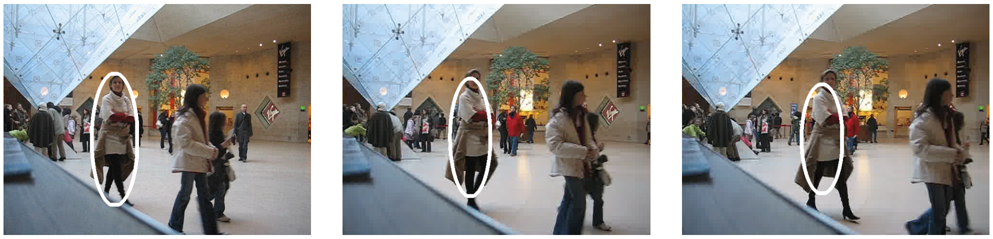


Figure 2.2: Example of a color-based particle filter for people tracking.

In order to describe the evolution of the object state in a compact manner, it is parametrized as a geometrical shape. Using this approach, transitions between consecutive states can be easily computed varying the parameters of this shape. Although this is a simple yet effective algorithm for tracking, the object shape is not accurately obtained during the process. An example of tracking with a color-based particle filter using a geometrical shape can be observed in Figure 2.2. In this figure, the object is represented with an ellipse.

The filter which is most commonly used for video object tracking in the literature ([43], [47], [62]) is the *Sampling Importance Resampling* (SIR) filter proposed by [21], which is a Monte Carlo method applied to recursive Bayesian filtering problems. This algorithm is derived from the SIS filter presented in Section 2.3 by an appropriate choice of importance density and the application of a resampling step at each time instant.

In this filter, the choice of importance density $q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)})$ is intuitive and simple to implement. This choice states that each particle at time k is drawn from a function that only depends on the particle at $k-1$. Substitution of this equality in Equation 2.7 yields to:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|x_k^{(i)}) \tag{2.13}$$

Moreover, a resampling step is applied at each time instant. This process generates a new set $\{x_{k-1}^{i*}\}_{i=1}^{N_s}$ by resampling with replacement N_s times from the approximate discrete representation of $p(x_{k-1}|z_{1:k-1})$. The result is an i.i.d. sample of the function presented in Equation 2.6. Thus, the weights are reset to $w_{k-1}^{i} = 1/N_s$ and the expression to compute the new weights at k becomes:

$$w_k^{(i)} \propto p(z_k|x_k^{(i)}) \tag{2.14}$$

These two considerations form the basis of the SIR color-based particle filter algorithm. This method tracks objects comparing the histogram of the pixels that lay inside a geometrical shape (typically, rectangle or ellipse) representing the object state and the histogram of the object model.

Resample Given a set of N_s particles S_{k-1} $=\{x_{k-1}^{(1)}, ..., x_{k-1}^{(N_s)}\}$, another set with the same number of particles S'_{k-1} $=\{x'^{(1)}_{k-1}, ..., x'^{(N_s)}_{k-1}\}$ is created using the SIR algorithm [21]. The new set of particles S'_{k-1} is created by randomly sampling (with replacement) the set S_{k-1}. Thus, some particles with high weights may be chosen several times, while others may not be chosen as the number of elements of the set does not change.

Propagation Particles of the new set S'_{k-1} are propagated using a function that describes the evolution of the object between consecutive time instants as showed in Equation 2.1. Usually, this evolution is modeled by a linear stochastic differential equation:

$$x_k^{(i)} = Ax'^{(i)}_{k-1} + Bv_{k-1}^{(i)} \tag{2.15}$$

Those particle parameters contained in $x_{t-1}^{(i)}$ that are supposed to change between consecutive images are first estimated using a deterministic matrix (A) in a *prediction* step. Then, they are slightly modified using a random variable $v_{t-1}^{(i)}$ with variance B to describe the trajectory and the evolution of the object scale in k. This random component is added in the *perturbation* step to the samples of the set creating N_s hypothetical states of the system. Randomness is a key point of particle filters, in which particles are able to jointly manage a set of N_s possible new states of the tracked object.

Evaluation: The evaluation process assigns to each particle i a weight $w^{(i)}$ associated with the probability of correct representation of the object. To weight the sample set, a similarity measure is required. Color-based particle filters, commonly relate this weighting with color distributions.

To weight the sample set, the similarity measure is computed between the target distribution (object model) and the distribution of the samples (object estimations). The distribution of each particle is formed by those pixels included in the geometrical shape defined by its propagated parameters. Particles with a color distribution closer to the target distribution will be assigned high weights, meaning that they represent the object better than those with lower weights.

Estimation

Once the weights of the samples are calculated, the mean state of the system at each iteration can be computed as:

$$E[S_k] = \sum_{i=1}^{N_s} w_k^{(i)} x_k^{(i)} \tag{2.16}$$

Since all the samples represent the same geometrical shape, mean state is a new particle with the same shape and whose parameters are defined by the weighted mean of the parameters of the N_s particles. This is one of the key points that motivates the use of a geometrical shape to represent the object.

2.4 Towards a Region-based particle filter

In this section, the first step towards a region-based particle filter is presented. This approach combines tracking and segmentation algorithms to perform both tasks at each iteration. On the one hand, the estructure of a particle filter is used for tracking the object along a sequence. On the other, image partitions are used to define the object shape from the geometrical shape that represents the state of the particle filter.

Here, we describe a particle filter that takes into account regions information for object tracking and segmentation. This method relies on the information provided by an image partition P_t to robustly track objects and estimate their shape as a union of regions from this partition.

Resample

The resamplig process is independent of the parameters tracked by the particle filter as it only considers the probability of a concrete state represented by the particle. This value has been computed during the *Evaluation* step of the previous iteration. Thus, the SIR algorithm can be applied without any changes at this point.

Propagation

In previous works, the parameters used for object tracking are propagated according to a dynamic model (see Section 2.3). Typically, these parameters are position, bounding box and additional information about the movement of the object. However, in this section, the aim of our algorithm is both to track and segment the target. Thus, propagation is performed in two levels:

Tracking: Four parameters are used to describe the object position along the sequence. Thus, the state of the object is represented by a four-dimensional vector:

$$x = \{u, v, w, h\} \tag{2.17}$$

where (u, v) represent the position of the top-left corner of the bounding box and (w, h) represent its width and height respectively.

As there is no prior knowledge about the evolution of each parameter, the function that defines the transition of their value between consecutive time instants is considered a Gaussian function centered at the previous value of the parameter.

Segmentation: In order to robustly define the shape of the object at each iteration, a binary mask is associated with each particle. This mask represents the shape of the tracked object at a given time instant and the four tracking parameters define its position and the size of its bounding box. This is an extension of the fixed geometrical shape towards new complex shapes to accurately segment the object. Let us consider a particle with a given binary mask representing an object at $k-1$. At time k the position and the size of its bounding box are propagated and the object is scaled with its new size and placed at its new position, generating a new binary mask. To propagate the object shape from $k-1$ to k, this mask is fitted into the image partition at time k. Several fitting approaches can be adopted [63]. In this work, the representation of the new object is defined by all regions of the partition with more than 40% of their pixels included in the logical positive area of the mask (*object mask*).

Any segmentation technique can be used to generate the image partition if it is dense enough. In this work, we create the partitions using [63]. This method ensures a computational time lower than a second for each image partition. Note that this time can be reduced segmenting only a neighborhood of the tracked object.

Evaluation

In order to weight the particles of the set, a similarity measure between color distributions is needed. In this work, we consider the Bhattacharya coefficient as measure between the distribution associated with the propagated mask of each particle and the model distribution.

Let $h(x_k^{(i)})$ be the histogram of those pixels that belong to the input image and are in the *object mask* associated with the i_{th} particle at time k. Consider q_{k-1} the updated model histogram at $k-1$ (see Section 2.5). Then, the Bhattacharya coefficient of the particle n is computed as:

$$\rho_t^{(n)}[h(x_k^{(i)}), q_{k-1}] = \sum_{b=1}^{N_b} \sqrt{h^{(b)}(x_k^{(i)}) q_{k-1}^{(b)}} \tag{2.18}$$

where N_b is the number of bins of each channel. In this work N_b has been set to 20.

Samples whose color distributions are similar to the target model should be favored. Thus, the weight of each sample is computed assuming that the Battacharya coefficient value has a Gaussian distribution centered at the maximum value of this coefficient ($\rho = 1$) with variance σ. In this work σ is set to 0.1. Note that, as the object is tracked with a complex shape that has been propagated from the previous time instant and fitted in the partition, particles that correctly represent the tracked object will have a value of the Battacharya coefficient very close to 1. Therefore, their likelihood of being selected in the *Resample* step (see Section 2.3) is high. In contrast, those particles that better represent the object in the classical approach very likely contain background pixels in their *object mask* as it does not fit the real shape of the object, which leads to a lower Battacharya coefficient value and a lower selection probability.

Estimation

The estimation of the object is obtained as the average of the state of the particles. In the classical approach, as all samples have the same associated shape, the average of the particles has the same shape and can be computed as the average of the parameters. Note that in the region-based case each particle has its own associated object shape obtained through region fitting (see Section 3.2). Thus, the object shape is estimated combining the masks associated with all particles.

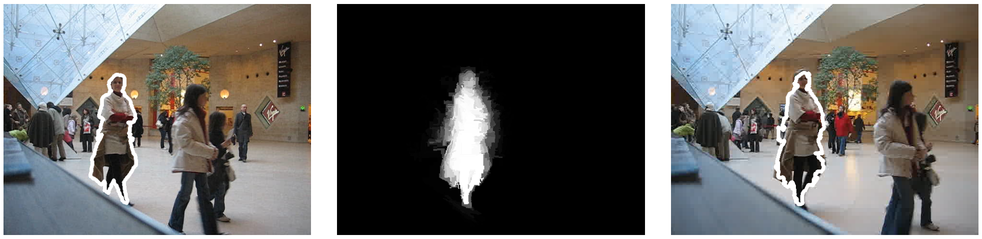


Figure 2.3: Extension of the color-based particle filter using region information. Left: tracking of a car with a classical particle filter that estimates the object with an elliptical shape. Right: result obtained with the proposed region-based algorithm.

Let $M^{(i)}(u,v)$ be the binary mask associated with particle i. Then the average mask $A_M(u,v)$ is computed as:

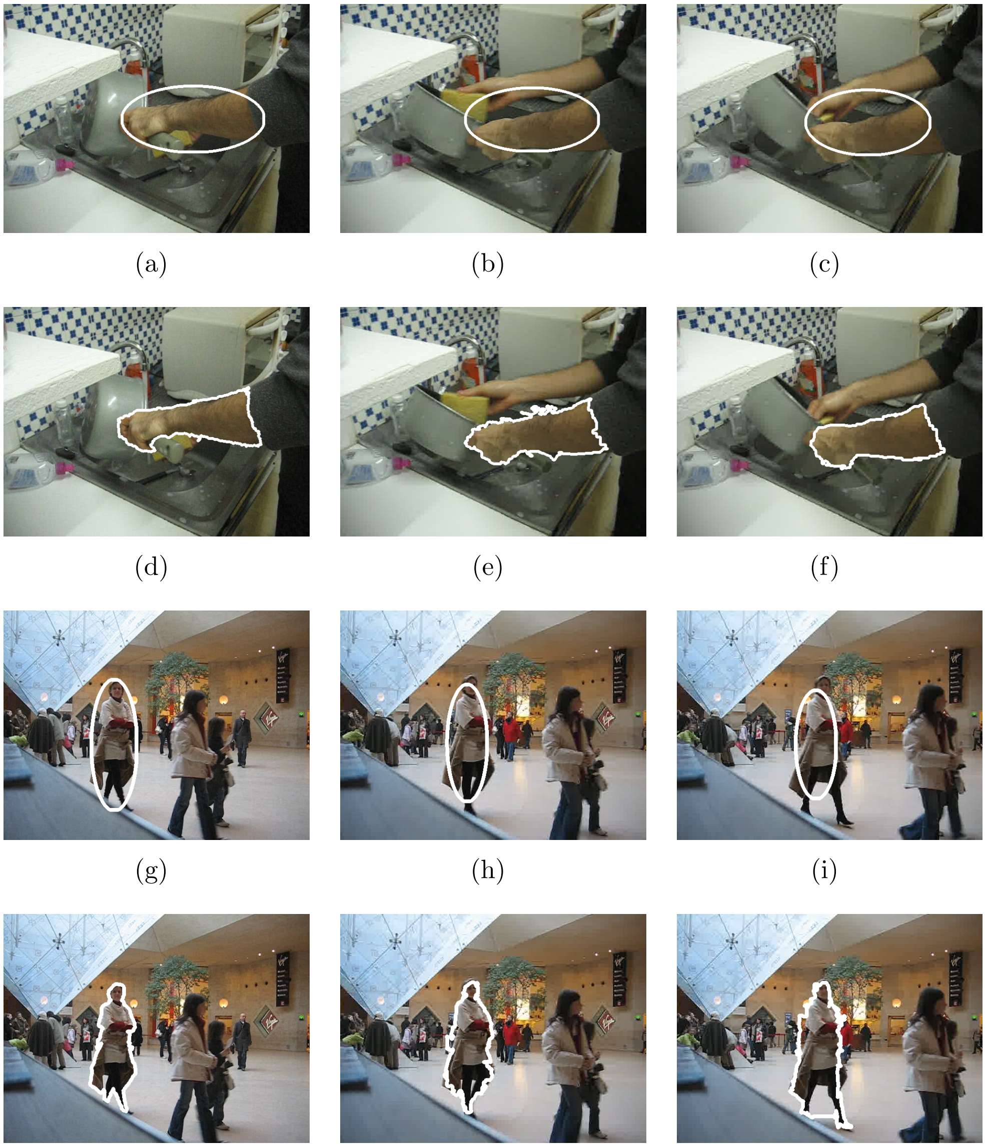
$$A_M(u,v) = \sum_{i=1}^{N_s} M^{(i)}(u,v)\pi^{(i)} \tag{2.19}$$

where N_s is the total number of particles and $\pi^{(i)}$ is the weight of the particle after the *Evaluation* step (see Section 2.4). The shape of the estimated object will be formed by all those pixels with a value higher than a threshold T_o. In this work T_o has been set to 0.5. As each mask $M^{(i)}$ is composed by a set of regions of the current partition P_k, the estimated object will also be composed by a certain number of regions of P_k. The use of regions of P_k to accurately estimate the object shape allows a correct model update. The object color is modeled as a class conditional color distribution computed with a histogram in the RGB color space. Therefore, given a pixel (u,v) with color $I(u,v)$, the likelihood of the pixel given that it belongs to the object is $p(I(u,v)|O) = h_O(I(u,v))$, where h_O is the object histogram. This histogram is initially learned from the object segmented in the first frame of the sequence.

The object model is updated at each frame, combining the previous model with the model derived from the current frame [64].

$$h_{O_k} = \alpha h_{O_{k-1}} + (1-\alpha)h_E \tag{2.20}$$

where $\alpha \in [0,1]$ is the learning rate and h_E is the histogram of the estimated object at time k. In this work α is set to 0.8.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 2.4: Comparison between the color-based and the region-based particle filter.

2.5 Region-based particle filter

In this section, we propose a region-based particle filter to jointly perform video object tracking and segmentation. In contrast to the algorithm presented in the previous section, in which tracking and segmentation algorithms were coupled to benefit from each other, this technique relies on a new theoretical approach of the particle filter algorithm based on regions. To this end, a set of regions that represents the object is propagated along the sequence. The introduction of these regions in the algorithm has some implications that will be analyzed in this section.

This technique is an extension of the work presented in [65] using motion information to build a more accurate adjacent graph between regions from consecutive frames, allowing to capture fast movements in the sequence. As a consequence, both the computational cost and the computational time are drastically reduced. Thus, the hierarchical global optimization presented in [66] may be used to jointly propagate the set of particles. Furthermore, we introduce Bayesian Estimation to locally refine particles before the randomness associated to the particle filter is used in the particles perturbation step.

The algorithm is described using the structure presented in Section 2.3 to allow the comparison with the color-based approach.

Let us define a new representation of both the state and measurement for the tracking problem in terms of regions. In our work, states are formed by a union of regions from the image partition while measurements consider both the image and its associated partition. Formally:

$$x_k = \bigcup_{r}^{n_k^o} R_k^r \tag{2.21}$$

$$z_k = [I_k, P_k] \tag{2.22}$$

where $P_k = \bigcup_{r=1}^{n_k} R_k^r$ is a partition of the image I_k, n_k is the number of regions that form the partition and n_k^o is the number of regions that characterize the object with $n_k^o \leq n_k$.

Given that the state estimation x_k is formed using a set of regions from an image partition P_k, several object representations that could be computed at pixel level are not allowed. In other words, analyzing a partition and assuming that the solution must be formed by regions from this partition, drastically reduces the solution space Ω and bounds the maximum quality of the object representation that can be obtained. Moreover, particles propagation is no longer only dependent on their previous state, it also depends on P_k, which is part of the measurement z_k because it can only be computed once the image at the current time instant k is available. Thus, the new importance density is:

$$q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)}, z_k) \tag{2.23}$$

As shown in [49], choosing $p(x_k|x_{k-1}^{(i)}, z_k)$ as the importance density minimizes the variance of the weights $w_k^{(i)}$ so that the *Effective Sample Size* (N_{eff}) [55] is maximized. Replacing Equation 2.23 into Equation 2.7:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \int p(z_k|x'_k)p(x'_k|x_{k-1}^{(i)})dx'_k \tag{2.24}$$

This integral over Ω involves the states represented by all the regions of the partition and all their possible combinations. Although the number of states to be analyzed is usually hughe, it is a finite number. If all the possible unions of regions are considered:

$$|\Omega| = \sum_{n=1}^{n_k} \frac{n_k!}{n!(n_k - n)!} \tag{2.25}$$

Thus, Equation 2.24 can be represented by a summation:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \sum_{c}^{|\Omega|} p(z_k|x_k^c)p(x_k^c|x_{k-1}^{(i)}) \tag{2.26}$$

This summation becomes intractable using a brute force approach. In order to obtain the weights at each iteraction, for each particle, the probablity associated with the transition between its previous state and all the possible solutions in Ω (regions and combinations of regions of P_k) should be computed $p(x_k^c|x_{k-1}^{(i)})$. Then, the probability of each particular combination of regions should be evaluated $p(z_k|x_k^c)$ (Section 2.5).

Resample

The resampling only considers the support points of the tracked pdf represented by the particles and their associated weights. These weights have been previously computed in the *Evaluation* step. Thus, the resampling algorithm described for the color-based approach is applied at this point. However, the expression of the weights in the region-based approach presented in Equation 2.26 shows that this process is not performed at each time step. This can be inferred from the dependence between w_k^i and w_{k-1}^i.

The resampling step is performed according to the N_{eff} value to avoid the *degeneracy problem* [23]. In other words, at each iteration, the variance of the particle weights is taken into account computing the N_{eff}. When this value falls below a certain threshold (T_N), a resampling step is applied. Note that, as this value is proportional to the inverse of the weights variance, the number of resampling steps computed by the algorithm is considerably reduced when the importance density presented in Equation 2.23 is considered.

Propagation

The propagation of particles between consecutive time instants is usually divided in two steps ([43], [47], [67]). As it can be observed in Equation 2.15, the first step consists in a *prediction* of the object state parameters and it is performed to ensure a minimum quality of the particles estimation. Then, randomness is introduced by a noise process $v_{k-1}^{(i)}$ which is independent for each particle. This step is known as *perturbation* of the particle parameters, and it is used to introduce diversity in the algorithm handling multiple hypotesis at the same time.

In the classical particle filter approach, measurements are not considered in the propagation step to reduce the complexity of the algorithm as discussed in Section 2.3. This is equivalent to the estimation step in recursive tracking, which is performed before the measurement is obtained at each time instant. In contrast, in a region-based approach, measurement information should be taken into account in the propagation process as the image partition P_k, which is part of the measurement, defines the regions that form the propagated state. This allows our algorithm to exploit color and structural information from the partition in the particles propagation process extending the theory of the classical color-based particle filter.

Prediction

Prediction is performed to ensure a minimum quality of the particles final estimation. In the classical approach, previous states and measurements are used to predict each new particle without any information of the current time instant (Equation 2.3).

We propose to perform particle prediction as a global process using the information provided by all particles and the partition at the current time instant P_k. This step will create a new set of particles optimizing a certain score function over the representation of the object in two partitions between consecutive time instants. In order to compute a single operation for all particles, a co-clustering of both partitions is performed.

In [65], a co-clustering between consecutive image pairs was proposed to predict object movement. This prediction was adressed using a co-clustering scheme based on the work of [68], which was adapted to a tracking scheme. In [66], an extension of the co-clustering technique considering the hierarchies associated with each partition was presented. Although this work outperformed state-of-the-art video segmentation algorithms when sequences with small variations were considered, both techniques showed some weaknessess when the tracked object suffers strong movement or deformations because motion is not taken into account. Its keypoints for the case of two partitions are briefly described to motivate our choice and modifications, as well as to present some results.

The authors of [66] use an additive score function to measure similarities between segments of multiple partitions from closely related images. These similarities rely on color and texture information of the segments. Moreover, an adjacency graph is built connecting overlapping regions of different partitions. Finally, a linear optimization process constrained by the hierarchical dependencies of these regions is performed to obtain the final co-clustered partitions. Despite the challenging results, their method is designed for labeling objects which closely maintain shape and position among partitions. In order to predict fast movement and deformations of the object between consecutive images, we propose to introduce movement information in the clustering process in the computation of interactions between regions. Moreover, the adjacency definition is crucial in any co-clustering process. For regions belonging to different partitions, this has a direct impact on their interactions. Previous co-clustering approaches ([68, 69, 66]) define adjacency between regions from different partitions as region overlapping without considering motion. In order to robustly link objects through different images, we compute the optical flow between consecutive images using [70]. We propose to introduce motion information at two stages of the algorithm:

Interactions between regions from different partitions (Intra similarities): similarities between regions from diferent partitions are computed as a similarities combination of their neighboring contour elements ([68, 66]). In our work, we propose to compare a given contour element c at position (x, y) with all contour elements close to $(x + f_x, y + f_y)$, where $f(x, y) = (f_x, f_y)$ is the optical flow at the element position.

Adjacency definition: Regions $R_i^{r_1}$ and $R_j^{r_2}$ from partitions P_i and P_j respectively are considered adjacent if at least a pixel from the motion compensated version of $R_i^{r_1}$ overlaps with a pixel of $R_j^{r_2}$.

After the co-clustering step, the object estimation represented by each particle is formed by a set of regions from P_t. These estimations are obtained propagating particle labels with a single optimization process. Each particle represents a point in the solution space Ω. As these points may not be the best estimations of the object, random sampling is further used in the perturbation step to analyze other neightbouring solutions. In order to improve the final set of particles obtained by this random sampling, Bayes theory is used to refine particles before perturbation is introduced in the algorithm.

Both object and background probabilities are considered for a subset of regions from P_k. Then, a new set of particles $\{p_A\}$ is created adding or substracting these regions from the propagated particles. This process is performed taking into account the relation between their object and background probabilities. Let us consider a particle $x_k^i = \bigcup_r^{n_k^o} R_k^r$, where R_k^r is a region from P_k. Moreover, let us assume that a set of pixels M_{k-1} from the previous frame is used to create both object and background models. Then, for each region of the particle and its neightbourhood, object and background probabilities are obtained using the Bayes Theorem:

$$p(R_k^r \in O_k | M_{k-1}) = \frac{p(M_{k-1} | R_k^r \in O_k) p(R_k^r \in O_k)}{p(M_{k-1})} \tag{2.27}$$

$$p(R_k^r \in B_k | M_{k-1}) = \frac{p(M_{k-1} | R_k^r \in B_k) p(R_k^r \in B_k)}{p(M_{k-1})} \tag{2.28}$$

Regions with at least one pixel at a distance from any pixel belonging to the particle below a certain threshold T_d are considered neightboring regions of the particle.

The probability of the model obtained at the previous time instant assuming that the region R_k^r under analysis belongs to the object is computed as the average likelihood of the model considering all the region pixels:

$$p(M_{k-1}|R_k^r \in O_k) = \frac{1}{N_k^r} \sum_{i,j \in R_k^r} p(M_{k-1}|I_k(i,j) \in O_k) \tag{2.29}$$

where N_k^r is the number of region pixels and I_k is the image under analysis. The same concept is applied to obtain the term associated with the background.

The probability of the region to belong to the object/background ($p(R_k^r \in O_k)/p(R_k^r \in B_k)$) without any prior information is computed taking into account only its position in the image. These probabilities are computed as the likelihood averaged over all the region pixels of belonging to foreground/background after convolving the estimated object at the previous time instant with a Gaussian kernel. This kernel is used to model the dynamics of the object as they are a priori unknown. Finally, the probability of the set of pixels used to compute the models $p(M_{k-1})$ is computed as the percentage of pixels from the previous image used to update the model.

Once the probability of belonging to object and background is computed for all the neighboring regions, these probabilities are directly compared to form a new set of propagated particles.

A new set of predicted particles $\{p_A\}$ is created using local Bayesian information to refine particles obtained after the global co-clustering process. For each particle, all its regions and neighboring regions are analyzed. We use the Bayes Factor (B) introduced by [71] to capture the relation between object and background probabilities for each region. It is computed as:

$$B_k^r = 2ln(\frac{p(R_k^r \in O_k|M_{k-1})}{p(R_k^r \in B_k|M_{k-1})}) \tag{2.30}$$

were B_k^r is the Bayes Factor of the region r from partition k.

For each particle, all its neightboring regions with a large Bayes Factor are merged with the particle. On the contrary, regions belonging to the particle with a B under a certain threshold are considered as background regions.

Perturbation

The second step of the propagation process is the perturbation of the estimated particles. This step is crucial to introduce diversity between particles and create multiple hypotheses leading to a good estimation of the object when combined. Randomness is used to generate these hypotheses.

As previously, let us consider Ω the subspace formed by all the regions from a partition P_k and all their possible combinations. This is the solution space of our tracking problem. Let us consider a co-clustered partition P_k^C after the estimation step. As the optimization has been globally performed for all particles, P_k^C also defines the union of regions from P_k that form each particle. Thus, N_s points of Ω are sampled to analyze. Moreover, these points have been refined using the Bayes Factor to obtain a new set of object estimations $\{p_A\}$. We name these points as *anchor points*. Each anchor point is formed as a union of regions $x^{(i)} = \bigcup_r^{N_r} R_{r,P_k^C}$.

Some regions that form the particle may belong to the object while others may not. In order to find better estimates of each anchor point, we will randomly search the best representation of the object in a neighborhood of these points included in Ω.

Two statements have been taken into account to perform this search. First, as showed in Equation 2.23, constraining the representation of the object to be formed by a set of regions from a partition leads to an importance density function in which the measurement is involved. This means that we can use the information from both image and partition to generate new particles. Second, using this density function the weight of each particle only depends on its representation at the previous time instant as presented in Equation 2.26. Thus, we can select as *perturbed particle* the best estimation of each anchor point in a restricted subspace of Ω without any variation on its weight.

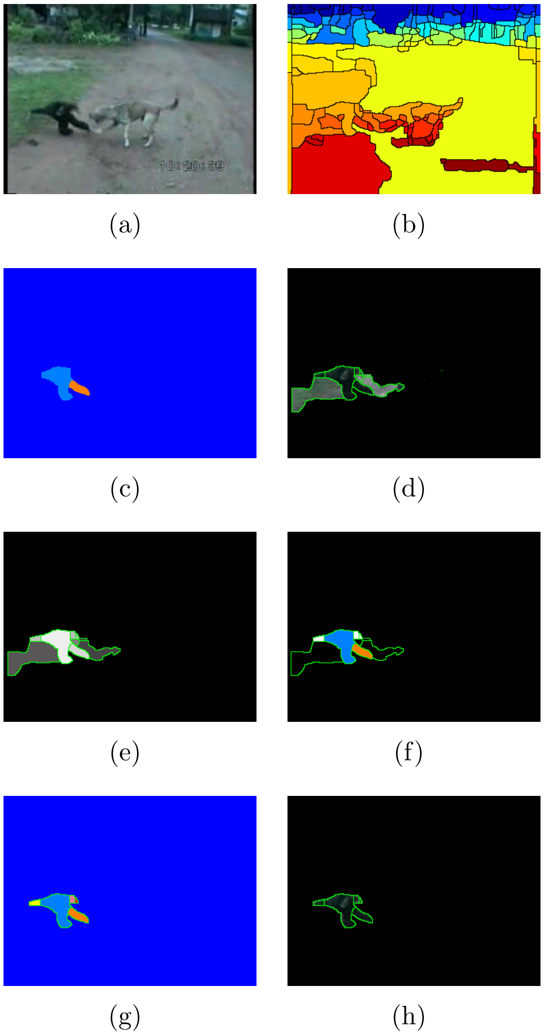
Each particle $x^{(i)}$ is perturbed as follows. First, a distance between the particle and each region of the partition is estimated. This distance is calculated as the average of Euclidean distances from each region pixel to the closest pixel of a particle region. Regions which are closer than D pixels (typically 100) are considered as candidates to be added to the particle, whereas all the regions that form the particle are considered as candidates to be suppressed. Then, the likelihood of these regions to belong to the object is obtained as $L(R_j) = Q_{k,k-1}(l,j) + Q_{k-1,k}(l,j)$ to ensure real values, where l is the union of regions that represent the object in $k-1$ and

j is a region from k. Finally, this likelihood is normalized ($L'(R_j)$) in the range $[0, 1]$ being 0 and 1 the selected regions with lower and higher scores from P_k respectively. The probability of change of each region is defined as the probability of the region that belongs to the particle to be suppressed and vice versa. It is computed as:

$$p_C(R_j) = \begin{cases} 1 - L'(R_j), & \text{if } j \text{ is part of the anchor point} \\ L'(R_j), & \text{otherwise} \end{cases} \tag{2.31}$$

Regions to be changed are randomly selected. Each region is selected with a probability $p_S(R_j) \propto e^{(p_C(R_j))}$. Once a region is selected, it is changed (included or suppressed from the particle) if a realization from a uniform random variable is lower than $p_C(R_j)$. This process is repeated until C changes have been produced, creating C potentially new particles. Those changes from potential new particles with a Diffusion distance lower than the same value of the initial anchor point are applied and a new particle is generated.

In Figure 2.5, an example of both prediction and perturbation is presented. In this figure, the prediction and perturbation processes are shown for a concrete frame (Frame 6) of the monkeydog sequence. The search of a better estimate for a single particle after the co-clustering step is presented. Images *(a)* and *(b)* show the original image and its segmentation. Once the co-clustering is performed, the object segmentation of Image *(c)* is obtained as a result of the estimation step. Let us consider a particle which is the union of the two regions that form the monkey in Image *(c)*. The task of the perturbation step is to randomly search if the particle contains regions that belong to the background or some parts of the background should be included as parts of the particle. The regions analyzed in this process are presented in Image *(d)* and their associated probabilities to belong to the object are showed in Image *(e)*. However, despite this probability, only three regions improve the Diffusion distance between the model and the particle when a region is added to it. These regions are represented in white in Image *(f)*. These regions are chosen with a certain probability and, after combining the information of all the particles, the segmented object is presented in Image *(g)*.



(a) (b)

(c) (d)

(e) (f)

(g) (h)

Figure 2.5: Prediction and perturbation example.

Evaluation

In the evaluation step, particles are weighted according to Equation 2.26. As each particle represents an object segmentation of the image, we compute these weights with an expression based on region similarities using the additivity property [68]. This way, we reduce the huge computational effort of comparing the particle in the previous time instant with all possible combinations of regions from P_k.

Thus, probabilities between combinations of regions, the model and the particle at the previous time instant are assumed to be proportional to scores of a similarity matrix:

$$\begin{aligned} w_k^{(i)} &\propto w_{k-1}^{(i)} \sum_c p(z_k|x_k^c) p(x_k^c|x_{k-1}^{(i)}) = \\ &= w_{k-1}^{(i)} \sum_c \left(m^T Z x_k^c \right) \left((x_k^c)^T Q' x_{k-1}^{(i)} \right) = \\ &= w_{k-1}^{(i)} m^T Z X Q' x_{k-1}^{(i)} \end{aligned}$$

where $w_{k-1}^{(i)}$ is the weight of the i_{th} particle at the previous time instant, m, x_k^c, $x_{k-1}^{(i)}$ are binary vectors encoding the regions that form the object in the initial partition, a certain combination of regions from P_k and the regions that formed the particle in P_{k-1}^o respectively. Matrices Z and Q' contain similarities between regions from the model and each region from P_k and similarities between each region from P_k and the regions that formed the particle in $k-1$. Finally, matrix X is formed by the summation of matrices created by all the possible combinations of regions from P_k. Note that matrix Z is computed only once for all the particles and Q' is formed using the information from Q previously computed in the prediction step of Section 2.5.

As all possible combinations of regions from P_k are considered, matrix X does not depend on a given particle. In fact, it can be computed without any other knowledge than the number of regions n_k, being the value of the elements in its diagonal equal to 2^{n_j-1} and to 2^{n_j-2} otherwise. Actually, we consider a matrix with elements equal to 1 in its diagonal and elements equal to 0.5 elsewhere because particle weights are normalized after this process.

Estimation

The estimation of the object is obtained as the average of the state of the particles. In the color-based approach, as all samples have associated a geometrical shape, the average of the particles has the same shape and can be computed as the average of the parameters. Note that in the region-based case each particle has its own associated object shape obtained through the propagation steps (Section 2.5). Thus, the object shape is estimated combining the masks associated with all particles.

Let $M^{(i)}$ be the binary mask associated with the *ith* particle. The average mask A_M is computed as:

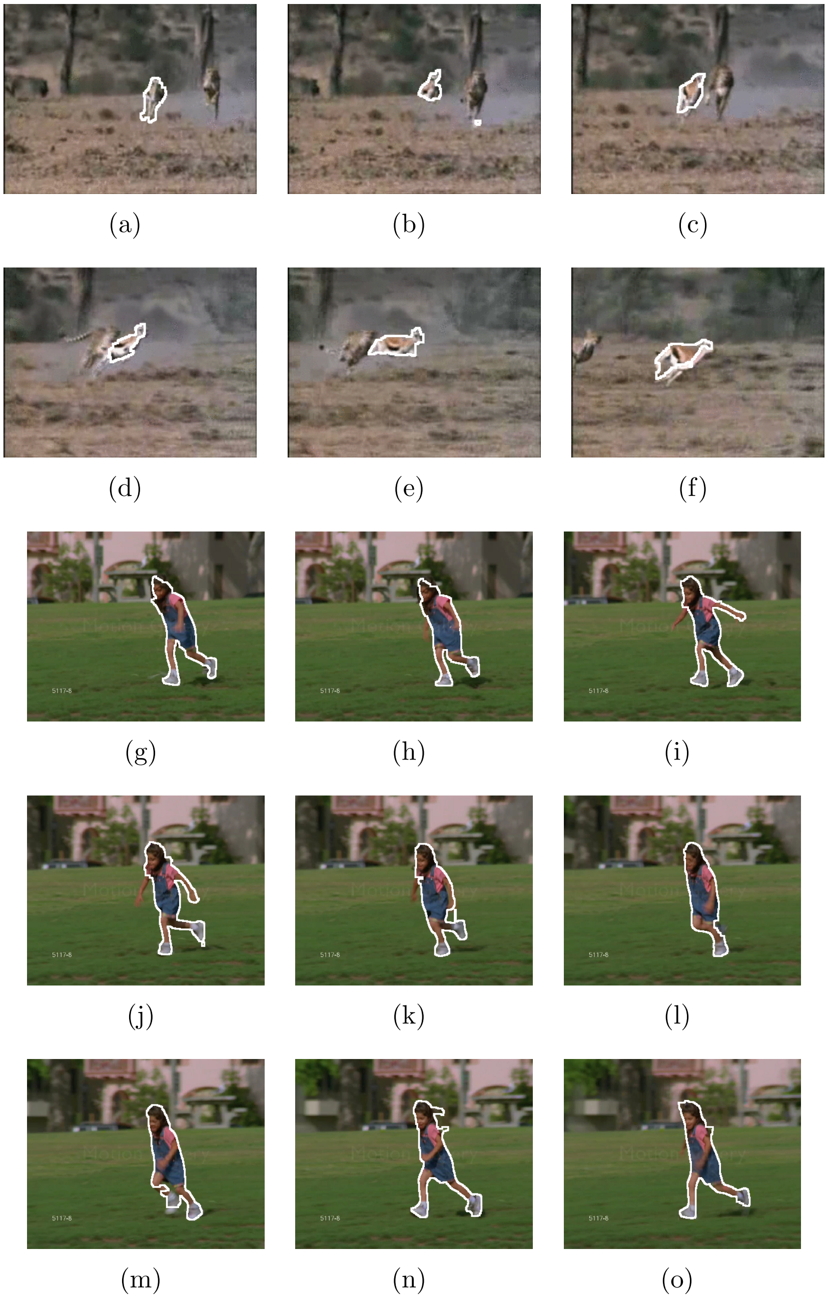
$$A_M = \sum_{i=1}^{N_s} w^{(i)} M^{(i)} \tag{2.32}$$

where N_s is the total number of particles and $w^{(i)}$ is the weight of the particle after the *Evaluation* step (see Section 2.5). The shape of the estimated object will be formed by all those pixels with a value higher than a threshold T_o. In this work T_o has been set to 0.5. As each mask $M^{(i)}$ is composed by a set of regions of the current partition P_k, the estimated object will also be composed by a certain number of regions of P_k.

2.6 Experiments

In this section, we present both qualitative and quantitative assessment of our region-based particle filter. This assessment is performed using three public datasets: LabelMe Video [29], the SegTrack dataset [26] and the SegTrackv2 dataset [28]. The first dataset is used to obtain tracking results and to analyze some statistics about the evolution of the object segmentation along the sequences. However, as objects are annotated as polygons in this database, it is not suitable to analyze the quality of their segmentation.

Segtrack and Segtrackv2 datasets are used for quantitave evaluation and comparison with other state of the art methods because of its accurate frame annotation. Moreover, a second set of sequences from the public database LabelMe Video [29] is used to obtain further graphical results and further insight. In all the experiments performed in this section, segmentations have been obtained using [9] and 80 particles have been used.



(a) (b) (c)

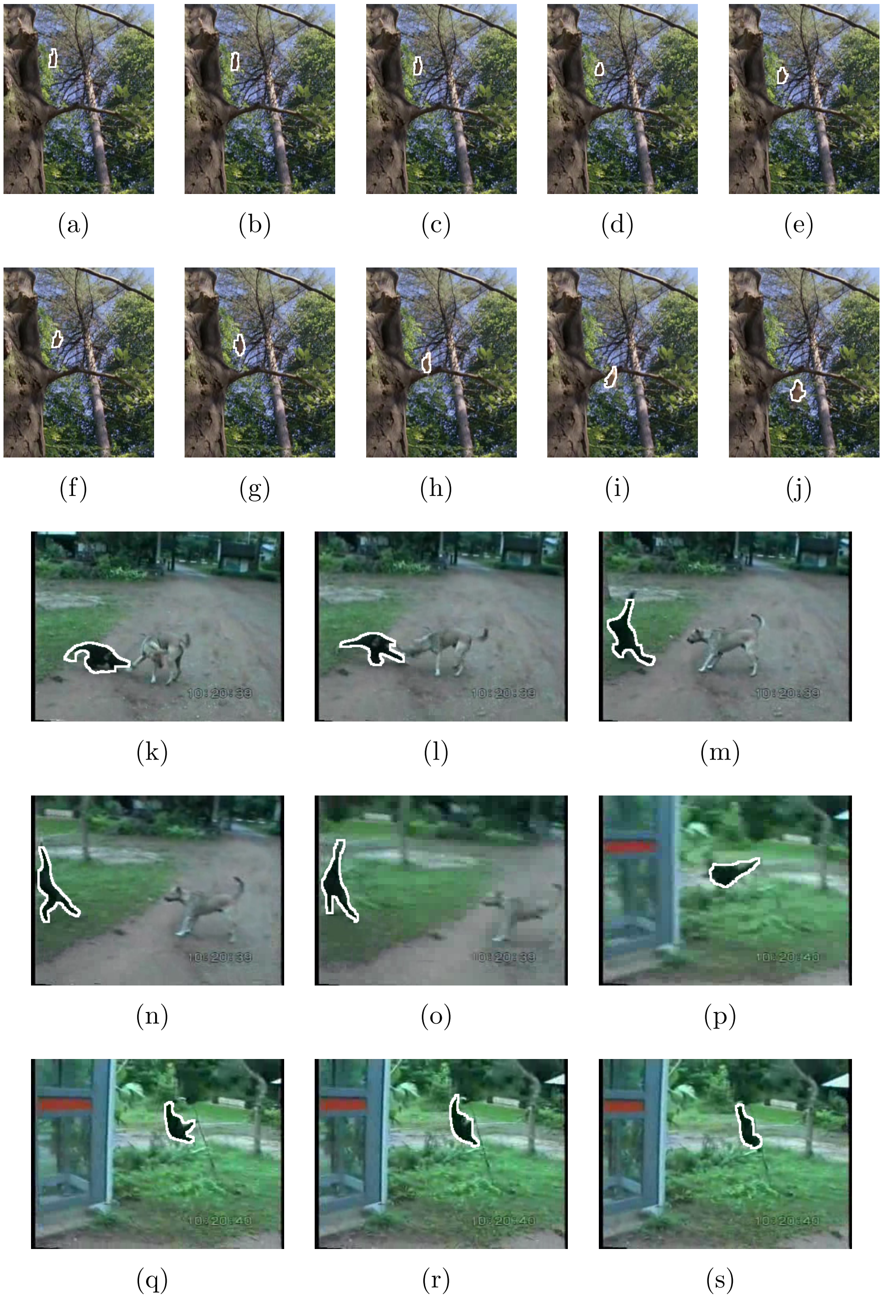
(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

(m) (n) (o)

Figure 2.6: Some results of the sequences Cheetah (rows 1-2) and Girl (rows 3-5). In this figure, we present the segmentation of a set of images from these sequences. These results have been obtained using the region-based particle filter with 80 particles.


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

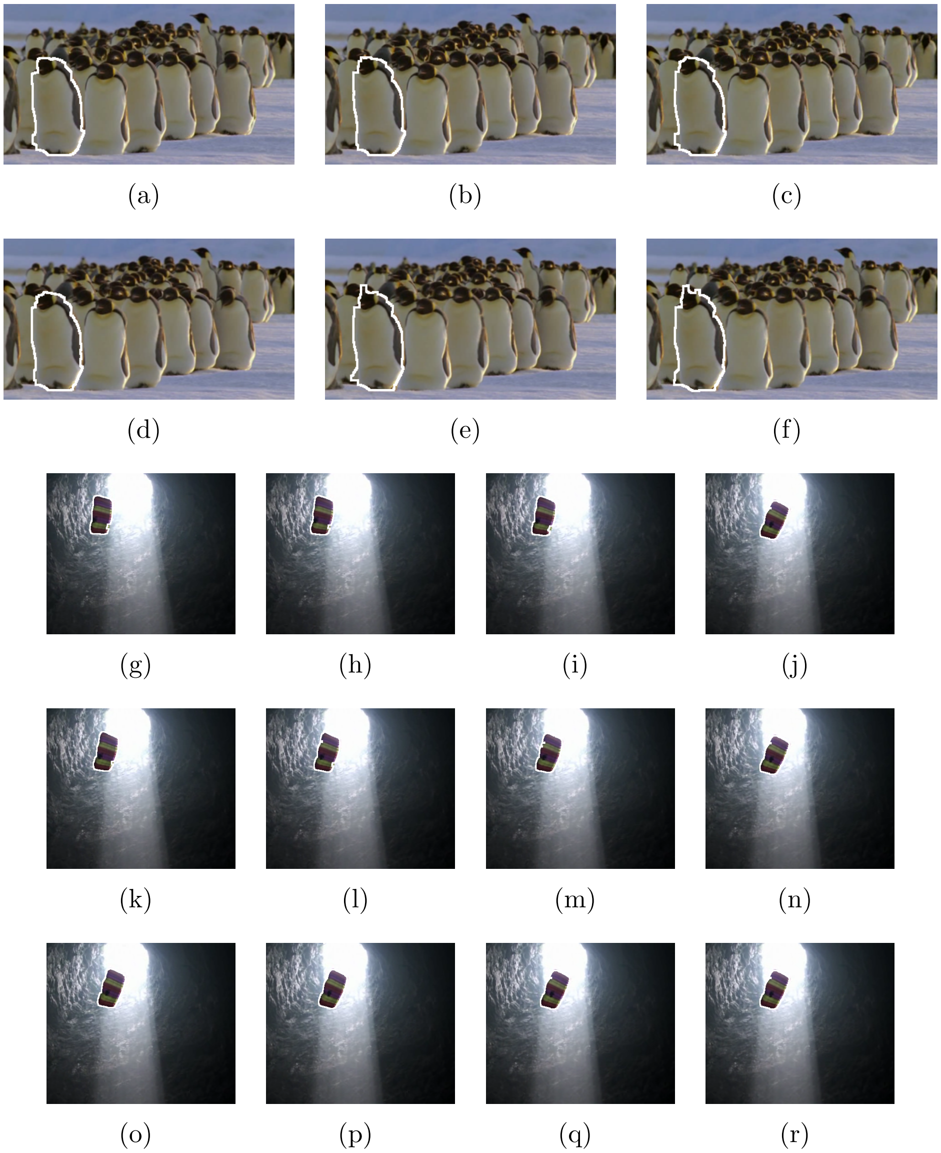
(f) (g) (h) (i) (j)

(k) (l) (m)

(n) (o) (p)

(q) (r) (s)

Figure 2.7: Some results of the sequences Birdfall (rows 1-2) and Monkeydog (rows 3-5). In this figure, we present the segmentation of a set of images from these sequences. These results have been obtained using the region-based particle filter with 80 particles.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i) (j)

(k) (l) (m) (n)

(o) (p) (q) (r)

Figure 2.8: Some results of the sequences Penguin (rows 1-2) and Parachute (rows 3-5). In this figure, we present the segmentation of a set of images from these sequences. These results have been obtained using the region-based particle filter with 80 particles.

	Error	**Variance**
Color-based	15.88%	2.78%
Region-based	13.40%	2.75%

Table 2.1: Tracking results of *Experiment 1*

Tracking

Experiment 1: The tracking accuracy of the region-based algorithm presented in Section 2.4 is analyzed in this experiment. This accuracy is assessed over the LabelMe Video dataset and it is computed in terms of the Euclidean norm of the error between the centroid position of the estimated and the real objects. In order to complete this analysis, the variance of this error is also calculated. This experiment explores the convenience of introducing regions in the tracking process.

However, before these metrics are calculated, the error associated with each dimension must be normalized with the height and the width of the object. The normalization step is required to relate the error with the size of the tracked object, as given an error it is more important when the object is smaller. The error for the sequence is computed as follows:

$$e = \frac{1}{L}\sum_{i=1}^{L}\sqrt{\left(\frac{\bar{x}_i - x_i}{w_i}\right)^2 + \left(\frac{\bar{y}_i - y_i}{h_i}\right)^2} \tag{2.33}$$

where $(\bar{x}_i, \bar{y}_i)$ is the centroid of the estimated object and (w_i, h_i) are the sizes of its bounding box in the ith frame.

These results are compared with the color-based tracking algorithm described in Section 2.3, which tracks the object as an elliptical region in the image. Tracking using the color-based approach is performed with 500 particles. The mean and variance of the error are presented for both methods in Table 2.1. As it can be observed, the error is reduced with the introduction of regions in the tracking process. This reduction can be achieved because of the accurate definition of the object shape and represents a 15.6% of improvement with respect to the error of the color-based approach. Moreover, the region-based tracker performs slightly better in terms of the error variance.

Experiment 2: In this experiment, the distance between the object centroid and both the color-based and the region-based approach presented in Section 2.5 with respect to the number of particles is analyzed. Here, the tracking quality of the region-based particle filter is assessed over the Segtrack dataset. In contrast to the previous experiment, we want to compare the performance of the region-based approach with the color-based approach when the number of particles used by the color-based particle filter is much larger than the number of particles used by the region-based technique.

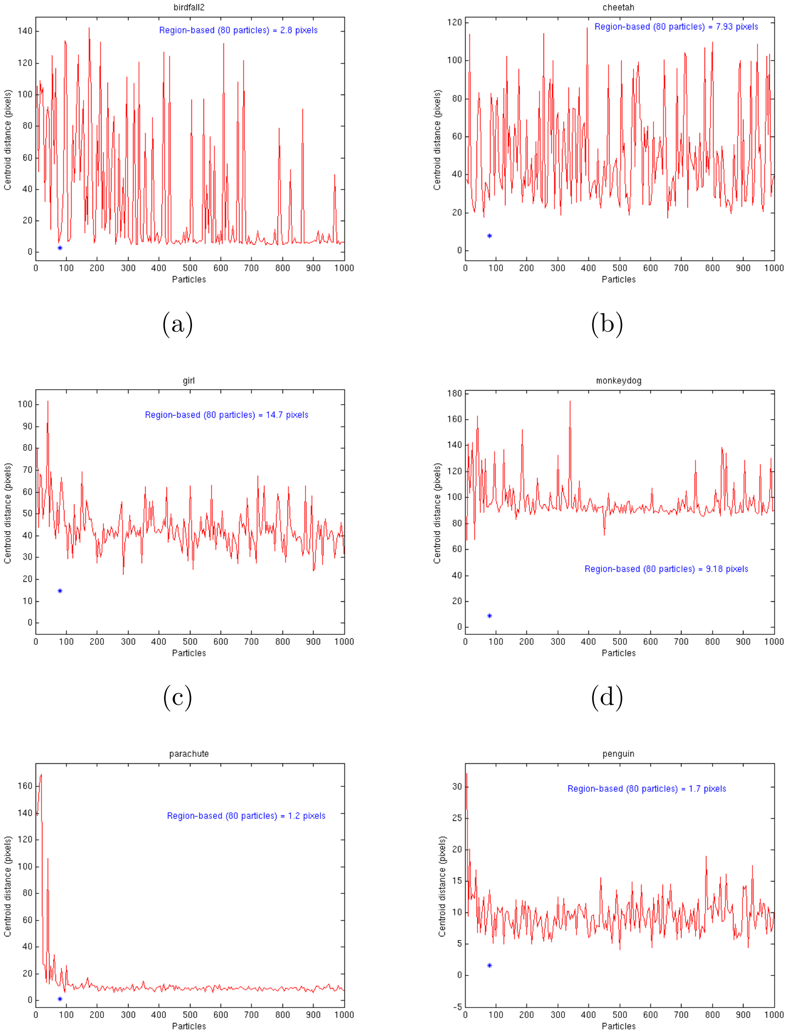
The results of this comparison are presented in Figure 2.9. In this figure, the Euclidean distance between the centroids of the tracked object and the estimation is presented for both algorithms. As each sequence of the dataset is analyzed independently, the error is not normalyzed using the size of the object. In the case of the region-based particle filter, the result is calculated using only 80 particles. As it can be observed, the region-based approach using this number of particles outperforms the color-based tracker using a number of particles in the range $[1, 1000]$ for all the sequences of the database.

Segmentation

Experiment 3: In this experiment, we evaluate the segmentation quality of the algorithm presented in Section 2.4 along time over the LabelMe Video dataset. This segmentation does not only depend on the particle filter algorithm. The role of the image partition in this process is crucial as the best segmentation of the object that can be obtained given a partition P (*upper bound* [72]) is formed by the union of regions in P.

Thus, the F-measure has been computed for the objects segmented by the proposed algorithm and for the upper bound given its annotated mask and its partition P. This representation has been obtained using the method proposed by [73]. The mean F-measure of the upper bound results averaged over all the frames of all the analyzed sequences is 0.89, whereas the mean F-measure of the traking results is 0.78. Therefore, the average loss in performance with respect to the upper bound is only 0.11.

To statistically analyze the temporal evolution of the segmentation, both the mean and the variance of the F-measure have been estimated at each time instant. This information is plotted in Figure 2.10. As it can be observed, the mean value of the F-measure provided by the region-based



(a) (b)

(c) (d)

(e) (f)

Figure 2.9: Distance between the object centroid and both the color-based (red) and the region-based (blue) approach with respect to the number of particles.

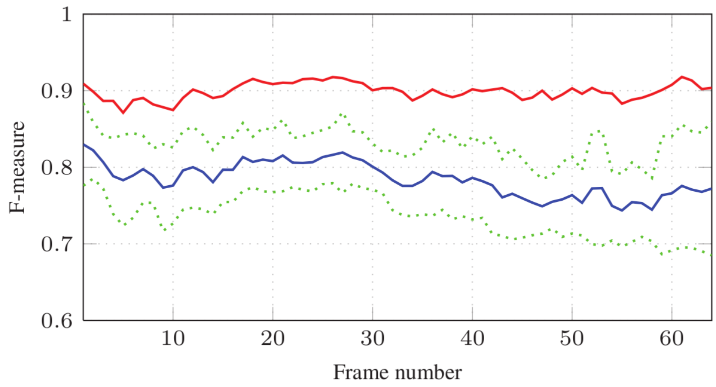


Figure 2.10: Statistical analysis of the segmentation performance using the F-measure. Red line represents the upper-bound results. Blue line shows the performance of the region-based particle filter and green lines represent its standard deviation.

particle filter (blue line) has an evolution close to that of the upper bound result (red line). The value of the correlation coefficient between these two time series is 0.9996. Such a high value indicates that the algorithm does not lose the objects being tracked, that the segmentation is estable along time and its results are consistently close to the upper bound ones.

Finally, The temporal evolution of the standard deviation of the F-measure results for the region-based algorithm is presented with green lines in Figure 2.10. These lines show again that the segmentation is very stable since that the disparity of the measure remains approximately constant.

Experiment 4: In this experiment, we analyze the evolution of the precision and recall measures along time. Specifically, the performance of the segmentations obtained with the method presented in Section 2.5 using these two measures over the SegTrack dataset is assessed.

In this case, the union of regions representing the object at each time instant is compared with its pixel level annotation provided by the dataset. Let us denote as O_M the binary mask representing the union of regions from P that define the estimated object. As it is described in Section 2.5, this mask is obtained giving a true value to those pixels from the averaged mask (A_M) with a value larger than a threshold T_o. On the other hand, let us define as GT the binary mask of the real object segmentation. Then, precision and recall are computed as follows:

$$P = \frac{|O_M \cap GT|}{|O_M|} \qquad\qquad R = \frac{|O_M \cap GT|}{|GT|} \tag{2.34}$$

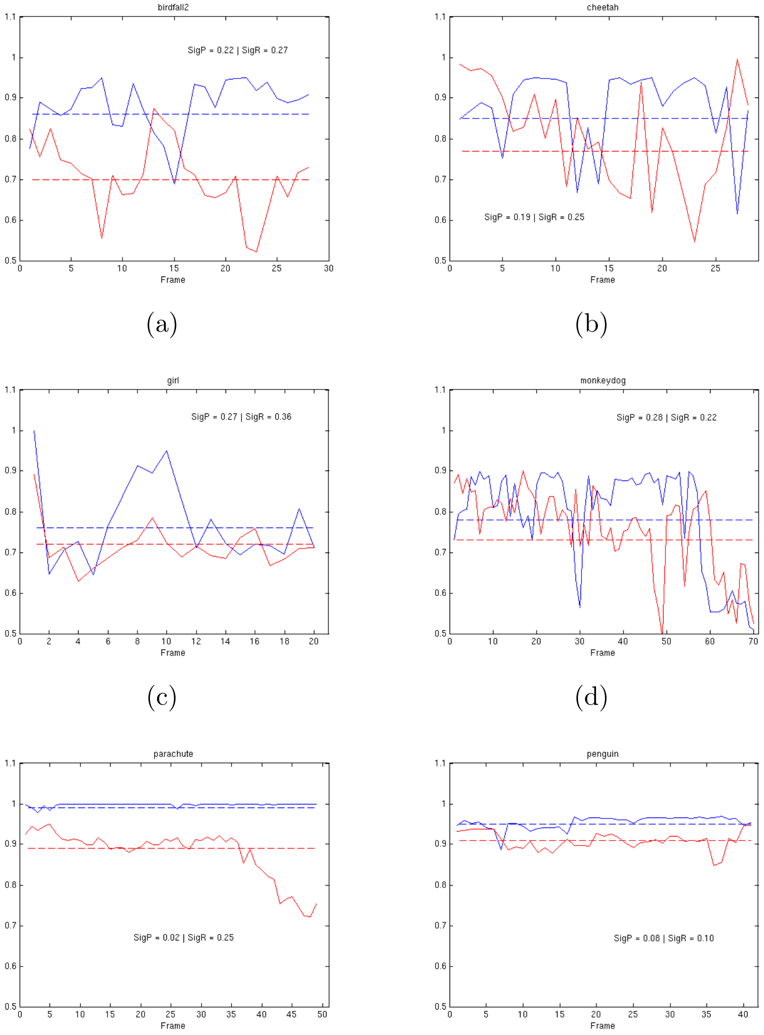
where $|\Delta|$ is an operator that counts the pixels with true value in the partition.

In Figure 2.11, the precision and recall for all the images in the SegTrack dataset can be observed. Moreover, the mean and the variance are presented for these sequences. As it can be observed, although the precision and recall values change along the sequence, they remain stable along time. This indicates two features of our region-based particle filter. First, although objects suffer strong movement and deformations in this dataset (See Monkeydog in Figure 2.7), our technique keeps segmenting the object iteratively without lossing the track. This feature shows that it is robust against rapid shape changes (Monkeydog), color changes (Cheetah), similar backgrounds (Penguin, Cheetah), deformations (Girl, Cheetah) and zooms (Birdfall2) (See Figures 2.6, 2.7 and 2.8).

Second, both measures remain stable during the sequences. In other words, there is not a decrease in the segmentation quality as time advances. This is also reflected in the low precision and recall variances. The only case in which a decrease in the recall can be observed between frames 37 and 50 is the Parachute sequence. This decrease indicates that a part of the object is considered as background, and it is caused by the initiall partition P_k. This partition merges part of the parachute with the background due to the darkness present at the bottom right part of the image.

Experiment 5: In this experiment, we present quantitative assessment of the region-based particle filter technique presented in Section 2.5. The SegTrack dataset [26] is used for this evaluation and comparison with other state of the art methods because of its accurate frame annotation. Moreover, some graphical results are obtained for further insight. In all experiments, segmentations have been performed using [9] and 80 particles have been used.

In order to quantitatively compare our results with other methods, we compute the *average pixel error rate per frame* as done in [40], [26], [38].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 2.11: Precision (blue) and recall (red). The mean of both sequence measures (dashed) and the variance are presented for all the sequences of the SegTrack database.

	Ours	[40]	[26]	[38]
Birdfall	**243**	288	252	454
Cheetah	**391**	905	1142	1217
Girl	1935	1785	**1304**	1755
Monkeydog	**497**	521	563	683
Parachute	**187**	201	235	502
Penguin	**903**	136285	1705	6627

Table 2.2: Segmentation results obtained with the region-based particle filter of Section 2.5 and comparison with other state-of-the-art methods.

This measure is computed as follows:

$$\epsilon(S_{1:N_f}) = \sum_j \sum_x \sum_y \frac{|S_j(x,y) - GT_j(x,y)|}{N_f} \tag{2.35}$$

where $S_{1:N_f} = \{S_1(x,y), ..., S_{N_f}(x,y)\}$ is the set of object segmentations, GT_j is the annotated ground truth of image j and N_f is the number sequence frames.

A detailed comparison of the proposed region-based particle filter with other method using this measure is presented in Table 2.2. As it can be observed in this table, our method outperforms the results of [40], [26], [38] in five out of six sequences of the SegTrack database.

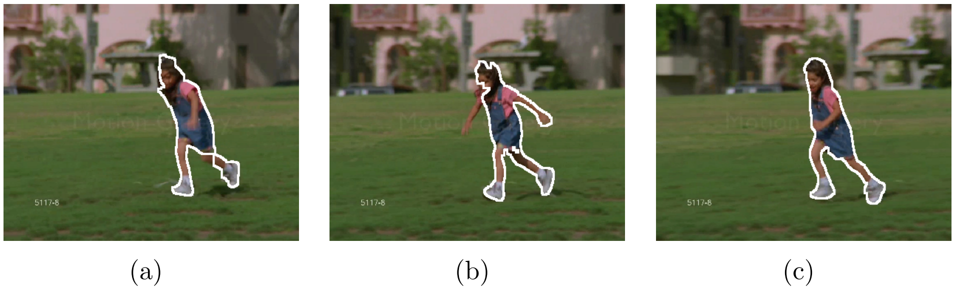
Note that this rate measures the average number of pixels that are misclassified per image without any distinction between foreground or background. To analyze the behaviour of the algorithm taking into account this distinction, in Table 2.3 we present mean and variance of precision and recall values for each sequence.

From the qualitative point of view, the results on the *monkeydog* video are particularly significant in two main aspects. First, the correct prediction of the object in a sequence with rapid movement is performed thanks to a tracking-oriented graph and a co-clustering scheme oriented to this task as presented in Section 2.5. This estimation would not be possible considering adjacency between regions as in [68]. Second, considering color information improves the result of the co-clustering when the shape of the object suffers strong deformations.

Furthermore, in Figure 2.12, several qualitative results of the sequence *girl* are presented. As it can be observed, our method produces robust ob-

	μ_P	μ_R	σ_P	σ_R
Birdfall	0.86	0.70	0.22	0.27
Cheetah	0.85	0.77	0.19	0.25
Girl	0.76	0.72	0.27	0.36
Monkeydog	0.78	0.73	0.28	0.22
Parachute	0.99	0.89	0.02	0.25
Penguin	0.95	0.91	0.08	0.10

Table 2.3: Precision and recall analysis.



(a) (b) (c)

Figure 2.12: Qualitative results of *Girl* sequence.

ject segmentations along the sequence. Images (b) and (c) show the frames with lower and higher average pixel error, respectively. The error introduced in Image (b) is mainly caused by the blurring effect of the arm when it moves very fast. However, the filter corrects these errors in only three frames even when a low number of particles is used (80). The perturbation step explores the space of solutions in an effective manner and finds satisfactory estimates for the original anchor points, being capable of both correctly segment the object and correct errors from other steps.

On the *cheetah* and *penguin* videos, the color information is not enough to perform a satisfactory segmentation of the tracked object. In these situations, shape descriptors and the orientation of the contours are the basis of a good performance. However, as the background is similar to the object, particles become very different and degeneration arises. This effect is eliminated using the resampling step and co-clustering, which fuse erroneous parts of the particles with the background. This is the process by means of which the parachute sequence achieves such a high performance.

The most challenging sequence for our algorithm is the *girl* video. In this sequence, an arm of the girl appears and the algorithm is not capable to track it because it does not have enough information. This is due to the fact that the Q matrix, which is involved in both the co-clustering and the random selection of regions to form the particles, is created using contour information. As we use an object segmentation of the previous frame and the other arm is not part of the contour, the co-clustering does not select the arm as a part of the object. Moreover, as the probability of selecting this region to include it as a part of the object is related with its similarity with the object used by the co-clustering, the likelihood of being selected is very low.

2.7 Conclusions

In this chapter, we have presented a novel technique for video object segmention based on a formulation of the particle filter algorithm in terms of a region-based image representation. We have explored the theory of particle filters and we have analyzed how the formulation and the approximations change when regions from an image partition are considered in the process.

In order to efficiently propagate particles, a hierarchical global optimization has been used to jointly propagate the set of particles that are used to obtain the object representation. Furthermore, we introduce a Bayesian Estimation step to locally refine particles before the randomness associated to the particle filter is used in the perturbation step. Using this refinement, the quality of the final segmentation is further improved.

Our approach has been asssed over the LabelMe and the SegTrack databases producing robust object segmentations and leading to competitive results compared with the state-of-the-art.

Chapter 3

Co-clustering

3.1 Introduction

The goal of unsupervised video segmentation is to efficiently extract coherent groups of voxels from sequences to represent the video information with many less primitives [16]. Such segmentation is useful for several higher-level vision tasks such as activity recognition [74], object tracking [75] and content-based retrieval [76]. In this context, temporal coherence of voxels along the sequences is one of the most important challenges.

Image segmentation approaches this problem computing an independent segmentation for each frame and obtaining further frame-to-frame region correspondances. These techniques may produce unstable sementation results, due to the fact that even small frame-to-frame changes cannot be expressed as a continuous function in general. Consequently, posing video segmentation as spatial region matching problem cannot always enforce consistency of region boundaries over time in the same way as volumetric do. For volumetric techniques, short-term temporal coherence can be obtained by a direct generalization of image segmentation techniques to the 3D domain. However, for both types of techniques, a hierarchical approach that extends pure pixel-level algorithms is necessary to obtain long-term temporal coherence as it is demonstrated in [1].

Following [2], video segmentation techniques can be classified into three categories: (a) frame-by-frame processing, that handles every frame of the sequence separately leading to low temporal coherence results [77], (b) streaming or iterative processing, that relies on a few previouly processed frames and improves temporal coherence with respect to the previous approach while requiring reasonable algorithm complexity [78] and (c) 3D volume processing, that jointly analyzes the whole video sequence and leads to the best results in terms of coherence but implies high complexity algorithms and memory requiments to perform this task [1].

Regardless of the previous classification, it is nowadays widely accepted that multiresolution descriptions provide a richer framework for subsequent analysis, both in the image [9] as in the video case [1]. This way, current techniques mainly rely on motion information to build a set of coherent partition sequences, describing the video at different resolutions.

Video sequences with global motion or little variation in the scene pose problems to motion-based segmentation approaches. In these cases, to strongly rely on motion information does not help to infer the semantics in the scene and can decrease the segmentation performance. Figure 3.1 presents an example of this behaviour, which can be worse in the case of cluttered background. In this example, it can be observed that standard state-of-the-art video segmentation techniques do not efficiently represent the sequence with a reduced number of voxels due to the small variations between consecutive images.

Co-clustering techniques aim at robustly segment a reference image (or various reference images) within a collection of closely related images (for instance, multiple views of a given scene, a set of medical images associated with the brain or a video sequence with small variations). When the number of resulting clusters is not known a priori, this task is also called correlation clustering [79].

These methods rely on finding similarities between image regions among the collection of images and optimizing a certain score function. In this process, a region adjacency graph that considers both spatial and temporal information is also taken into account. Co-clustering approaches that model the problem as a Quadratic Semi-Assignment Problem [80] have been reported to outperform other co-clustering strategies [68]. However, such solutions present inconsistencies on the clusters propagation among images which prevent to obtain a coherent labeling through the collection of video images.

To robustly handle sequences with small variations, we propose a video segmentation method based on the co-clustering of a sequence of region-based hierarchical image representations. Moreover, we extend this co-clustering to produce a multiresolution representation of the video sequence.

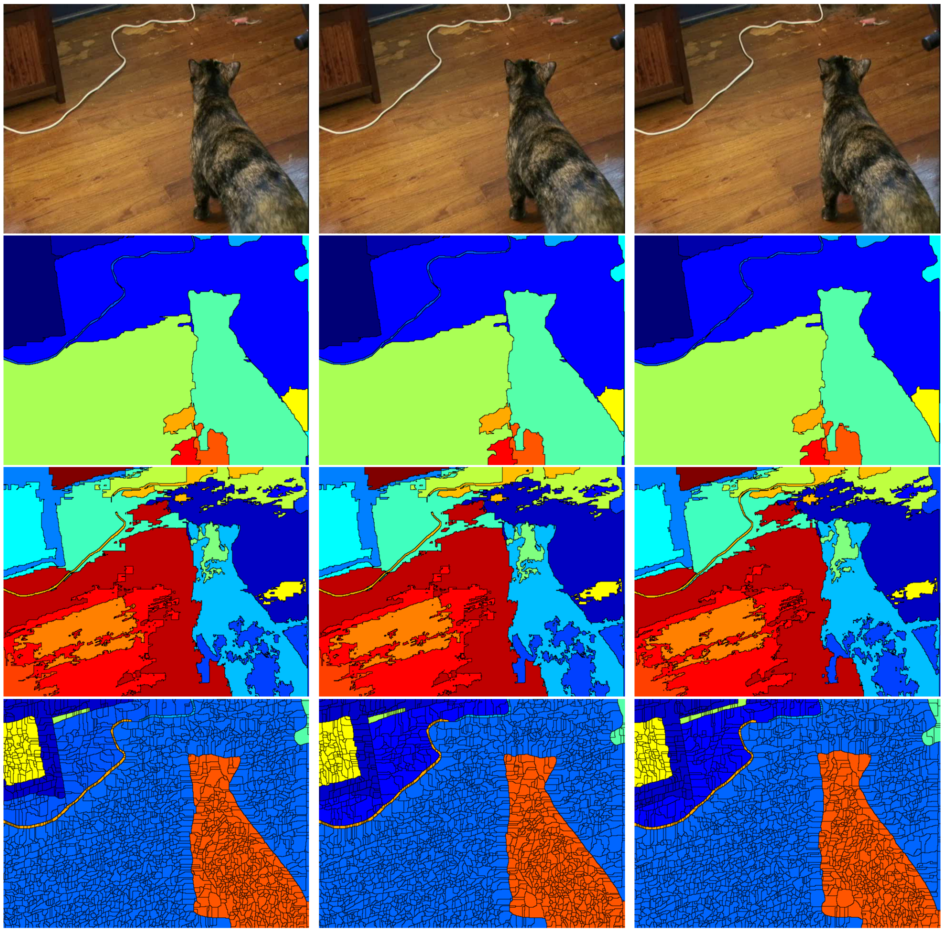


Figure 3.1: Example of the video segmentation results on a sequence with little variation. Results are obtained with the minimum number of regions to achieve a given quality. First row: original frames of sequence *zoe1* from the Video Occlusion/Object Boundary Detection Dataset. Second row: segmentation results of [1]. Third row: segmentation results of [2]. Fourth row: our results.

The main contributions of this thesis to the field of **co-clustering video sequences**, which are described in this chapter, are:

An **optimization on hierarchies** that fully exploits the tree information avoiding inconsistencies of previous co-clustering approaches. To this end, partitions are coded with boundary variables that allow an efficient representation of the hierarchical constraints (Section 3.5). In this thesis, we present that technique in a generic framework since we believe that it may have applications beyond that of co-clustering of video sequences.

An **iterative approach** for video segmentation based on the previous optimization process (Section 3.6), that combines the information at different resolutions.

This chapter is organized as follows. In Section 3.2, we review the most relevant techniques in the areas of video segmentation, co-clustering and co-segmentation, specially those that produce hierarchical results. In Section 3.3, we formally present the problem of co-clustering between images. Before introducing hierarchies in the co-clustering process, we discuss some concepts associated with hierarchies in Section 3.4. Then, the proposed co-clustering of hierarchies is presented in Section 3.5. This process uses a set of images and their associated hierarchies to obtain a multiresolution of partitions clustering hierarchy nodes. In Section 3.6, we present an strategy to apply this technique to the video segmentation problem, making the optimization problem tractable. Then, in Section 3.7, we present a set of evaluation tools developed to measure the quality of partitions and hierarchies. Some experiments are conducted in Section 3.8 in order to assess the proposed techniques. Finally, we draw some conclusions in Section 3.9.

3.2 Related work

In [1], a hierarchical graph-based method in which appearance and motion are used to group voxels is presented. This technique builds a coherent region-based representation of the entire video, processing it as a single stream. In our approach, we propose as well a multiresolution representation of the video sequence. Nevertheless, we avoid jointly processing the entire video and exploit the information provided by independent hierarchical segmentations. Note that this information is richer than the information provided by a single partition obtained independently for each image as different resolutions may be selected at different parts of the scene.

The concept of hierarchical graph-based video segmentation is also used in [2]. In this work, sequences are processed relying on motion information and using bursts of frames in order to reduce the complexity of the algorithm. The information of these bursts is combined to create a supervoxel hierarchy of the entire video. Sequence partitions are then obtained using the uniform entropy slice in [81]. In our work, we also process groups of images instead of the whole collection. Moreover, we iteratively propagate contour information at different resolutions. This allows us to reduce the complexity while the quality of the segmentations is preserved.

The work in [82] extends the hierarchical image segmentation of [9] to the case of video, including motion information. To make the approach tractable, [83] proposes a spectral graph reduction which allows defining an iterative segmentation process for video streaming. In our work, although we present a global framework, we also propose an iterative segmentation process to make the problem tractable.

Previous techniques decrease their performance when scenarios with small variations are considered (Figure 3.1) because motion does not help to describe semantics in the scene. To overcome this situation, we tackle the problem with a co-clustering approach.

In the context of biomedical imaging, [69] stated a coclustering problem as a Quadratic Semi-Assignment Problem (QSAP) and, as in [80], it tackled its solution with a Linear Programming (LP) relaxation approach. In [80], the optimization function is computed from distances between regions and linear constraints are imposed on these distances. This relaxation creates a number of inequalities that grows as $O(n^3)$, where n is the number of regions.

In [69], these constraints are only imposed over cliques in an adjacency graph on the regions. This approach bounds the number of constraints to $O(n^2)$. Moreover, a regularization parameter was introduced in [68] to avoid trivial solutions in the optimization process. Although these approaches reduce the complexity of the problem, the solution of the optimization presents inconsistencies. These inconsistencies appear because the proposed constraints do not force the solution of the problem to be a partition. Furthermore, inconsistencies have a bigger impact when initial partitions are oversegmentations of the original images.

In our approach, we also define the co-clustering problem as a QSAP, but partitions are defined in terms of boundaries between regions. This allows us to reduce the complexity of the problem. Moreover, we substitute the previous constraints by imposing the structure of the hierarchies; this way, in addition to preventing inconsistencies, resulting partitions are closer to the semantic level.

Closely related to co-clustering between image partitions is the problem of co-segmentation, first introduced by [84]. These methods take as input two or more images containing a common foreground object with varying backgrounds and attempt to segment the foreground object from the background. [85] extends the previous concept to the multiple foreground segmentation case. In it, the user has to define the number of background objects in the image collection and sets of adjacent regions (*candidates*) are selected from an initial segmentation. To obtain a tractable problem, every set of regions is represented as a tree. In our case, we do not require any parameter and, for each image, a single hierarchy is computed.

Co-segmentation has also been applied to image sequences in a single resolution framework ([86], [87]) or using hierarchies [88]. Note that co-segmentation algorithms would generally fail when tackling the case of scenes with small variations, since background in consecutive frames may also maintain its appearance. The work in [88] proposes an optimization process over the nodes of the hierarchy. The use of nodes to define the inter image relations for all levels of the hierarchies would lead to an unfeasible number of variables and constraints. This problem is tackled in [88] by restricting the inter relations to the highest level of the hierarchies. We solve that problem by defining the optimization process over boundary segments, which makes the problem tractable.

In this thesis, we propose a method to generate a multiresolution collection of coherent segmentations along a sequence with small variations. These segmentations are created clustering nodes from a set of non-coherent hierarchies associated with the video. This allows our technique to efficiently keep semantic contours at different resolutions and to eliminate random boundaries.

3.3 Co-clustering

Let us consider a set of images $\{I_j\} = \{I_1, I_2, ..., I_N\}$ and their associated partitions $\{P_j\} = \{P_1, P_2, ..., P_N\}$. Each of these partitions is formed by a group of n_j regions $P_j = \{R_j^1, R_j^2, ..., R_j^{n_j}\}$, where $P_j = \cup_{r=1}^{n_j} R_j^r$.

A co-clustering between all partitions is defined by a binary matrix $X \in \{1,0\}^{n \times c}$, where $n = \sum_{j=1}^{N} n_j$ is the total number of regions of the partitions and c is the number of clusters resulting from this process. Note that this number of clusters is not fixed by the algorithm because it is a priori unknown in a general situation. Each column x_l with $l \in \{1, ..., c\}$ corresponds to a single cluster that is formed with a set of regions from the partitions. A region is assigned to the l_{th} cluster if it has a true value at the l_{th} position of its row. Regions from partitions are assigned to only one cluster if matrix X is constrained to have rows with unit norm.

The quality of a cluster is measured by considering its intra and inter image interactions between the subsets of regions chosen from each image. A complex Hermitian affinity matrix is created to represent the similarity between frames:

$$Q = \begin{bmatrix} Q_{11} & \cdots & Q_{1N} \\ \vdots & \ddots & \vdots \\ Q_{N1} & \cdots & Q_{NN} \end{bmatrix} \tag{3.1}$$

where Q is a complex matrix of size $n \times n$ that contains N^2 blocks. Each of these blocks contains the information associated to the interaction of the regions from two partitions. The elements of the diagonal contain the information of the interaction between regions from the same partitions.

The score associated with a certain co-clustering X is computed as:

$$tr(X^T Q X) = \sum_{l=1}^{c} x_l^T Q x_l \tag{3.2}$$

where $Q \in \mathbb{C}^{n \times n}$ is a matrix that measures affinities between regions. This matrix is constructed using an *additive score function* over elements of the region contours [68].

Consider the union of two adjacent regions, $R_U = R_1 \cup R_2$ and a score function computed over the contour elements of regions R_1 and R_2. Then, this function is additive if the summation of the score of both regions is equal to the score of the contour elements that belong to R_U.

Let us consider that the contours of all the regions in the i_{th} partition are represented by q_i contour elements. Then, a union of regions J is represented using a vector $b_J^{(i)} \in \mathbb{C}^{q_i}$ in which, for each contour element $k \in \{1, ..., q_i\}$ of the union J we define $b_J^{(i)}(k) = e^{i\theta_k}$, where θ_k is the angle between the outward normal of $R_J^{(i)}$ at the k_{th} contour element and the X-axis. Let $B^{(i)} = (b_1^{(i)}, b_2^{(i)}, ..., b_{n_i}^{(i)}) \in \mathbb{C}^{q_i \times n_i}$ be a matrix that represents all the regions of the i_{th} partition in terms of their contours.

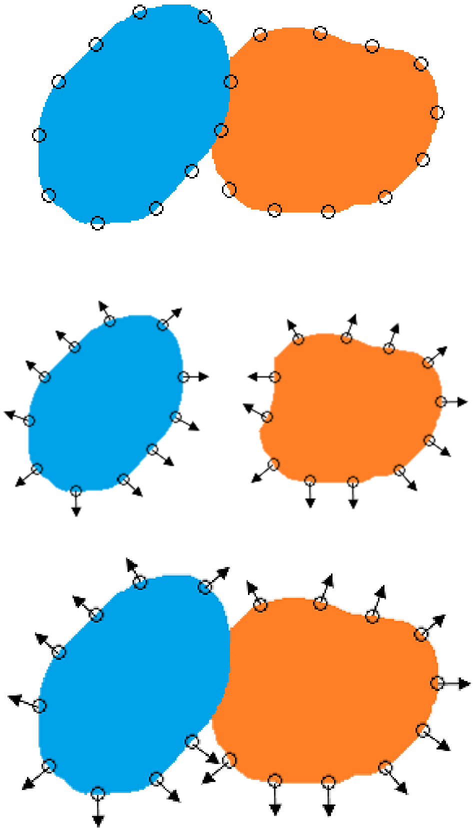


Figure 3.2: Example of union of two adjacent regions in terms of their contour elements. From top to bottom: Contour elements associated with two adjacent regions; the separation between both regions is defined by two elements. Outward normal of the regions at the elements positions; note that elements shared by both regions have opposite normals. Resulting contour elements and angles after the union of both regions; common elements are cancelled whereas elements belonging to only one of the regions are preserved.

Moreover, let us consider a vector $x^{(i)} \in \{0, 1\}^{n_i}$ that represent a union of regions from $P^{(i)}$. As it can be observed in Figure 3.2, common elements are cancelled because they have opposite normals. On the contrary, boundary elements that belong to only one region are preserved. So, $B^{(i)}x^{(i)} \in \mathbb{C}^{q_i}$ is a vector that contains the normal vectors of the contour elements of the union of regions described by $x^{(i)}$. A matrix $W^{(i,j)} \in \mathbb{C}^{q_i \times q_j}$ contains the similarity between contour elements from partitions $P(i)$ and $P(j)$. Then, a matching function between unions of regions from $P(i)$ and $P(j)$ respectively is:

$$(B^{(i)}x^{(i)})^H W^{(i,j)}(B^{(j)}x^{(j)}) \tag{3.3}$$

Thus, we define the correspondence between regions from partitions in i and j as:

$$Q^{(i,j)} = {B^{(i)}}^H W^{(i,j)} B^{(j)} \tag{3.4}$$

Matrix Q is computed with similarities between pairs of regions from the same partition (*Intra image similarities*) and from different partitions (*Inter image similarities*). In [68], intra image similarity is proportional to the number of contour elements that share both regions and to their color similarity. In turn, inter image similarities are captured comparing the HOG descriptor of the contour elements of the two regions. Then, co-clustering of both partitions becomes an optimization problem:

$$\begin{aligned} \max_X tr(X^T Q X) \quad s.t.\ & X_{i,j} \in \{0,1\}\ \forall i,j \\ & \sum_j X_{i,j} = 1\ \forall i \end{aligned} \tag{3.5}$$

This is a Quadratic Semi-Assignment Problem (QSAP) [69], and it can be expressed as:

$$\begin{aligned} \max_Y tr(QY) \quad s.t.\ & Y_{i,j} \in \{0,1\}\ \forall i,j \\ & Y_{i,i} = 1\ \forall i \end{aligned} \tag{3.6}$$

A Linear Programming relaxation for this type of problems was presented in [80] imposing distances between regions based on the triangular inequality. Further relaxation approaches ([69],[68]) make use of distances defined over cliques in a region adjacency graph. Considering these relaxations, the optimization process was stated in [69] as:

$$\begin{aligned}
&\min_{D} \sum_{i,j} q_{i,j} d_{i,j} \\
&s.t. \; 0 \leq d_{i,j} \leq 1 \quad d_{i,i} = 0 \, \forall i \quad d_{i,j} = d_{j,i} \, \forall i,j \\
&\quad\; d_{i,j} \leq d_{i,k} + d_{k,j} \quad \forall e_{i,j}, e_{i,k}, e_{k,j} \in G
\end{aligned} \tag{3.7}$$

where $e_{i,j}$ are vertices of the region adjacency graph G that encodes the connectivity of regions belonging to the input image partitions and $d_{i,j} \in \{0,1\}$ is an element of matrix D that stores the distance between regions i and j. Regions that belong to the same cluster have distances equal to 0.

Note that the approach of [19] was subject to trivial solutions which had to be avoided by adding a regularizing parameter. For example, in [19], the minimal symmetric difference between clusters is trivially obtained when all segments are put into one cluster. In contrast, this approach of co-clustering maximizes the sum of support for region boundary elements remaining after the mergings, and naturally avoids trivial mergers. The trivial solution of putting all regions into one cluster eliminates all boundary elements from the optimization so their contribution vanishes. The trivial solution of making no mergings at all leaves the objects in the image fragmented, decreasing the total support. The optimal operating point therefore lies somewhere in between these two ends of the solution space. An example of this optimization process is presented in Figure 3.3.

3.4 Working with hierarchies

In this thesis we explore region-based hierarchical image representations to describe semantic contours in video sequences with small variations. Towards this goal, in this section we start by discussing a few concepts related to region-based hierarchies.

Each node of the hierarchy represents a region in the image, and the parent node of a set of regions represents their merging. For simplicity, let us assume that this hierarchy is binary (regions are merged by pairs). This structure is referred to as Binary Partition Tree in [89]. Note that this assumption can be done without loss of generality, as any hierarchy can be transformed into a binary one.

Commonly, such hierarchies are created using a greedy region merging algorithm that, starting from an initial *leave partition* P^1, iteratively merges

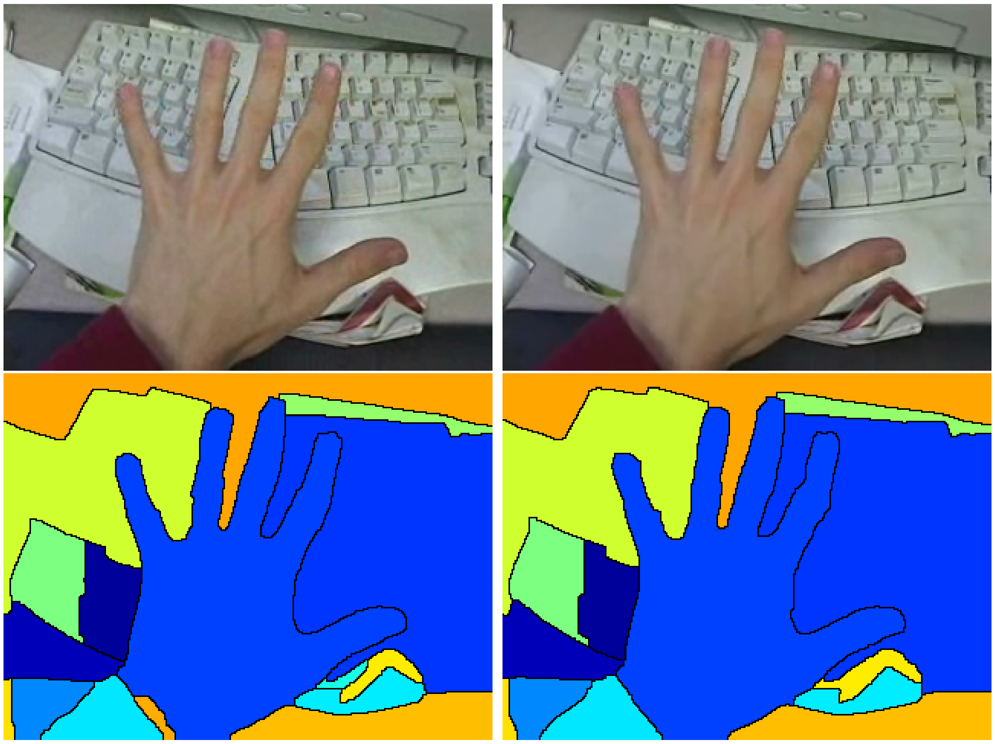


Figure 3.3: Co-clustering optimization example. At the first row two images involved in the optimization are shown. The result of this co-clustering is presented at the second row. Colors of the resulting partitions define unions of regions.

the most similar pair of neighboring regions under a certain similarity measure. The concept of region similarity is what makes the difference among the various approaches. The merging process ends when the whole image is represented by a single region, which is the root of the tree. The set of mergings that creates the tree, from the leaves to the root, is referred to as *merging sequence.*

In Figure 3.4, the merging sequence obtained for a given leaves partition and the partitions that are generated in this process are presented. Given the previous example, let us define a vector $b = [b_{1,2}\ b_{1,3}\ b_{2,3}\ b_{2,4}\ b_{3,4}]$ that encodes the boundaries between leaves. Using this notation, the partition generated after the first merging is represented by the sequence $[0\ 1\ 1\ 1\ 1]$, where 1 represents an active boundary. Note that not all possible configu-

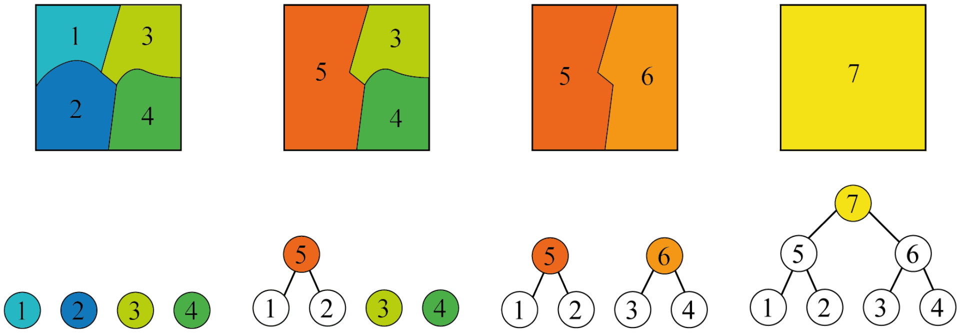


Figure 3.4: Partitions generated with mergings of regions from the initial leaves partition P^1. The evolution of the hierarchy at each step is shown below the correspondant partition.

rations of vector b generate a partition (i.e. [0 1 0 1 1]) but there are some possible partitions that are not included in the hierarchy.

In a binary hierarchy, a merging sequence contains N^1 partitions, where N^1 is the number of leaves (regions in P^1). This is the set of partitions that is usually analyzed when working with hierarchies. Still, we generate partitions which may not be included in the merging sequence. For instance, in Figure 3.4, the partition formed by $\{R_1, R_2, R_6\}$ would be generated and coded by the boundary combination [1 1 1 1 0]. Note that this partition is not represented in the binary hierarchy of the figure. This is done by analyzing all possible configurations of nodes in the hierarchy leading to a partition. Thus, we explore a larger number of contour combinations which allows us to use different resolutions at different parts of the image depending on its semantics. In other words, we use the semantics associated with nodes and their relations in the hierarchy to obtain coherent partitions, but we do not take into account the order in which mergings are performed in the tree.

An example with a real image that further illustrates the creation of a hierarchy and the possible manners to select nodes and combinations of nodes from it is presented in Figure 3.5.

In this example, the best object segmentation (O_S) in terms of quality and number of cluster using nodes of the tree is formed by the fusion of nodes 7 and 11. These nodes are selected from different scales of the tree and their union creates a coherent partition. Moreover, the set of partitions

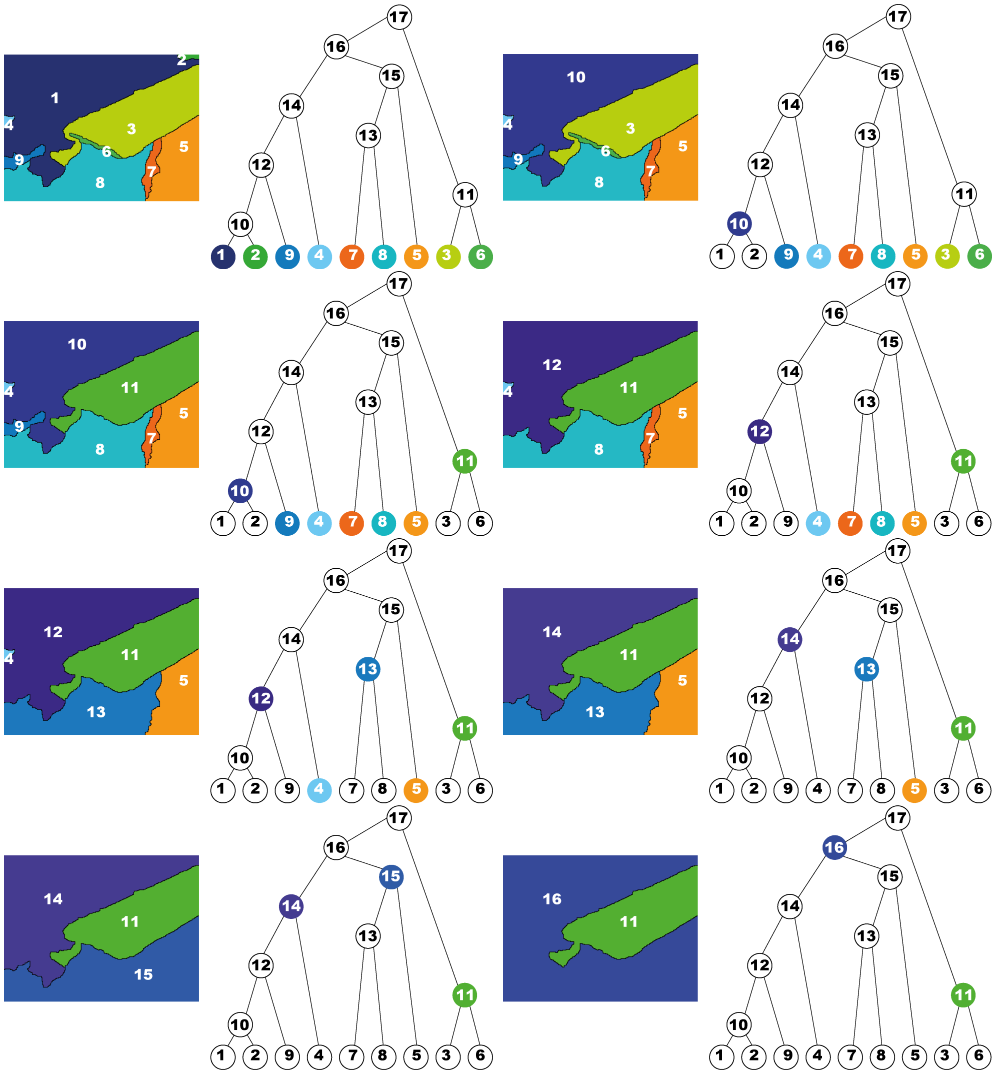


Figure 3.5: Partitions created using nodes of the hierarchy versus partitions from the merging sequence.

that can be obtained with the merging sequence are shown. As it can be observed, in this example O_S cannot be found in the merging sequence. Our technique creates co-clustered partitions using nodes of the hierarchy to introduce the semantics of the tree in the process while the number of possible solutions is not constrained to the partitions of the merging sequence.

3.5 Co-clustering of hierarchies

Let us assume that we have a collection of images, representing the same scene, which share a set of common contours but present a large number of random boundaries (e.g.: a video sequence with small variations or a multiple view scene representation). In this section we first present a global framework for, given such a collection of images and their associated and non-coherent hierarchies, obtaining a partition collection by clustering nodes from these hierarchies. This partition collection aims at keeping only the common contours and at producing coherent regions through the collection; that is, the various instances of the same object (or part) receive the same label in all the partitions of the collection (Figure 3.6).

This is achieved by coding in the *boundary matrix* the whole set of possible boundaries between adjacent regions in the collection. This matrix contains information about both the intra boundaries (between adjacent regions in the same image) and the inter boundaries (between adjacent regions in different images). The optimal boundary configuration (the co-clustering result) is achieved through an optimization problem that combines the boundary matrix information and the information about similarity between regions, which is coded in the *similarity matrix*. As previously, the similarity matrix contains the information about intra and inter similarities between regions. Intra similarities are computed using global region descriptors while inter similarities rely on descriptors computed over all contour elements. To avoid inconsistencies in the result, some constraints are impossed to the optimization process. In our approach, intra constraints are obtained from the hierarchies, whereas the common triangular equations are adopted as inter constraints. In addition, we extend the previous hierarchical co-clustering to a multiresolution framework.

Co-clustering problem definition

Formally, let us consider that we have a collection of M images $\{I_i\} = \{I_1, I_2, ..., I_M\}$ and their associated hierarchies $\{H_i\} = \{H_1, H_2, ..., H_M\}$. The merging sequence of a given hierarchy H_i defines a set of partitions $\{P_i^p\} = \{P_i^1, P_i^2, ..., P_i^{N_i^1}\}$, where P_i^1 is the *leave partition* on which the hierarchy is built and N_i^1 is the number of regions in P_i^1. The p-th partition of hierarchy H_i (P_i^p) is formed by a set of N_i^p regions $\{R_i^{p,k}\} = \{R_i^{p,1}, ..., R_i^{p,N_i^p}\}$, where $\Psi \in \mathbb{R}^2$ and $\Psi = \cup_{k=1}^{N_i^p} R_i^{p,k}\ \forall\ p$.

To encode all possible partitions $\{\pi_i^q\}$ ($\{P_i^j\} \subset \{\pi_i^q\}$) represented by a given hierarchy H_i, let us define its *intra boundary* matrix, $B_{ii} \in \{0,1\}^{N_i^1 \times N_i^1}$. This is a binary matrix whose components are variables that relate all regions in P_i^1. This way, $B_{ii}(m,n) = 1$ if, for the partition being coded, the boundary between leaves m and n is active; that is, if regions m and n have not been merged.

Note that, by correctly zeroing some elements of this matrix, the whole set of partitions in H_i ($\{\pi_i^q\}$) can be unequivocally described. This allows the co-clustering to fully exploit the richness of the hierarchical representation.

Boundaries between leaves of different partitions are coded in the *inter boundary* matrices, $B_{ij} \in \{0,1\}^{N_i^1 \times N_j^1}$. Regions m and n from partitions P_i^1 and P_j^1 respectively belong to the same cluster if $B_{ij}(m,n) = 0$.

Then, a co-clustering between nodes from a collection of hierarchies is defined by a binary matrix, the *boundary matrix*, $B \in \{0,1\}^{N \times N}$ where $N = \sum_i N_i^1$. It encodes the intra and inter boundary information between leaves of the M images in the collection.

$$B = \begin{bmatrix} B_{11} & ... & B_{1M} \\ \vdots & \ddots & \vdots \\ B_{M1} & ... & B_{MM} \end{bmatrix} \tag{3.8}$$

Note that B only encodes the information of the leaves. The hierarchical information is introduced in the optimization process through the intra constraints (Section 3.5).

In practice, not all the variables represented in this matrix are usefull, as boundaries between non adjacent leave regions are not considered in the process. Thus, in contrast to previous partition-based approaches in which the number of constraints was bounded by $O(n^2)$ ([69], [68]), our maximum number of intra constrains is proportional to n.

Our objective is to find the optimal boundary configuration that defines a collection of partitions $\{\pi_1^*, \pi_2^*, ..., \pi_M^*\}$ using nodes from hierarchies that are put in correspondace to form clusters. As proposed in [80], the co-clustering can be stated as an optimization problem. To compact notation, let us define $B_{i,j}(m,n) = b_{m,n}$:

$$\begin{aligned} &\min_{B}\ tr(QB) \\ &s.t.\ \ b_{m,n} \in \{0,1\}\ \forall m,n\ \ b_{m,m} = 0 \end{aligned} \tag{3.9}$$

where Q is a complex-valued Hermitian affinity matrix that measures the co-clustering quality.

Optimization Constraints

As commented in Section 3.2, we constrain the optimization process using the information in the hierarchy to avoid the inconsistencies of previous approaches. Previous co-clustering techniques ([69], [68]) use constraints that rely on the triangular equation to this purpose. This is, for each three-clique of adjacent regions, the labelling of these three regions to a single or to multiple clusters should be consistent. The main drawback of this approach is that label inconsistencies are only avoided in a reduced neighbourhood of each region. This information is expected to be propagated using the region adjacency, but inconsistencies are not specifically avoided out of this neighbourhood.

In this work, as we perform co-clustering between hierarchies, we exploit the tree information to both encourage semantic fusions between regions and to reduce the number of constraints involved in the optimization.

Intra Constraints

Each hierarchy H_i contributes in two aspects to the optimization process. First, it defines the mergings between regions of its leave partition P_i^1 to form clusters. Second, it also includes the order in which these regions should be merged to represent each node of the tree. Note that this order is not conditioned by the merging sequence. These two contributions of the hierarchy information lead to a large number of constraints among the regions forming the subtree below a given node. Nevertheless, in this work, all these original constraints have been encoded with only two coupled constraints per node.

First, for a given parent node and in order to merge its two siblings, all the leaves that form the boundaries between these two siblings should be merged. This is imposed by:

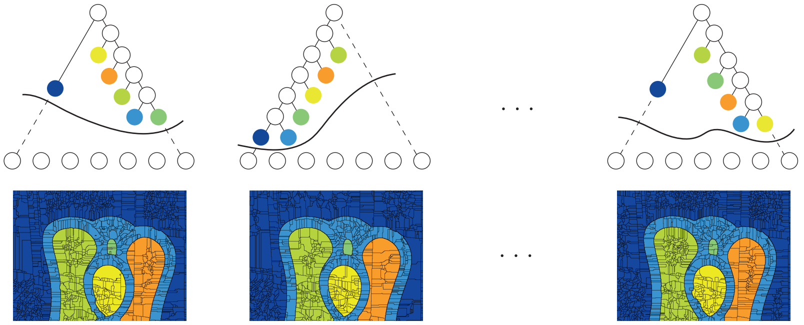


Figure 3.6: Co-clustering of hierarchies from a collection of images. First row: nodes selected from the tree to create partitions. Second row: clusters created with unions of leaves describing tree nodes. Lines represent the cut in the tree producing the optimal partition.

$$\sum_{n \neq l}^{m,n} b_{m,n} = (N_c - 1) b_{m,l} \tag{3.10}$$

where N_c is the total number of common region boundaries from the leave partition that represents the union of both siblings, m is a region from the first sibling and n, l are regions from the second sibling. This condition imposes that all the variables representing boundaries between two siblings should have the same value.

Second, for a given parent node and in order to merge its two siblings, the leaves that form their respective subtrees must also be merged:

$$\sum^{n,l} b_{n,l} + \sum^{n',l'} b_{n',l'} \leq N_m b_{m,o} \tag{3.11}$$

where N_m is the total number of inner region boundaries from the leaves partition of both siblings, m is a region from the first sibling, o is a region from the second sibling and n, l and n', l' are inner regions from the first and second sibling respectively. This condition imposes that for a given node, a variable representing a boundary between two siblings can only impose a merging if all the leaves associated with the node are merged.

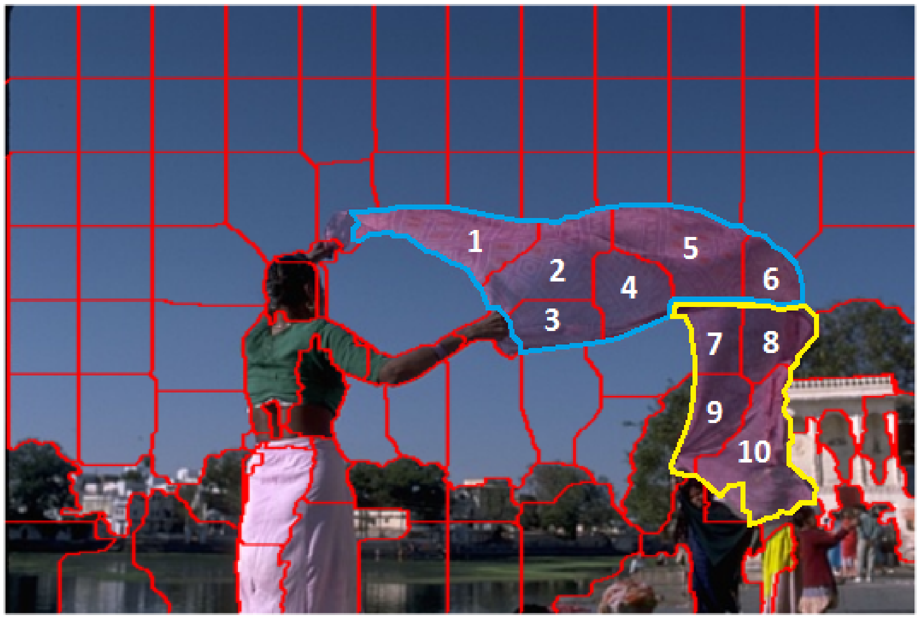


Figure 3.7: Example that illustrates the conditions imposed by Equation 3.10 and Equation 3.11 over boundaries of a partition given two siblings.

Note that Equation 3.10 guarantes that all boundaries between two siblings are either active or non active at the same time. Therefore, the second constraint 3.11, coupled with the first one, ensures that the optimization process propagates the second condition to all the node boundaries.

An example that illustrates the conditions imposed by these constraints over regions from a given partition is presented in Figure 3.7. Let us consider a leaves partition (P^1) defined by the red boundaries presented in the image. Moreover, we consider two siblings that are nodes of the hierarchy built on P^1 marked with yellow and blue.

Equation 3.10 imposes that all boundaries of the leaves partition that define the division between two siblings have the same value in order to create a partition avoiding inconsistencies. In this example, the division between both siblings is defined by regions 4, 5, 6, 7, 8. In terms of adjacency between regions from different siblings, region 7 is adjacent to regions 4 and 5 whereas region 8 is adjacent to regions 5. Thus, the boundaries involved in this constraint are $b_{4,7}$, $b_{5,7}$ and $b_{6,8}$. Using Equation 3.10:

$$b_{4,7} + b_{5,7} = 2b_{6,8} \tag{3.12}$$

Note that, if $b_{m,n}$ are binary (active or non active), this equation is only satisfied when all the boundaries have the same value. Furthermore, they can be permuted without changing the result.

On the other hand, Equation 3.11 imposes that regions that define the separation between both siblings (4, 5, 6, 7, 8) can only be merged if all the inner regions belonging to the siblings are merged. In other words, subtrees associated with these siblings have been previously merged. Using Equation 3.11:

$$(b_{1,2}+b_{2,3}+b_{2,4}+b_{2,5}+b_{3,4}+b_{4,5}+b_{5,6})+(b_{7,8}+b_{7,9}+b_{8,9}+b_{8,10}+b_{9,10}) \leq 12b_{6,8} \tag{3.13}$$

In this case, $b_{6,8}$ can be substituted by any other boundary that define the separation between both siblings ($b_{4,7}$ or $b_{5,7}$). In this example we use the same boundary at the rigth part of Equation and Equation to show that they are coupled. It can be observed that when $b_{6,8}$ is non active all the inner boundaries of both siblings should be merged. Otherwise, this equation is not imposing any constraint.

As all the inner boundaries should be merged before merging any boundary that defines the saparation between two siblings () and all the boundaries that form this separation should be active or non active at the same time, the hierarchical information of the tree is introduced in the optimization process.

Inter Constraints

These constraints control the correspondances between nodes from different hierarchies. In this case, as we do not have any hierarchical relation for these nodes, the triangular equation is used to create the inter constraints:

$$b_{m,n} \leq b_{m,l} + b_{l,n} \quad \forall e_{m,n}, e_{m,l}, e_{l,n} \in G \tag{3.14}$$

where $e_{m,n}$ is the edge between leaves m and n of the region adjacency graph G computed from the leave partitions.

Similarities

Our co-clustering technique exploits the randomness of those partition contours that do not belong to semantic objects. In this process, the computation of region similarities is crucial to correctly match regions from different partitions. Two types of similarities are computed: *intra similarities* (between leaves from the same hierarchy) and *inter similarities* (between leaves from different hierarchies).

Previous clustering works in segmentation and cosegmentation frameworks ([68], [88]), use the color information to compute intra similarities. We propose to compute these similarities as:

$$W_{ii}(m,n) = \alpha_{m,n}\,(1 - e^{1-d_B(m,n)}) \tag{3.15}$$

where $\alpha_{m,n}$ is the length of the common boundary between leaves m, n and $d_B(m,n)$ is the Bhathacharyya distance [46] of the 8-bin RGB color histograms of regions m, n.

Inter similarities are used to create clusters combining nodes from different hierarchies. In [68], inter similarities are computed using a HOG-based descriptor. Although this gradient information may be enough in some cases, additional descriptors able to robustly match region contours are required. However, only those descriptors that can be efficiently computed should be taken into account.

We propose to combine three simple yet effective descriptors, which are computed over the contour elements of each partition. These descriptors are combined in a feature vector associated with each contour element, what allows us to keep the additivity property that is the key to formulate our problem as a linear optimization.

Inter image similarity between regions m and n from partitions P_i^1 and P_j^1 respectively should be proportional to their joint probability $p(m,n)$. We considere three types of information to model differences between regions from different partitions: changes of *color/illumination*, *deformations* and small changes of *position*. In terms of probability, we consider these processes to be independent:

$$p(m,n) = p_C(m,n)p_D(m,n)p_P(m,n) \tag{3.16}$$

The *color information* is obtained from a histogram of pixels in a neighborhood of the boundary elements. As each contour element can be represented by two pixels in the image (one pixel from the analyzed region and another from the adjacent region), two histograms are computed in the direction of the normal. Each histogram is computed over a window centered on the pixel of the region which is closer to the boundary in that direction and they are averaged. To handle possible *deformations*, shape information around each contour element is captured with a HOG descriptor. In our work, HOGs are computed using the gPb [90] information. Finally, *position* changes are captured with the Euclidean distance between elements.

Similarity between contour elements is computed as:

$$W_{ij}(u,v) = e^{(f_i^u - f_j^v)^T \Sigma^{-1} (f_i^u - f_j^v)} \tag{3.17}$$

where f_i^u is the feature vector of contour element u that belongs to P_i^1. This vector is formed as the concatenation of the three types of descriptors previously described. We allow matchings between contour elements that are closer than 20 pixels. Otherwise, $W_{ij}(u,v) = 0$.

Once both inter and intra similarities are computed for all contour elements of the leave partitions, a similarity matrix between regions is built for each pair of hierarchies.

$$Q_{ij} = {O_i}^H W_{ij} O_j \tag{3.18}$$

where O_i, O_j are complex matrices that describe the edges orientations (computed using the gPb [90] information) of all contour elements from partitions P_i^1 and P_j^1, and W_{ij} encodes the inter similarities between these elements.

Finally, the similarity matrix Q that measures the quality of the co-clustering is built using the information of all the inter and intra similarity matrices as in Equation 3.8.

Optimization process

Using the similarity matrix and the constraints presented in this section, the optimization process of Equation 3.9 can be formulated as:

$$\begin{aligned} &\min_B \sum_{m,n} q_{m,n} b_{m,n} \\ &s.t. \quad b_{m,n} \in \{0,1\} \quad b_{m,n} = b_{n,m} \quad \forall n,m \\ &\sum_{n \neq l}^{m,n} b_{m,n} = (N_c - 1) b_{m,l}\ , \ \sum^{n,l} b_{n,l} \leq N_m b_{m,o} \ \forall \mathfrak{p} \in \{H_i\} \\ &b_{m,n} \leq b_{m,l} + b_{l,n} \quad \forall e_{m,n}, e_{m,l}, e_{l,n} \in G \end{aligned} \tag{3.19}$$

where $\mathfrak{p}$ represents any parent node in the collection of hierarchies. The result of this optimization is a binary matrix B^* that describes the collection of optimal partitions $\{\pi_1^*, \pi_2^*, ..., \pi_M^*\}$. Thus, nodes from the collection of hierarchies $\{H_i\}$ have been clustered with the same label and semantic contours are preserved through the collection.

Multi-resolution

In previous sections, a technique that creates a collection of partitions clustering nodes from their associated hierarchies has been presented. This technique creates a single partition per hierarchy, coherently describing semantic contours of the original collection of images at a given resolution.

Nowadays, it is commonly accepted that multiresolution region-based descriptions provide a rich framework for image and video analysis [91], [1]. In this section, we extend the previous hierarchical co-clustering to a multiresolution framework as it is illustrated in Figure 3.8.

This is, for each hierarchy involved in the optimization process (H_i), we cluster nodes to obtain N_r partitions, forming a new optimal hierarchy ($\mathcal{H}_i^*$) that represents the image at N_r different resolution levels ($\mathcal{H}_i^* = \{\pi_i^{1*}, \pi_i^{2*}, ..., \pi_i^{N_r*}\}$). Moreover, the collection of optimal partitions generated for each resolution should keep their inter correspondances.

Let us consider a clustering problem as presented in Equation 3.19, from which a boundary matrix B is obtained for each generated partition. The number of active boundaries in B has a direct relation with the resolution of the resulting partitions and, in particular, that of intra boundaries. When imposing in the optimization process a low (high) number of intra contours, coarser (finer) resolutions are obtained. We have observed that parameterizing the search in the solution space with respect to the number of intra contours allows the algorithm to produce a set of well distributed resolutions. Formally, given a collection of hierarchies ($\{H_i\}$), their nodes are clustered to form a collection of partitions of a given resolution ($\{\pi_1^{r*}, \pi_2^{r*}, ..., \pi_M^{r*}\}$) by constraining the optimization problem presented in Equation 3.19 with an additional condition for each hierarchy:

$$(T_r - \beta) \cdot N_b \leq \sum^{m,n} b_{m,n} \leq T_r \cdot N_b \qquad (3.20)$$

where N_b is the number of active boundaries to encode the leave contours, T_r is the maximum fraction of these contours to describe the r-th coarse

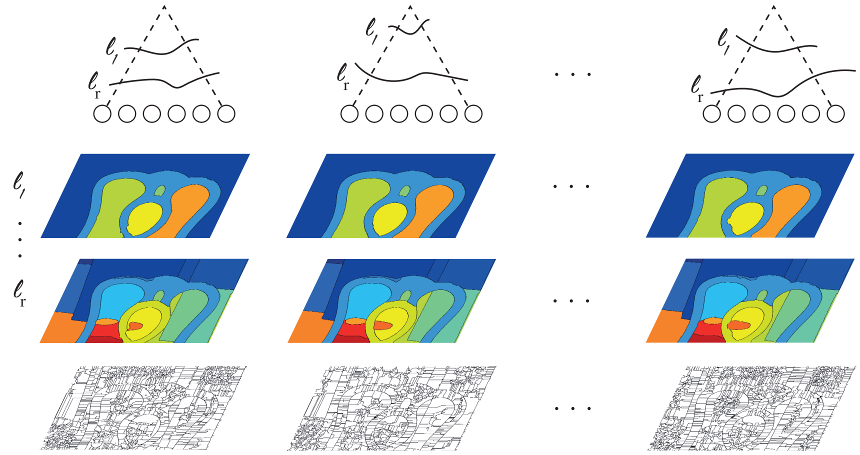


Figure 3.8: Multiresolution hierarchy co-clustering of an image collection. First row: different cuts of each tree associated with different resolutions. Second and third row: optimal partitions generated by the previous hierarchy cuts. Fourth row: leave partitions

level and β represents the maximum difference in number of boundaries between consecutive levels.

This approach allows two search strategies. When $\beta = T_r - T_{r-1}$, a complete set of consecutive, equal sized subspaces is analyzed. On the contrary, when $\beta < T_r - T_{r-1}$ a coarser sampling of the solution space is performed.

3.6 Multi-resolution video co-clustering

In this section we propose to particularize the technique presented in Section 3.5 to a multiresolution video segmentation algorithm for sequences with small variations. Note that the previous co-clustering technique could be adapted to a 3D volume approach, as in [1]. However, such an approach would require high memory resources (Section 3.1). Thus, we adopt an iterative approach as in [83] (Figure 3.13).

We propose to propagate clusters along sequences at various resolutions, taking into account the information in previous processed frames. As in [2], we use pieces of video and propagate the result through the sequence. In our case, we propagate semantic contours using information from different granularities in the optimization process. This is a forward-only online processing, and the results are good and efficient in terms of time and complexity.

In particular, for each image (I_i) in the sequence and for a given resolution (r), we perform a joint hierarchical co-clustering with the clustering result of the two previous frames at two different scales: the resolution level under analysis and the leave partition scale (see Figure 3.13). Precisely, we construct the boundary matrix B using the optimal partition in $i-2$ at level r (π_{i-2}^{r*}) and the leave partitions in $i-1$ and i (P_{i-1}^{1} and P_{i}^{1}).

Moreover, the optimization problem in 3.19 and 3.20 is further constrained imposing two additional conditions. In order not to modify previous co-clustering results, regions in π_{i-2}^{r*} must not be merged

$$\sum^{m,n} b_{m,n} = N_v \tag{3.21}$$

where $b_{m,n}$ are intra or inter boundary variables from π_{i-2}^{r*} and P_{i-1}^{1} that encode the boundaries between clusters of π_{i-2}^{r*} and π_{i-1}^{r*}, and N_v is the cardinality of these variables.

In turn, regions in P_{i-1}^{1} must be merged to form π_{i-1}^{r*} and inter correspondances between clusters must be kept:

$$\sum^{m,n} b_{m,n} = 0 \tag{3.22}$$

where $b_{m,n}$ are intra or inter boundary variables from π_{i-2}^{r*} and P_{i-1}^{1} that encode the unions of inter and intra clusters of π_{i-2}^{r*} and π_{i-1}^{r*}.

Leave partitions (P_{i-1}^{1} and P_{i}^{1}) are used to allow computing fine boundary similarities, whereas boundaries from π_{i-2}^{r*} and π_{i-1}^{r*} are included to enforce previous semantic contours. With this iterative process, clusters are robustly propagated through hierarchies in an efficient manner.

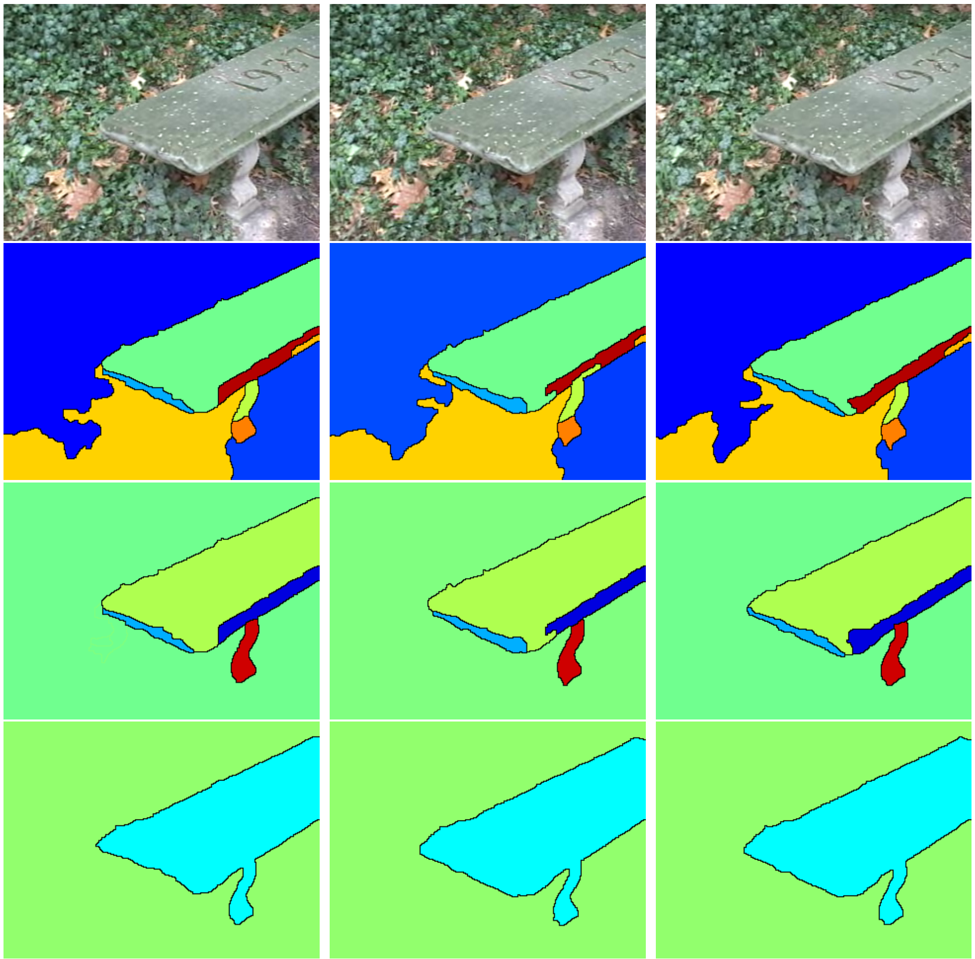


Figure 3.9: Sequence *bench*. First row: original images. Second-fourth rows: partitions at three different levels of resolution obtained clustering hierarchy nodes using our technique.

3.7 Evaluation tools

Maximum partition quality

In order to assess the quality of objects represented by unions of regions, object masks from pixel-level annotated databases are used. The Jaccard index [92] between the mask and a union of regions of the partition, is a common measure to asses the segmentation quality. For any two unions of

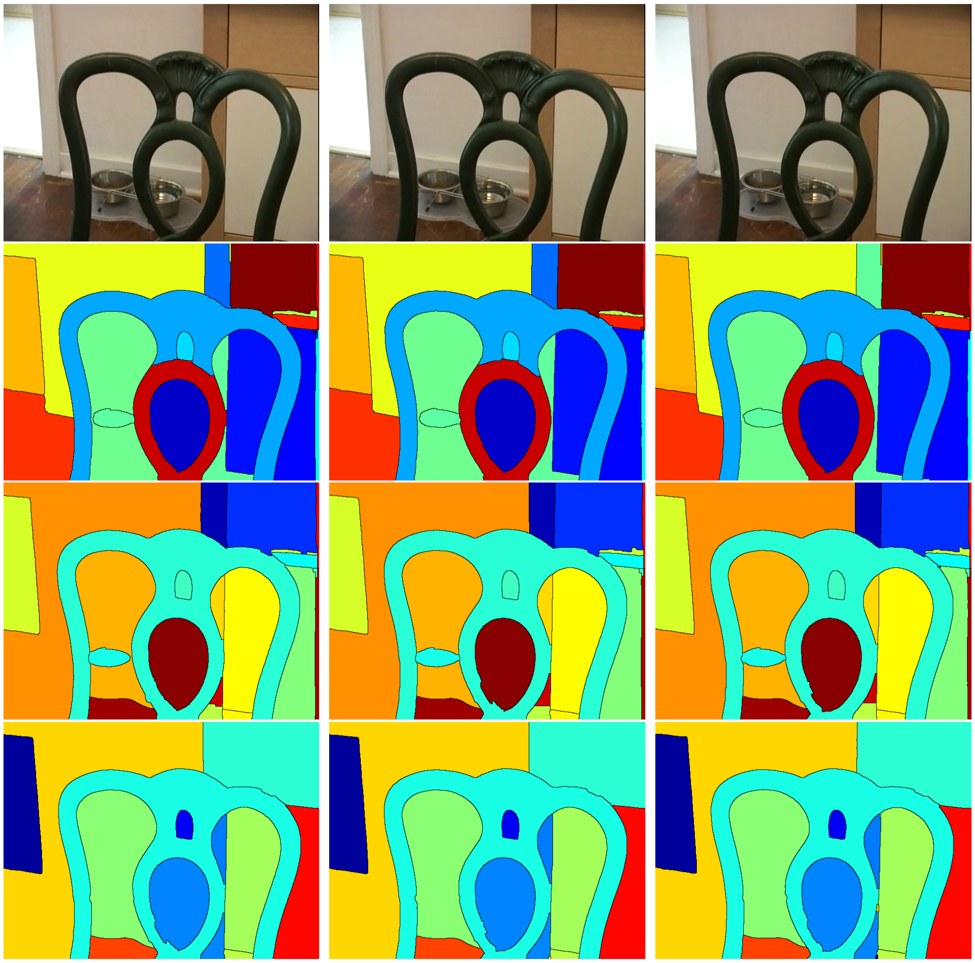


Figure 3.10: Sequence *chair1*. First row: original images. Second-fourth rows: partitions at three different levels of resolution obtained clustering hierarchy nodes using our technique.

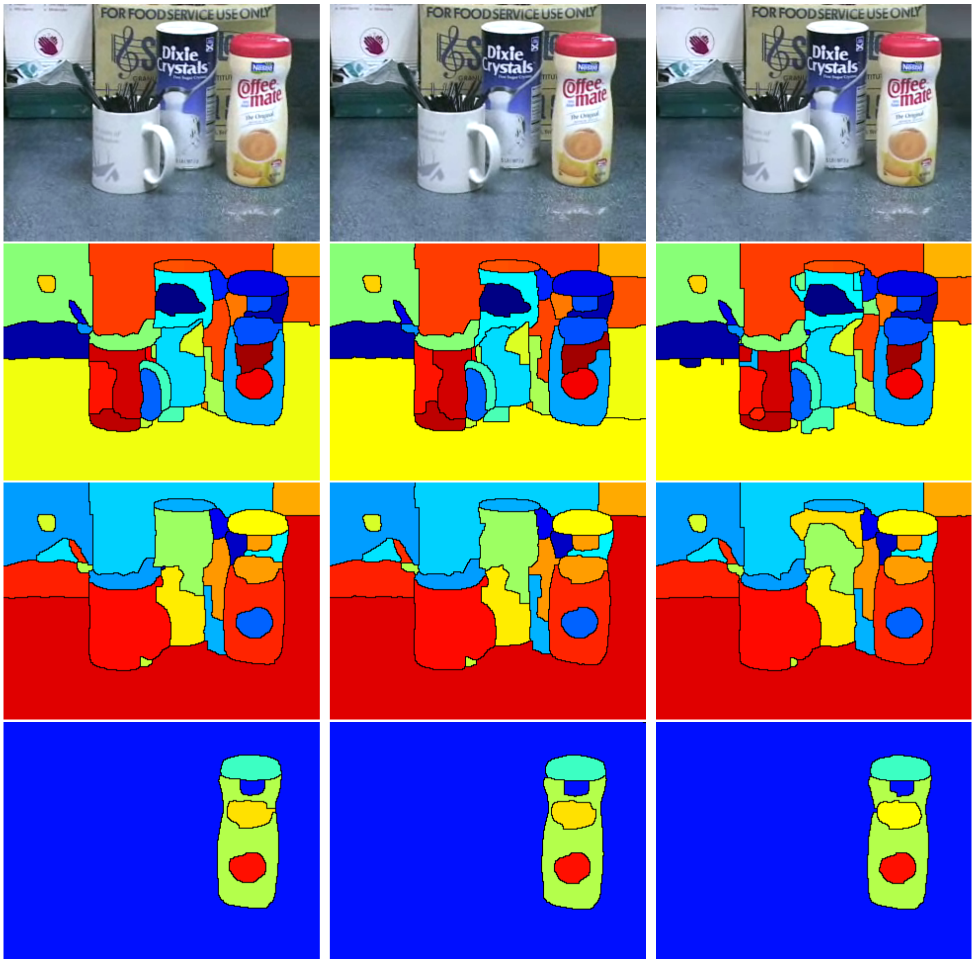


Figure 3.11: Sequence *coffee_stuff*. First row: original images. Second-fourth rows: partitions at three different levels of resolution obtained clustering hierarchy nodes using our technique.

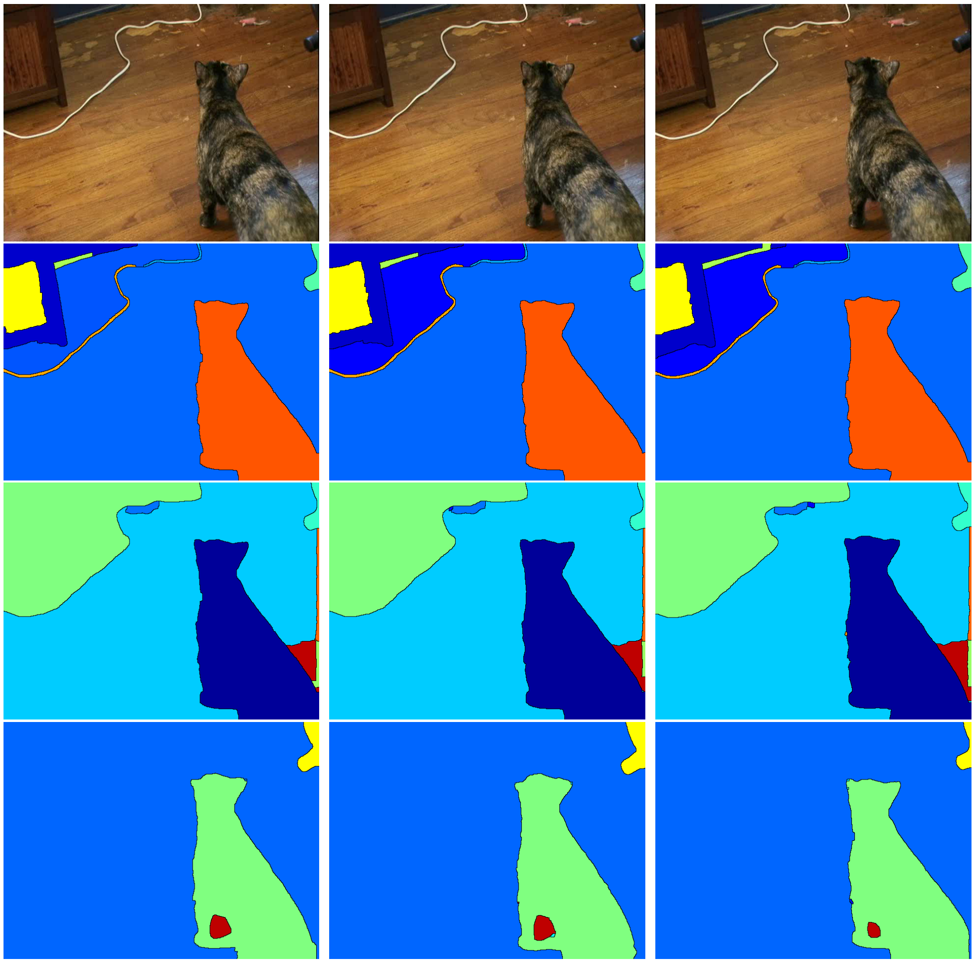


Figure 3.12: Sequence *zoe1*. First row: original images. Second-fourth rows: partitions at three different levels of resolution obtained clustering hierarchy nodes using our technique.

regions A and B, the Jaccard index is computed as:

$$J(A,B)=\frac{|A\cap B|}{|A\cup B|} \tag{3.23}$$

Note that, the quality of any co-clustering is bounded by the maximum quality of the initial partition of the reference image. Then, to assess the co-clustering as independently as possible of the segmentation technique, a comparison between the Jaccard obtained by the co-clustering and the upper bound imposed by the initial partition is required.

Formally, let P be a partition formed by a group of regions $P=\{R^1,R^2,...,R^n\}$ and M a binary mask with a region R_o that represents the annotated object. Let $p\in\{0,1\}^n$ be a binary vector such that $p(i)=1$ if region R^i is considered as a part of the annotated object. Our approach to find the union of regions from P that maximizes the Jaccard index models the problem as a binary search:

$$\begin{aligned}\max_x \frac{c\cdot x+d}{g\cdot x+h} \quad s.t \quad & x\in\{0,1\}^n\\ & g\cdot x+h\geq 0\end{aligned} \tag{3.24}$$

where c, $g\in\mathbb{R}^n$ and d, $h\in\mathbb{R}$.

Let us consider the intersection between the annotated object and a certain union of regions $U=\cup_{r=1}^{n_u}R^r$ from P. This intersection can be decomposed as:

$$|A\cap U|=\sum_{r=1}^{n_u}|A\cap R^r|=\sum_{r=1}^{n}|A\cap R^r|\cdot p(r) \tag{3.25}$$

Thus, $c(r)=|A\cap R^r|$ for all the regions from P and $d=0$.

A similar expression can be obtained to decompose the union between both subsets of regions. If we define $|A|$ as the number of pixels that form the annotated object and:

$$|A\cup U_j|=|A|+\sum_{r=1}^{n_u}|\tilde{R_j}^r|=|A|+\sum_{r=1}^{n_j}|\tilde{R_j}^r|\cdot p_j(r) \tag{3.26}$$

where $|\tilde{R_j}^r|=|\bar{A}\cap R_j^r|$ is the number of pixels from R_j^r that are not included in R_o, then $g(r)=|\tilde{R_j}^r|$ and $h=|A|$.

This type of optimization problem is referred to as a Linear Fractional Combinatorial Optimization (LFCO) problem in [93], which also presents an efficient way to solve it. An analogous approach is used in [94] for computing the F-measure of the region boundaries.

Maximum hierarchy quality

Let us consider combinations of nodes from a certain hierarchy as possible representations of the annotated object. Then, the maximum quality that can be obtained in terms of the Jaccard index is given by the partition formed by the hierarchy leaves. This upper bound can be computed using the method presented in Section 3.7.

As presented in [95], not only the quality of the segmentation (*Consistency*) is important. Moreover, the number of regions that represent the object (*Efficiency*) should also be taken into account. In this work, we propose to reduce the number of clusters that represent the object taking advantage of region mergings at higher nodes of the tree.

In order to assess the performance of this process, we present an efficient algorithm to obtain the highest segmentation quality that can be achieved using C nodes of a hierarchy to represent the annotated object.

Let H be a hierarchy formed by a group of nodes $H = \{N^1, N^2, ..., N^{n_N}\}$ and M a binary mask with a region R_o that represents the annotated object. Let $h \in \{0,1\}^{n_N}$ be a binary vector such that $h(i) = 1$ if node N^i is considered as a part of the annotated object. The search of the maximum segmentation quality that can be achieved with a subset $U = \{U^1, U^2, ..., U^C\} \subset H$ of C nodes, can be stated as an LFCO problem as presented in Equation 3.24. As the quality of the segmentation is assessed using the Jaccard index, the parameters that model the problem are obtained as shown in equations 3.25 and 3.26. However, the relations between nodes from the hierarchy should be modeled using some additional constraints.

In particular, once a node is included in U, all its descendents should be discarded. This constraint is satisfied appliying the condition $h(k) + h(s_{ki}) \leq 1$ recursively between parents and their children in the hierarchy, where k is a given node and s_{ki} is its i_{th} child. The optimization problem is finally writen as:

$$\max_{h_r} \frac{\sum_{k=1}^{n_N} |A \cap N_r^k| \cdot h_r(k)}{\sum_{k=1}^{n_N} |\tilde{N_r}^k| \cdot h_r(k) + |A|} \quad s.t \;\; h_r \in \{0,1\}^{n_N}$$
$$h_r(k) + h_r(s_{ki}) \leq 1 \qquad \sum_{k=1}^{n_N} h_r(k) = C \tag{3.27}$$

where $|A|$ is the area of the annotated object, $|A \cap N_r^k|$ is the intersection between the annotated object and node k from the hierarchy H_r and $|\tilde{N_r}^k|$ is the area of node k which does not intersect with the annotated object. This problem can be efficiently solved as presented in Section 3.7.

3.8 Experimental Evaluation

In this section, we present both qualitative and quantitative evaluations of our multiresolution hierarchical co-clustering (MRHC). As our technique aims at segmenting sequences with small variations, we use the Video Occlusion/Object Boundary Detection Dataset [95] for evaluation and comparison with state-of-the-art methods in the fields of video segmentation ([1], [2], [82]) and co-segmentation ([96], [85]). Comparisons have been made using the implementations from respective authors. In order to asses the contribution of the multi-resolution framework (Section 3.6), we also evaluate the performance of our algorithm at a single level with the best overall results (OURS-SL). This is the result of selecting the level of the resulting multiresolution that obtains the best results. Moreover, based on the baseline in [97], we consider a system that propagates labels from regions obtained with [9] along the sequence using the optical flow of [70] (UCM-P). This technique uses the optical flow to propagate regions between consecutive frames and selects the set of regions of the new frame that better represent the propagated regions. A random hierarchy created from the leave partitions of [9] is also used as baseline technique.

The dataset includes 30 short sequences (42 objects) with indoor and outdoor scenes, noise and compression artifacts, unconstrained handheld camera motions and moving objects. For each sequence, the annotation of a single frame is provided as ground truth for segmentation assessment (Section 3.8). To assess temporal consistency (Section 3.8), we have manually

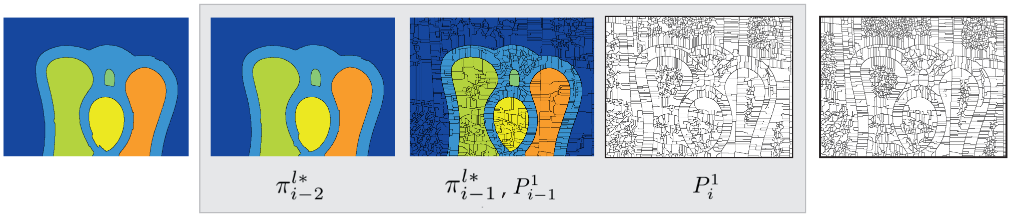


Figure 3.13: Iterative algorithm to propagate semantic information through a video. As it can be seen, information of different coarse levels (π^{l*}_{i-2}, π^{l*}_{i-1}, P^1_{i-1}) is used to compute the optimal current frame partition without modifying the previous results.

annotated the remaining frames by merging regions from the leave partitions of [9]. The evaluation is performed using two types of measures. First, we use the measures presented in [97]: boundary precision-recall (BPR) from [9] and a volume precision-recall metric (VPR). Second, as in ([95], [68]), we use *Consistency* as the Jaccard index computed between a set of regions of a partition and the ground-truth and *Efficiency* as the minimum number of regions requested to obtain a given consistency.

Formally, the consistency (C) associated with an efficiency (E) of N_E regions is computed as:

$$C = \max \frac{|R \cap G|}{|R \cup G|} \tag{3.28}$$

where $R = \cup_i^{N_E} R_i$, R_i are regions from the reference frame partition at a given coarse level and G is the annotated reference object.

In order to qualitatively assess our technique and to explore its limitations, we also analyze a subset of sequences from the SegTrack v2 Dataset [98], some of them containing strong deformations and rapid variations. In all the experiments, hierarchies have been obtained using [9] and 30 resolution levels have been created per sequence ranging between $[40\%, 10\%]$ the number of leaf contours ($\beta = 0.1$).

Segmentation assessment

In this experiment, we assess the segmentation quality of a given frame. The set of optimal partitions of this frame for all the resolution levels is considered. Then, for each efficiency value, the maximum consistency over this set of levels is selected; that is, fixing the number of regions, we select through the various resolutions the best Jacard object representation. Moreover, the BPR curve is considered to assess the quality of segmentation boundaries (Figure 3.14).

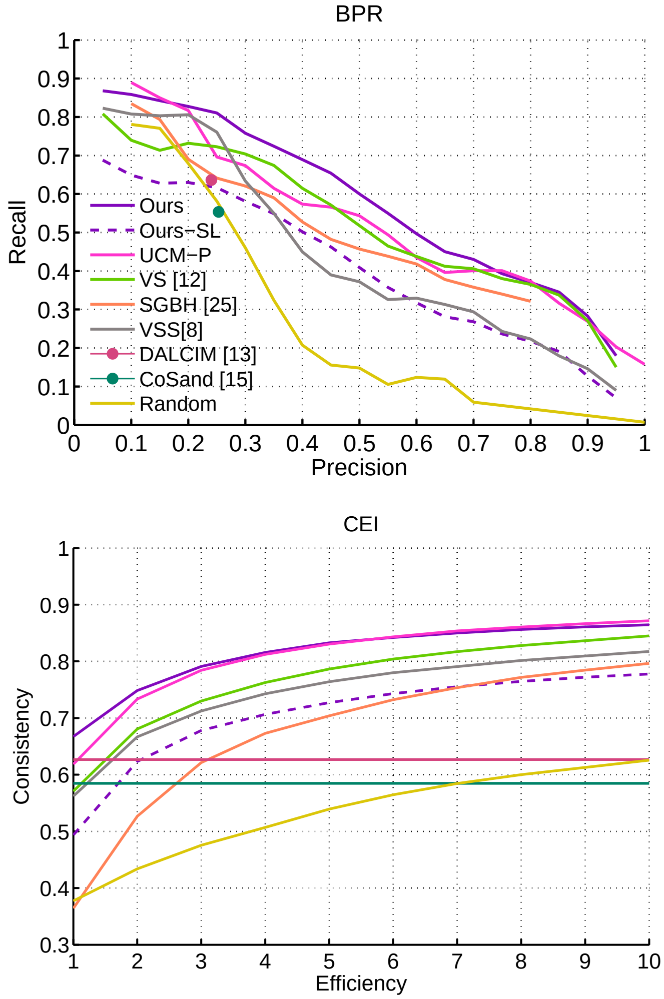


Figure 3.14: Comparison between different methods evaluating their boundary precision-recall (BPR) and their consistency for different levels of efficiency both over a single image (CEI).

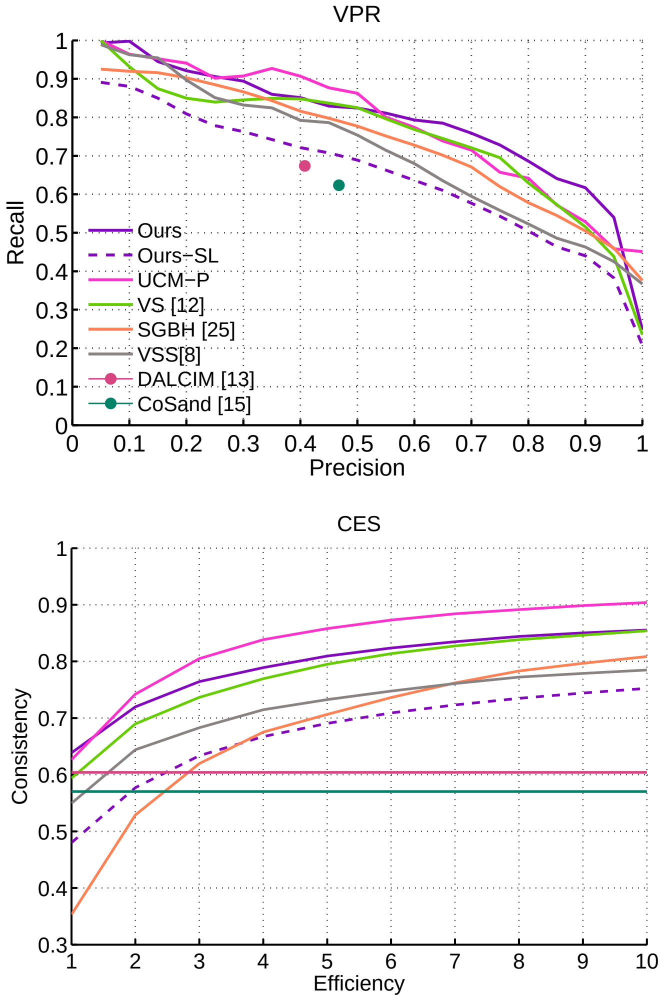


Figure 3.15: Comparison between different methods evaluating their volume precision-recall (VPR) and their consistency for different levels of efficiency both over a sequence (CES).

Co-segmentation results have been obtained fixing the number of clusters with respect to the number of objects in the scene, as proposed by the authors ([96], [85]). We report the best results for up to a given number of clusters, since consistency does not improve when increasing the number of clusters. These algorithms are competitive when the object is represented with one region. Still, our technique obtains better consistency for all efficiency levels. due to hierarchies and similarities among frames to describe objects.

Regarding video segmentation algorithms, our technique outperforms the three assessed state-of-the-art methods ([1], [2], [82]). In [2], colour similarities are used to propagate supervoxels information. In contrast, our description of contours using colour, texture and distance measures, obtains better segmentation accuracy and BPR for all precision levels. Although the optical flow used in [1] is a powerfull descriptor, it is not enough to accurately segment objects in this type of sequences, specially with a low number of regions. As it can be observed in Figure 3.14, in terms of boundaries, their recall is close to our results for large precision values. However, in terms of object area, regions selected by our algorithm represent the object with higher accuray.

Table 3.1 shows the number of objects from the database in which our algorithm obtains better/worse consistency for more than 50% of the efficiency levels shown in Figure 3.14.

Temporal coherence assessment

In this section, we extend the previous "efficiency versus consistency" analysis to the temporal domain, in order to assess the stability of partitions along video sequences.

The sequence consistency of a label (temporal cluster) is computed averaging the consistency values obtained at each frame by the region associated to this label. Results of the best sequence consistency achieved for all the resolutions, using the number of labels represented by each efficiency level, are plotted in Figure 3.14. In order to complete the analysis, we also present the VPR curve as computed in [97].

As it can be observed, sequence consistency results are very similar to segmentation consistency ones (Figure 3.14). This stability shows that all methods correctly maintain the coherence of the partitions along the sequence. These results validate the iterative strategies used in [2] and in our approach (see Section 3.6). In both volume precision-recall and consistency-efficiency values, our method outperforms the analyzed state-of-the-art approaches and only the propagation method based on [97] obtains better volume recall for low precision values and better efficiency. This confirms the results that were reported in previous works ([97], [83]).

	Better		**Worse**		**Inconclusive**	
	Ref.	Seq.	Ref.	Seq.	Ref.	Seq.
[1]	76%	64%	19%	26%	5%	10%
[2]	88%	79%	7%	19%	5%	2%
[82]	81%	74%	16%	14%	3%	12%
[96]	88%	89%	12%	9%	0%	2%
[85]	90%	93%	10%	7%	0%	0%
OURS-SL	81%	89%	12%	5%	7%	6%
UCM-P	65%	34%	27%	62%	8%	4%

Table 3.1: Objects of the database for which our algorithm obtains better/worse image (Ref) or sequence (Seq) consistency at more than 50% efficiency levels. Otherwise, it is said to be inconclusive.

A more detailed comparison of the presented algorithms for the objects in the database can be found in Table 3.1.

Qualitative assessment

In this section, we present results on two sequences from the Segtrack v2 database [98] to qualitative evaluate our algorithm. This database allows analyzing the limits of our technique, since video objects in it may undergo strong deformations and rapid movements.

Figure 3.16 shows two images of the sequence *Parachute*. As it can be observed, the parachute is correctly segmented along the sequence at a given resolution. Moreover, its coloured stripes are coherently segmented through the sequence. In this sequence, although the parachute does not suffer strong deformations, the hierarchies associated with consecutive frames vary along the sequence. Thus, selecting nodes at different resolutions allows our method to correctly introduce the hierarchical information in the optimization process. Furthermore, as the object shape gradually changes, our method is able to coherently segment it at several resolution levels along the video.

Figure 3.17 shows two images of the sequence *Girl*. In this sequence, a girl runs and her shape undergoes strong deformations due to arm and leg rapid movements. Although the shape of the girl is correctly identified in both partitions as the union of a few regions (high consistency at medium

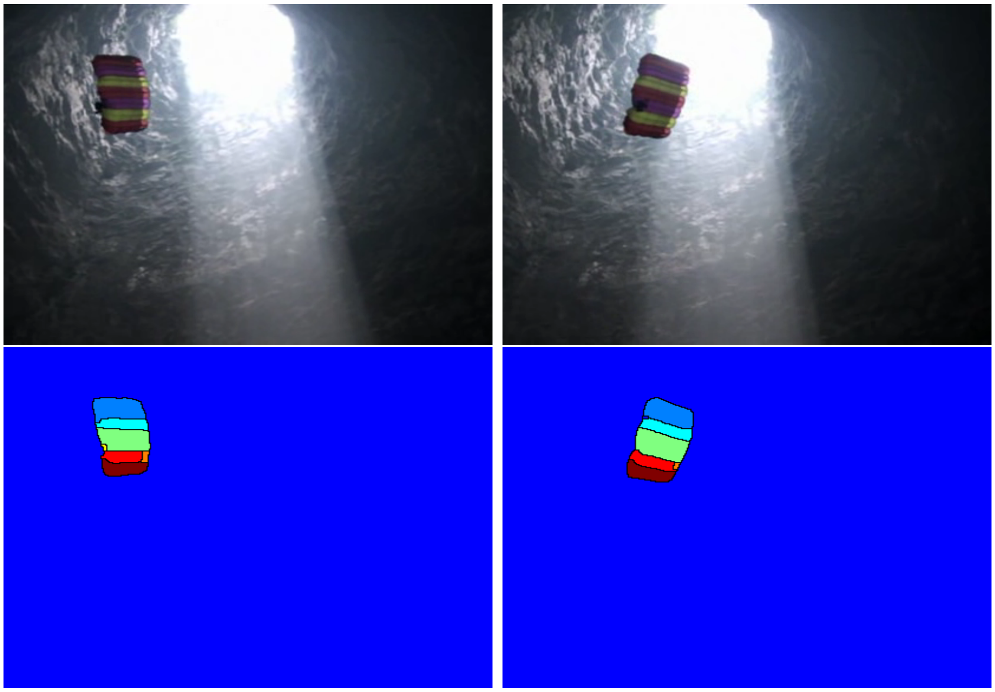


Figure 3.16: Qualitative evaluation of sequence *Parachute* from the SegTrack v2 database. First row: original images. Second row: result of our iterative video segmentation method.

efficiency for segmentation), not all its parts have been coherently matched (worse efficiency for temporal coherence). This is mainly caused by ptical flow errors and blurring in the sequence.

3.9 Conclusions

In this work, we have presented a co-clustering framework that creates a coherent region-based multiresolution representation of an image collection, by clustering nodes from a collection of independent hierarchies. The co-clustering problem is formulated as a QSAP problem. Inconsistencies commonly derived from such optimization problems are avoided modelling the problem through boundary variables and effectively using hierarchical constraints. This way, our method robustly creates inter and intra relations between regions from the image collection.

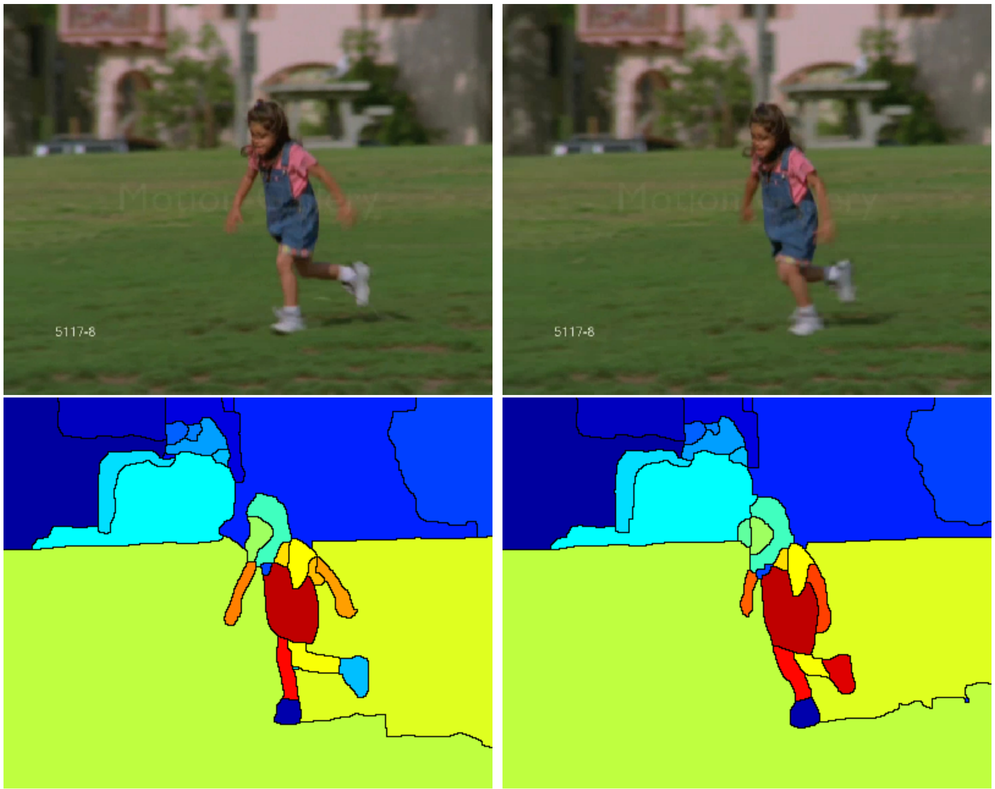


Figure 3.17: Qualitative evaluation of sequence *Girl* from the SegTrack v2 database. First row: original images. Second row: result of our iterative video segmentation method.

This co-clustering framework has been particularized to obtain a video segmentation technique that coherently segments scenes with small variations. We have adopted an iterative strategy that allows reducing the algorithm complexity and memory requirements, while achieving high temporal coherence. We have assessed the results over the Video Occlusion/Object Boundary Detection Datase against five SoA techniques and three baseline ones. In all cases, our technique outperforms the SoA methods in video segmentation and co-segmentation for this type of sequence in all range of efficiencies.

Chapter 4

Applications

4.1 Introduction

In this chapter, the proposed techniques for object and video segmentation presented in Chapters 2 and 3 respectively are used as tools to tackle problems in a context for which they were not initially thought.

While this Thesis was being developed, we had the opportunity of applying these algorithms in two different applications. First, our region-based particle filter presented in Chapter 2 was used to segment objects from sequences generated by a humanoid robot to perform 3D reconstruction extracting a set of different views of the object. In this application an accurate segmentation of the object being tracked was crucial for a robust reconstruction. This reconstruction is the first step for the interaction between a humanoid robot and objects in the scene.

In the second application, we explored the extension of our multi resolution video segmentation technique presented in Chapter 3 to the context of uncalibrated multiview segmentation. Our clustering technique, which originally was developed to analyze video sequences at different resolutions, was used to automatically propagate semantic information through a set of views of the same scene. As a result, semantic segmentation of objects may be improved in those views in which considering a single image is not enough to robustly perform this task.

The third application is focused on solving a rate-distortion optimization problem using the hierarchical co-clustering presented in Chapter 3. In this application, depth information of multiple views is jointly used to obtain an initial 3D segmentation. This segmentation is projected to a single view where a hierarchical segmentation is built. Then, a rate-distortion optimization procedure is applied to obtain the optimal segmentation for the views choosing nodes of the hierarchy.

For each application, we show how the proposed techniques presented in the context of object and video segmentation can be also usefull to tackle other problems in different scenarios.

4.2 3D Shape Reconstruction from a Humanoid Generated Video Sequence

Introduction

Most humanoid robots rely on vision systems in order to perceive the environment and resemble human capabilities. In particular, monocular vision is preferred for small-sized humanoids that are certainly constrained to be equipped with lightweight, low-cost and low-energy consumption devices.

For these robots, there is a tradeoff between a suitable camera and the quality of the images acquired during the biped march, as the stepping impacts cause jerky camera movements which generate continuos blurring along the related video sequence.

In the context of 3D object reconstruction, analyzing a monocular video sequence acquired by a humanoid robot represents a difficult task which involves solving for camera localization as well as extracting meaningful image features under challenging motion conditions. This application investigates the feasibility to estimate, in a multi-view fashion, the 3D geometry of an interest object from the video frames generated along the march of a humanoid. In order to capture multiple views, the robot performs a circular trajectory generated through a locomotion control that corrects the positions and orientations of the robot in accordance with vectors lying on a virtual circle of known radius. For object segmentation, a strategy has been developed that aims at selecting a suitable set of video frames for robustly reconstructing the 3D shape of the object. In a first stage, blurred images are eliminated from the sequence as well as those frames where parts of the object appear outside the image limits. From this subset, object segmentation is performed using a region-based particle filter approach, from which a consistency score is assigned to each frame. The video frames with the highest scores that also observe a uniform distribution of the sampled object views are finally selected for 3D shape recovery. The process is illustrated in Figure 4.1, where the final selected video frames are shown as camera poses surrounding an object of interest. In this sense, the main contribution of this application is a method that is capable of analyzing a video sequence generated by a humanoid robot for the purposes of 3D object reconstruction from multiple views relying on the segmentations obtained with the region-based particle filter.

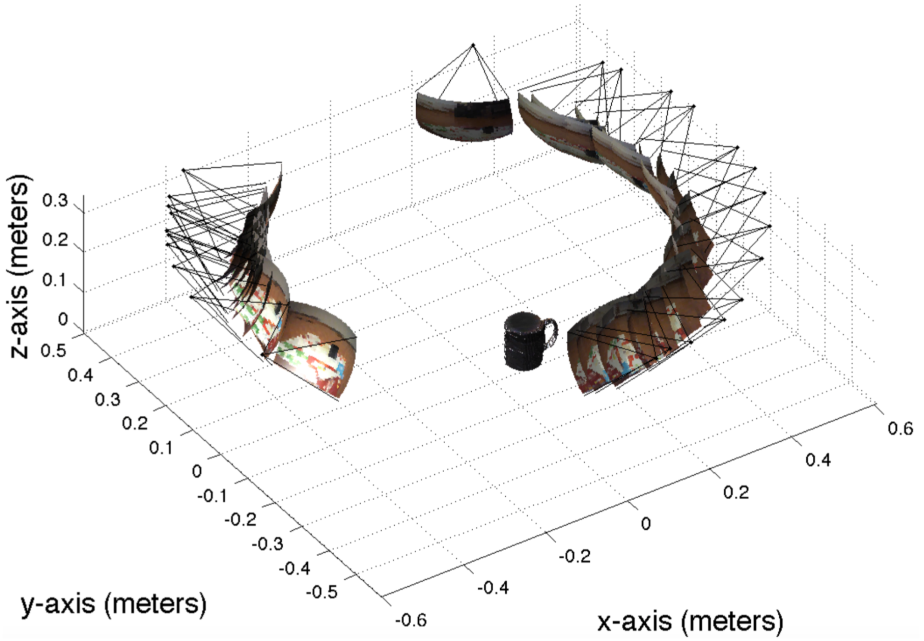


Figure 4.1: The proposed strategy. A humanoid robot records a video sequence that samples multiple views of the shape of an interest object. A visual-based locomotion control uses the monocular localization of the robot to correct its stepping and performs the required trajectory. An analysis of the recorded sequence is applied in order to determine the suitability of each frame for the purposes of contour extraction. Finally, from the selected frames, a particle filter-based object segmentation process is coupled with a space carving algorithm for estimating the geometry of the object. The figure shows the 25 camera poses of the selected video frames and the estimated 3D shape of the object.

Framework

Once the video has been recorded by the robot, the object must be segmented in order to create a 3D model. In this work, we adapt the region-based particle filter presented in Chapter 2 to extract the 2D shape of the object from partitions (See Figure 5(b)) associated with a set of multiple views of the object. Only images containing the entire object without blur should be processed and a subset of these images is finally selected to robustly reconstruct the object in accordance with their final segmentation quality. A diagram of the proposed framework is presented in Figure 4.2.

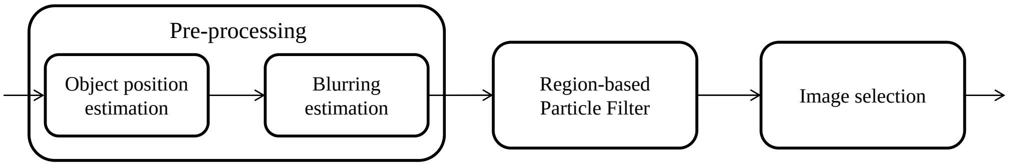
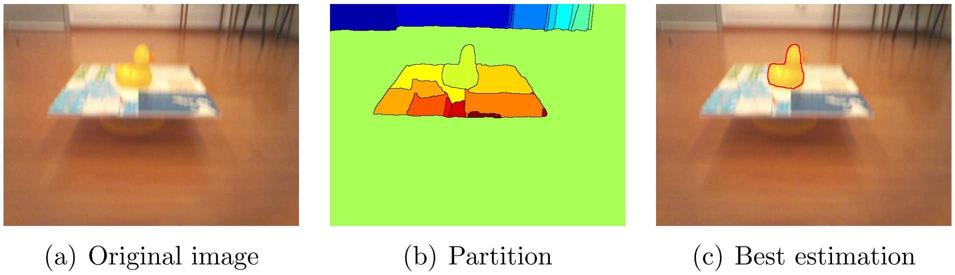


Figure 4.2: Proposed framework to robustly extract a set of object segmentations from different views imposing a minimum quality for its reconstruction. A pre-processing step discards images in which the object may not be correctly segmented. Then, a region-based particle filter obtains object segmentations. Finally, best segmentations are selected to generate the 3D reconstruction.

Pre-processing

As the shape of the object is extracted from a partition, the final quality of the model is highly dependent on the image partitions generated along the sequence. These object segmentations at different views are further used by a space carving algorithm [99] to reconstruct the 3D model. Although the smoothness of this model increases with the number of segmentations extracted from different views, the larger the number of images considered in this process the higher the probability of an erroneous object shape estimation at least in one view. The quality of the final reconstruction is drastically reduced by segmentation errors. Thus, in this application, it is preferred to have a smaller set of high quality segmentations from different view points than a larger number of 2D segmentations that may contain



(a) Original image

(b) Partition

(c) Best estimation

Figure 4.3: In (a) a blurred image is presented. Images (b) and (c) show its associated partition and the best estimation of the object given this partition, respectively. As it can be observed, the blurring effect creates erroneous contours which are not capable to represent a correct segmentation of the object. In this example, the beak of the duck is not included in the object segmentation. As a result, this part of the duck will not be reconstructed.

errors. To this end, a subset of images from the sequence is selected to robustly create a 3D reconstruction of the object.

Two main situations can be found in which a region-based particle filter may not correctly recover the shape of the object. First, when a part of the object is not present in the image. And second, when the blurring effect degradates the quality of the object contours. In order to avoid erroneous estimations, two pre-processing steps select those images in which the object can be correctly segmented. These steps estimate the position of the object in the scene and the blurring of the image respectively.

Blurring estimation

Blur is one of the conventional image quality degradations and it can be caused by various factors. In our application, this effect arises due to the rapid camera movement of the robot. The quality of partitions decreases drastically when the blurring effect appears, producing corrupted contours and mixing object and background pixels in the same regions (Figure 4.3).

Since the image gradient is highly related to image blurring [100], our blur detector computes the magnitude of this gradient to estimate the blur present in an image. Then, a histogram of the gradient is built (in this work, 20 bins have been used). As the contours of a clear image are more precisely defined than the contours of a blurred image, its histogram is expected to contain some contours with large values. On the contrary, contour magnitudes of blurred images should be small.

To this extend, the accumulation of the last 10 bins of the histogram is used to classify the image. If this summation represents more than 0.5% of contour pixels, the image is classified as clear. Otherwise, the blurring effect is said to be present.

Position estimation

The position of the object in the scene is computed using its relative position with respect to the camera. Due to the camera movement, the object may not be completely observed, and some of its parts can be projected out of the image. When this situation arises and the image is selected to generate the 3D model, the part which is not included in the scene will not appear in the final reconstruction even if it is correctly segmented in other views. To avoid this problem, a classical implementation of a color-based particle filter [47] is used to estimate the position and the bounding box of the object along the sequence. Following a conservative policy, images where the detected bounding box is closer than 25 pixels to an image border are not taken into account to extract the object contours.

Region-based Particle Filter

In this application, the Region-based Particle Filter presented in Chapter 2 is used to segment the object along a sequence propagating its shape through time. To this end, similarities between regions are analyzed. Then, parts associated with both the object and background are put in correspondance for each pair of views.

Finally, the estimation of the object is obtained as the combination of the state of the particles. Note that in the region-based case each particle has its own associated object shape obtained through the two previous steps. Thus, the object shape is estimated combining the masks of all the particles. As a result of this combination, a certain probability of belonging to the object is assigned to each region. The final object shape is estimated considering those regions with a probability higher than a given threshold (In this application, 50% has been used). The capacity of estimating the 2D shape of the object in an image view given its shape in a similar view makes this algorithm suitable for reconstruction applications.

Image selection

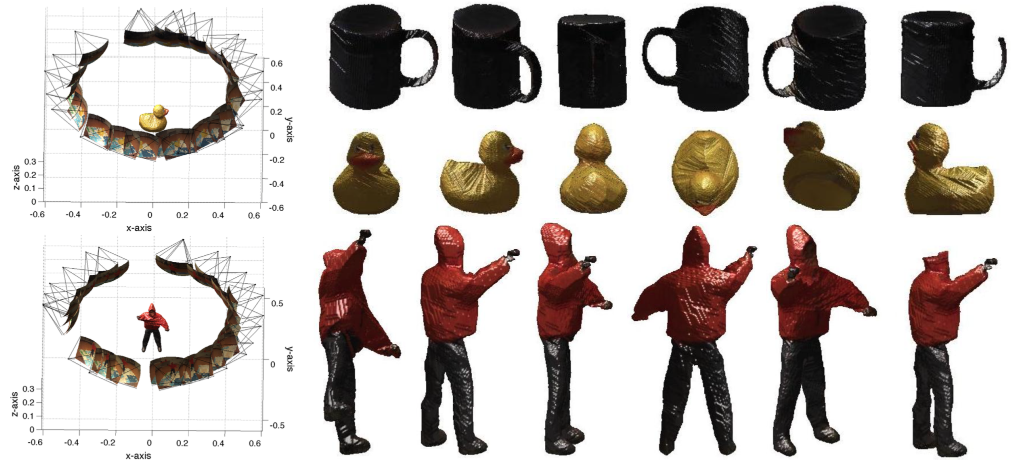
As it has been previously commented, errors in the object shape estimation rapidly degradate the quality of the final reconstruction. In order to avoid this degradation, only a subset of the views analyzed by the region-based particle filter are used to create the 3D model.

Images are selected according to the Diffusion distance [101] between the segmented object histogram and the model. This distance models the histogram difference as a diffusion process. Furthermore, it is robust to deformation, lighting change and noise in histogram-based descritors. In addition, it has linear computational complexity which improves other cross-bin distances with quadratic complexity or higher. Using this distance, a similarity coefficient is computed for each image of the sequence:

$$c_k = 1 - d_k \tag{4.1}$$

where d_k is the Diffusion distance between the model and image k.

Moreover, the circular distribution of the cameras is taken into account to correctly represent the entire 3D object. The image with the highest coefficient is chosen first. Then, from the rest of images, the view with the highest coefficient which is not included in a temporal window of 7 frames centered in any chosen image is selected. This process is iterated until 25 frames are chosen or the coefficient falls below a threshold. The resulting set of views is used to robustly reconstruct the object as it can be observed in Figure 4.4.



(a) Camera positions of the selected video frames for Duck and Action Man.

(b) Rendered views of the estimated 3D geometry.

(c) Incorrect reconstructions.

Figure 4.4: 3D shape estimation results. The selected video frames for reconstruction are shown, for two of the objects, in (a). Random views of the recovered 3D models are shown in (b). Incorrect reconstructions of the object as a consequence of adding a low quality segmentation appear in (c). For rendering the different views, we applied a voxel coloring method that assigns, for each surface voxel, its corresponding pixel color taken from the camera that is closer to the voxel.

4.3 Multiresolution co-clustering for uncalibrated multiview segmentation

In this application, we present an extension of the co-clustering algorithm presented in Chapter 3 to a multiview scenario using semantic information. The objective of this work is to perform semantic segmentation given a set of views of the same scene. This application has been developed in the PhD thesis of Carles Ventura Royo with remarcable results and illustrates the large number of scenarios in which our co-clustering technique may be used.

Semanic segmentation techniques have experimented a drastic quality increase due to the recent introduction of Convolutional Neural Networks (CNNs). One of the key aspects of CNNs is that they require large amounts of annotated visual content to be trained. Global scale labels combined with pixel-wise annotations have allowed the training of CNNs for the semantic segmentation task.

Recently, the limitations of such annotated databases have been exhaustively analyzed paying attention, among other aspects, to both the generalization across datasets and to the balance, location and size of the annotations. As a result, a strong bias towards some specific objects has been reported (e.g.: 25-30% of the instances are from the person class). On the contrary, the high variability of other classes is not correctly reflected in the databases (differences among instances of a concept, i.e. intra-class variability, or among views of a given instance, i.e. view variability). This leads to a large variation in semantic segmentation performance for different classes. Fourth row of Figure 4.5 shows an example of strong changes in performance due to view variability.

The problem of view variability can be palliated if several views of the scene are available and jointly processed. This implies putting into correspondence objects in the various views of the scene. The task of multiview segmentation, which can be very accurately solved when the camera parameters are known, becomes much more complicated when these parameters are not available.

Several approaches can be followed to tackle uncalibrated multiview segmentation: typically, extending video segmentation techniques or using co-segmentation algorithms. In Chapter 3 of this thesis, it is reported that, in the context of video segmentation of scenes with little motion, co-clustering techniques outperform other approaches. Moreover, as previously discussed in that chapter, multiresolution region-based image representations have shown to provide a richer framework that improves the performance of subsequent analysis. In the current application, an extension of the co-clustering approach presented in Chapter 3 of this thesis towards a two-step multiresolution co-clustering for uncalibrated multiview is explored. The first step allows the algorithm to reach a given resolution in the representation, whereas the second step introduces label coherence through the set of views. Both steps are alternatively applied in an iterative approach and their combination provides a multiresolution, multiview representation with a reduced, yet accurate, set of labels. Second and third rows of Figure 4.5 show the co-clustering results obtained with the algorithm described in Chapter 3 and the two-step approach presented in this application, respectively. As it can observed, the two-step approach obtains better correspondances in a multiview segmentation scenario than the method presented in Chapter 3 due to a co-clustering divided in two

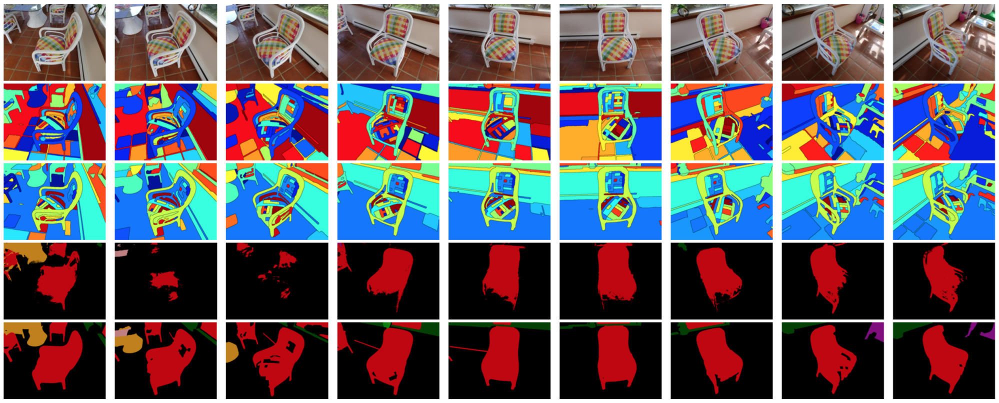


Figure 4.5: Generic and semantic segmentation results. First row: Original views. Second row: Co-clustering from Chapter 3. Third row: Best level of the proposed multiresolution generic coclustering. Fourth row: Semantic segmentation of [3]. Fifth row: Proposed automatic multiview semantic segmentation. Note the improvements in coherence in the labeling of the generic segmentation (rows 2 and 3), and in the object representation in several views of the semantic segmentation (rows 4 and 5). These results belong to the PhD thesis of Carles Ventura Royo.

steps and the introduction of motion compensation.

The second contribution of this application is a semantic multiresolution co-clustering for uncalibrated multiviews. Given the previous co-clustering result, a global optimization is applied. Semantic information is introduced in this optimization to further improve the quality of the multiresolution, multiview representation.

The third contribution is an unsupervised resolution selection technique that, using the semantic information, obtains a single, multiview coherent labeling with an accuracy close to the multiresolution representation. This unsupervised technique has been used to select the resolution for the results presented in the third and fifth rows of Figure 4.5. In Figure 4.6, a qualitative assessment of these contributions on the datasets from [4] can be observed.

These and other results can be found in the PhD thesis of Carles Ventura Royo, in which this technique is compared, in the generic case, with two baselines and five state-of-the-art techniques [1, 2, 82, 102, 85] and, in the

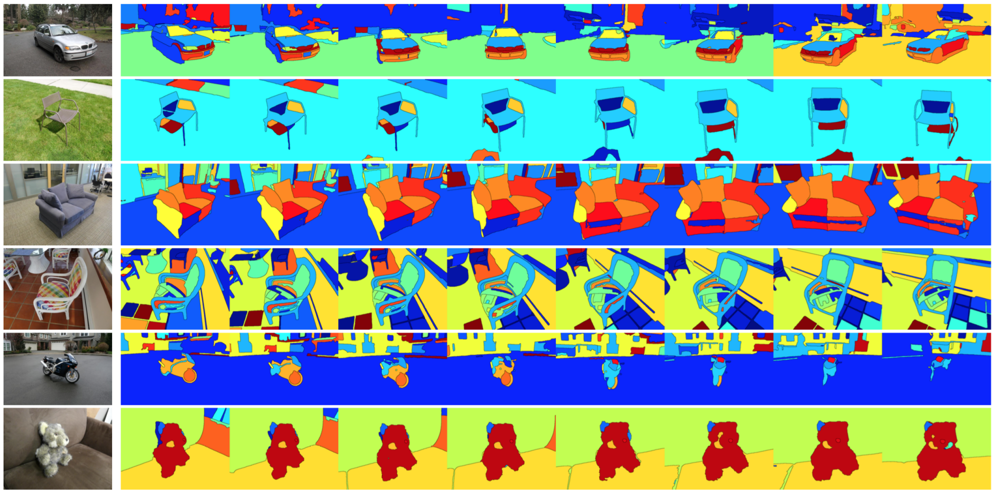


Figure 4.6: Qualitative assessment for generic co-clustering applied to BMW, Chair, Couch, GardenChair, Motorbike and Teddy datasets [4]. First column: original images. Other columns: results for generic two-step iterative co-clustering.

semantic case, with one baseline and one CNN technique [3]. Comparisons are made using the implementations from respective authors.

4.4 Scene segmentation via rate-distorsion optimization

In this application, we present an particularization of the co-clustering optimization presented in Chapter 3 to tackle a rate-distorsion multiview problem. The objective of this work is to perform scene segmentation given a set of views using a rate-distorsion measure over their associated hierarchies. This application has been developed in the PhD thesis of Marc Maceira Duch obtaining very good results.

RGB-D data provides dense representation of the scene from a single view. In multiview sequences, depth data associated with each view heavily augments the amount of data needed to store the 3D information. In this context, extracting a 3D model of a scene from multiple depth maps removes redundant information of the views while obtaining a unique 3D representation of the scene. This representation can be further used in tasks as action detection, scene recognition or scene labeling.

The use of 3D planar models to represent the structure of a scene has been explored in multiple applications. 3D planar models have shown to be useful to extract the structure of a scene using only color data, to extract 3D planes from stereo configurations or to co-segment multiple view objects. Those representations allow to extract a 3D model of the scene while segmenting multiple views.

In this application, we propose a method to jointly extract a 3D planar representation and a consistent segmentation of the scene from multiple depth maps and the camera information of each view. The information of each depth map is projected to the 3D domain using the camera parameters. An initial grouping algorithm is used to obtain a relation between segments of the different views. This information is back-projected to an unique view where a hierarchy of regions is built. By assigning a rate-distortion measure to each node of the hierarchy in this unique view, we are able to retrieve the optimal representation from the hierarchy in terms of rate-distortion. A diagram of this framework can be observed in Figure 4.7.

Using this technique, we obtain two important outputs. First, a planar decomposition of multiple depth maps with a rate-distortion inspired method. The set of planes obtained represents regions in the multiple views with the same 3D plane. Second, a consistent segmentation for the multiple views. This segmentation is used in a depth-map coding application

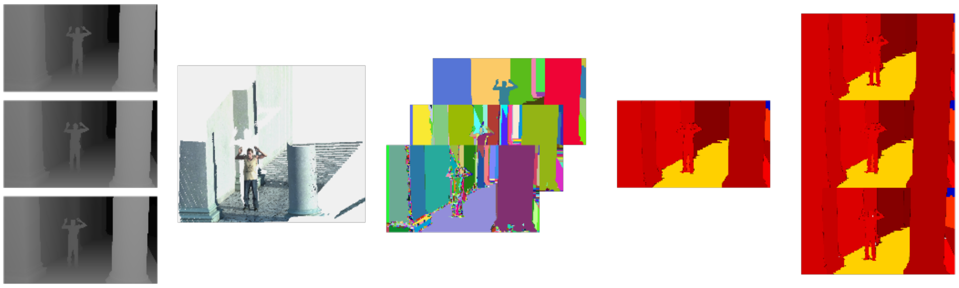


Figure 4.7: From left to right: Depth images of multiple views. The depth information of the multiple views are back-projected to the 3D world to generate a unique point cloud. The point cloud is segmented in the 3D domain and projected to a reference view where a hierarchy of regions is built. A rate-distortion optimization finds the optimal partition in the hierarchy. The partition in the reference view defines a partition in each of the input views.

showing the benefits of the proposed representation.

The results associated to this research can be found in the PhD thesis of Marc Maceira Duch, in which the resulting segmentation is used to robustly encode depth maps of multiple views over-performing the HEVC coding standard.

Chapter 5

Conclusions

This thesis is mainly divided in three parts. In the first part, a new formulation of the particle filter theory using regions has been presented. To perform both **object tracking and segmentation** using this algorithm, we have considered the particles that represent the object as unions of regions from a partition and we have included this partition as part of the measurement. We have gradually developed this theory, from a hibrid tracker using regions from a partition fitting a geometrical shape for object representation towards a generic region-based object segmentation algorithm that extends the particle-filter theory. During this transition, the propagation of particles between consecutive frames is paramount as they are no longer represented by a geometrical shape. We have developed a joint optimization to propagate the complete set of particles using a single process. To do so, we have used a co-clustering that relies on the similarities between contour elements from partitions and optical flow. Using this approach, particles shape can be adapted to object deformations along the sequence. The findings of this part of the Thesis were published in:

D. Varas and Marqués, F., "A Region-Based Particle Filter for Generic Object Tracking and Segmentation", in ICIP - International Conference on Image Processing, Orlando, 2012.

D. Varas and Marqués, F., "Region-based Particle Filter for Video Object Segmentation", in CVPR - Computer Vision and Pattern Recognition, Ohio, 2014.

We have conducted experimental validation on the LabelMe Video and the

Segtrack datasets. We have performed a comparison in terms of segmentation performance with six state-of-the-art algorithms and we have shown that the use of a particle filter methodology combined with a region-based representation of the object robustly tracks object in different situations, including rapid changes in position, strong deformations and sequences in which the objects and background are very similar. Moreover, we have showed that using this approach, the number of particles used to track the object can be drastically reduced.

From the point of view of comparing the final segmentation of the tracked object, we have shown that using a refinement step based on the Bayes rule to improve the segmentation quality and guide the randomness associated with the particle filter may further improve the results. We plan to submit the findings of this part of the Thesis to a journal this year.

The importance of particles propagation to accurately segment objects along video sequences motivated a deep study of the co-clustering technique. This study revealed a great potential of this technique that has been investigated in the second part of this Thesis.

This part has been focused on the study of **multi resolution video segmentation**. We have presented a novel technique that performs unsupervised video segmentation at multiple resolutions using a set of hierarchies associated with frames from sequences with small variations. To do so, we have performed an optimization on these hierarchies that fully explotes the tree information avoiding inconsistencies of previous clustering approaches. Moreover, we have developed an iterative approach to perform video segmentation that combines the information at different resolutions. The findings of this part of the Thesis were published in:

D. Varas, Alfaro, M., and Marqués, F., “Multiresolution hierarchy co-clustering for semantic segmentation in sequences with small variations”, in ICCV - International Conference on Computer Vision, 2015.

We have conducted experiments on the Video Occlusion/Object Boundary Detection dataset comparing our technique with both video segmentation and co-segmentation approaches. We have proved that clustering nodes from a set of non-coherent hierarchies without restricting this process to the merging sequences associated with the trees, can robustly produce video segmentation at different resolutions.

The last part of this Thesis has been devoted to the **development of**

tools which use the object and video segmentation techniques that have been previously discussed. We have shown that these techniques can be usefull to tackle various problems in different scenarios.
First, our region-based particle filter has been used to segment objects from sequences generated by a humanoid robot in order to perform 3D reconstruction extracting a set of different views of the object. Due to the stepping of the humanoid, the recorded sequence is contaminated with artefacts that affect the correct extraction of contours along the video frames. To overcome this issue, the best segmentations generated by our method were selected taking into account not only the score associated with the estimation, but also the robot position. A subset of camera poses and video frames were obtained to produce consistent 3D shape estimations of the objects. The findings of this part of the Thesis were published in:

P. A. Martínez, Varas, D., Castelán, M., Camacho, M., Marqués, F., and Arechavaleta, G., "3D Shape Reconstruction from a Humanoid Generated Video Sequence", in IEEE International Conference on Humanoid Robots, Madrid, 2014.

Second, we have explored the extension of our multi-resolution video segmentation technique to the context of uncalibrated multiview segmentation. We have developed a two-step iterative co-clustering for uncalibrated views that provides a coherent multiresolution representation. Then, we have exploited semantic information to propose a global semantic multi resolution co-clustering optimization. Finally, we proposed an unsupervised resolution selection technique that automatically obtains a single coherent labeling of the views. This work has been developed in the PhD Thesis of Carles Ventura and the finding of this part will be submitted to a journal this year.
Third, we have used the hierarchical co-clustering to tackle a rate-distortion optimization in a multiview scene segmentation scenario. The point cloud of the multiple views is projected to a single view where a hierarchical structure is built. A rate-distortion optimization procedure is applied in a hierarchical representation in that view obtaining an optimal partition inside the hierarchy. The resulting partition in the reference view jointly segments the multiple views. This work has been developed in the PhD Thesis of Marc Maceira and the finding of this part have been submitted to ICASSP 2017.

Bibliography

[1] M. Grundmann, V. Kwatra, M. Han, and I. Essa, "Efficient hierarchical graph based video segmentation," *IEEE CVPR*, 2010.

[2] C. Xu, C. Xiong, and J. J. Corso, "Streaming hierarchical video segmentation," in *Proceedings of European Conference on Computer Vision*, 2012. [Online]. Available: http://web.eecs.umich.edu/~jjcorso/pubs/jcorso_ECCV2012_streamgbh.pdf

[3] S. Zheng, S. Jayasumana, B. Romera-Paredes, V. Vineet, Z. Su, D. Du, C. Huang, and P. Torr, "Conditional random fields as recurrent neural networks," in *International Conference on Computer Vision (ICCV)*, 2015.

[4] S. N. S. Adarsh Kowdle and R. Szeliski, "Multiple view object cosegmentation using appearance and stereo cues," in *European Conference on Computer Vision (ECCV 2012)*, October 2012.

[5] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *Computer Vision and Image Understanding*, vol. 115, no. 2, pp. 224 – 241, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1077314210002171

[6] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon, "Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera," in *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, ser. UIST '11. New York, NY, USA: ACM, 2011, pp. 559–568. [Online]. Available: http://doi.acm.org/10.1145/2047196.2047270

[7] W. Hu, N. Xie, L. Li, X. Zeng, and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 41, no. 6, pp. 797–819, Nov 2011.

[8] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, 2004. [Online]. Available: http://dx.doi.org/10.1023/B:VISI.0000022288.19776.77

[9] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, "Contour detection and hierarchical image segmentation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 898–916, May 2011. [Online]. Available: http://dx.doi.org/10.1109/TPAMI.2010.161

[10] K. Maninis, J. Pont-Tuset, P. Arbeláez, and L. V. Gool, "Convolutional oriented boundaries," in *European Conference on Computer Vision (ECCV)*, 2016.

[11] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 844–851, Aug. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.868685

[12] D. B. Goldman, C. Gonterman, B. Curless, D. Salesin, and S. M. Seitz, "Video object annotation, navigation, and composition," 2008.

[13] J. Wang, Y. Xu, H.-Y. Shum, and M. F. Cohen, "Video tooning." Association for Computing Machinery, Inc., August 2004. [Online]. Available: http://research.microsoft.com/apps/pubs/default.aspx?id=69004

[14] J. Shi and J. Malik, "Motion segmentation and tracking using normalized cuts," in *Computer Vision, 1998. Sixth International Conference on*, Jan 1998, pp. 1154–1160.

[15] M. Irani, B. Rousso, and S. Peleg, "Computing occluding and transparent motions," *International Journal of Computer Vision*, vol. 12, pp. 5–16, 1994.

[16] W. Brendel and S. Todorovic, “Video object segmentation by tracking regions,” in *2009 IEEE 12th International Conference on Computer Vision*, Sept 2009, pp. 833–840.

[17] J. Sivic, F. Schaffalitzky, and A. Zisserman, “Object level grouping for video shots,” *International Journal of Computer Vision*, vol. 67, no. 2, pp. 189–210, 2006. [Online]. Available: http://dx.doi.org/10.1007/s11263-005-4264-y

[18] X. Bai, J. Wang, D. Simons, and G. Sapiro, “Video snapcut: Robust video object cutout using localized classifiers,” *ACM Trans. Graph.*, vol. 28, no. 3, pp. 70:1–70:11, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1531326.1531376

[19] R. E. Kalman, “A new approach to linear filtering and prediction problems,” *Transactions of the ASME – Journal of Basic Engineering*, vol. 82, pp. 35–45, 1960.

[20] Y. Bar-Shalom and T. Foreman, *Academic and Data Association*, A. P. Inc, Ed., 1988.

[21] N. Gordon, D. Salmond, and A. Smith, “Novel approach to nonlinear/non-gaussian bayesian state estimation,” *Radar and Signal Processing, IEEE Proceedings F*, vol. 140, no. 2, apr 1993.

[22] J. Arróspide, L. Salgado, and M. Nieto, “Multiple object tracking using an automatic variable-dimension particle filter.” in *ICIP*. IEEE, 2010, pp. 49–52. [Online]. Available: http://dblp.uni-trier.de/db/conf/icip/icip2010.html#ArrospideSN10

[23] M. S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, “A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking,” *IEEE Trans. on Signal Processing*, vol. 50, no. 2, 2002.

[24] A. Yilmaz, O. Javed, and M. Shah, “Object tracking: A survey,” *ACM Computing Surveys*, vol. 38, p. no 4, 2006.

[25] A. Bugeau and P. Pérez, “Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts,” *J. Image Video Process.*, vol. 2008, pp. 3:1–3:14, Jan. 2008. [Online]. Available: http://dx.doi.org/10.1155/2008/317278

[26] D. Tsai, M. Flagg, A. Nakazawa, and J. Rehg, “Motion coherent tracking using multi-label MRF optimization,” *International Journal of Computer Vision*, vol. 100, no. 2, 2012. [Online]. Available: http://dx.doi.org/10.1007/s11263-011-0512-5

[27] Y. Yang and G. Sundaramoorthi, “Modeling shape, appearance and self-occlusions for articulated object tracking,” *CoRR*, vol. abs/1208.4391, 2012.

[28] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, “Video segmentation by tracking many figure-ground segments,” in *ICCV*, 2013.

[29] J. Yuen, B. C. Russell, C. Liu, and A. Torralba, “Labelme video: building a video database with human annotations,” *ICCV*, 2009.

[30] P. Fieguth and D. Terzopoulos, “Color-based tracking of heads and other mobile objects at video frame rates,” in *CVPR*, 1997.

[31] D. Comaniciu and P. Meer, “Mean shift: a robust approach toward feature space analysis,” *PAMI*, vol. 24, no.5, May 2002.

[32] C. Stauffer and W. E. L. Grimson, “Adaptive background mixture models for real-time tracking,” in *Computer Vision and Pattern Recognition, 1999. IEEE Computer Society Conference on.*, vol. 2, 1999, p. 252 Vol. 2.

[33] E. Maggio and A. Cavallaro, “Hybrid particle filter and mean shift tracker with adaptive transition model,” in *Proc. Int. Conf. Acoustics, Speech, and Signal Processing*, 2005, pp. 221–224.

[34] V. Vilaplana and D. Varas, “Face tracking using a region-based mean-shift algorithm with adaptive object and background models,” *WIAMIS*, pp. 17–20, 2009.

[35] V. Vilaplana and F. Marques, “Region-based mean shift tracking: Application to face tracking,” in *ICIP*, 2008.

[36] A. Nakhmani and A. Tannenbaum, “Particle filtering with region-based matching for tracking of partially occluded and scaled targets,” *SIAM Journal on Imaging Sciences*, vol. 4, no. 1, pp. 220–242, 2011.

[37] C. Bibby and I. Reid, "Robust real-time visual tracking using pixel-wise posteriors," in *ECCV*, 2008.

[38] P. Chockalingam, N. Pradeep, and S. Birchfield, "Adaptive fragments-based tracking of non-rigid objects using level sets," in *ICCV*, 2009.

[39] D. Zhang, O. Javed, and M. Shah, "Video object segmentation through spatially accurate and temporally dense extraction of primary object regions," in *CVPR*, 2013.

[40] Y. J. Lee, J. Kim, and K. Grauman, "Key-segments for video object segmentation," ser. ICCV, 2011. [Online]. Available: http://dx.doi.org/10.1109/ICCV.2011.6126471

[41] T. Ma and L. J. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation." in *CVPR*. IEEE, 2012, pp. 670–677.

[42] D. Banica, A. Agape, A. Ion, and C. Sminchisescu, "Video object segmentation by salient segment chain composition," in *ICCV Workshops*, 2013.

[43] M. Isard and A. Blake, "CONDENSATION - conditional density propagation for visual tracking," *International Journal of Computer Vision*, vol. 29, pp. 5–28, 1998.

[44] K. Nummiaro, E. Koller-Meier, and L. V. Gool, "A color-based particle filter," *First International Workshop on Generative-Model-Based Vision*, pp. 53–60, 2002.

[45] H. Ling and K. Okada, "Diffusion distance for histogram comparison," in *CVPR*, 2006.

[46] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bulletin of Calcutta Mathematical Society*, 1943.

[47] K. Nummiaro, E. Koller-Meier, and L. Van Gool, "An adaptive color-based particle filter," *Image and Vision Computing*, vol. 21, No. 1, pp. 99–110, 2003.

[48] R. Pawula, “Generalizations and extensions of the fokker- planck-kolmogorov equations,” *Information Theory, IEEE Transactions on*, vol. 13, no. 1, pp. 33–41, 1967.

[49] A. Doucet, S. Godsill, and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering,” *STATISTICS AND COMPUTING*, vol. 10, no. 3, pp. 197–208, 2000.

[50] J. MacCormick and A. Blake, “A probabilistic exclusion principle for tracking multiple objects,” in *Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on*, vol. 1, 1999, pp. 572–578 vol.1.

[51] J. Carpenter, P. Clifford, and P. Fearnhead, “Improved particle filter for nonlinear problems,” *IEE Proceedings - Radar, Sonar and Navigation*, vol. 146, no. 1, pp. 2–7, Feb 1999.

[52] P. Moral and L. Miclo, *Séminaire de Probabilités XXXIV*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, ch. Branching and interacting particle systems approximations of feynman-kac formulae with applications to non-linear filtering, pp. 1–145.

[53] K. Kanazawa, D. Koller, and S. Russell, “Stochastic simulation algorithms for dynamic probabilistic networks,” in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, ser. UAI’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 346–351. [Online]. Available: http://dl.acm.org/citation.cfm?id=2074158.2074197

[54] N. Bergman, “Recursive bayesian estimation: Navigation and tracking applications,” Linköping Studies in Science and Technology. Thesis No 579, May 1999.

[55] J. S. Liu and R. Chen, “Sequential Monte Carlo methods for dynamic systems,” *Journal of the American Statistical Association*, vol. 93, pp. 1032–1044, 1998.

[56] A. Doucet, N. J. Gordon, and V. Krishnamurthy, “Particle filters for state estimation of jump markov linear systems,” *IEEE Transactions on Signal Processing*, vol. 49, no. 3, pp. 613–624, Mar 2001.

[57] P. D. Moral, "Measure-valued processes and interacting particle systems. application to nonlinear filtering problems," *The Annals of Applied Probability*, vol. 8, no. 2, pp. 438–495, 1998. [Online]. Available: http://www.jstor.org/stable/2667309

[58] R. van der Merwe, A. Doucet, N. de Freitas, and E. Wan, "The unscented particle filter," 2000.

[59] G. Kitagawa, "Monte carlo filter and smoother for non-gaussian nonlinear state space models," *Journal of Computational and Graphical Statistics*, vol. 5, no. 1, pp. 1–25, 1996. [Online]. Available: http://www.jstor.org/stable/1390750

[60] N. Gordon, D. Salmond, and A. F. M. Smith, "Methodology for monte carlo smoothing with application to time-varying autoregressions," in *Int. Symp. Frontiers Time Series Modeling*, 2000.

[61] B. P. Carlin, N. G. Polson, and D. S. Stoffer, "A Monte Carlo Approach to Nonnormal and Nonlinear State-Space Modeling," *Journal of the American Statistical Association*, vol. 87, no. 418, pp. 493–500, 1992. [Online]. Available: http://www.jstor.org/stable/2290282

[62] J. Kwon and F. Park, "Visual tracking via particle filtering on the affine group," in *ICIA 2008.*, 2008, pp. 997–1002.

[63] V. Vilaplana, F. Marques, and P. Salembier, "Binary partition trees for object detection," *IEEE TIP*, vol. 17, no. 11, pp. 2201 –2216, nov 2008.

[64] H. Gilad and D. Weinshall, "Motion of disturbances: Detection and tracking of multi-body non rigid motion," in *Motion, Machine Vision and Applications 11*, 1997, pp. 122–137.

[65] D. Varas and F. Marques, "Region-based particle filter for video object segmentation," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 3470–3477.

[66] D. Varas, M. Alfaro, and F. Marques, "Multiresolution hierarchy coclustering for semantic segmentation in sequences with small variations," in *2015 IEEE International Conference on Computer Vision*, December 2015.

[67] A. D. Bimbo and F. Dini, "Particle filter-based visual tracking with a first order dynamic model and uncertainty adaptation," *Computer Vision and Image Understanding*, vol. 115, no. 6, 2011.

[68] D. Glasner, S. Vitaladevuni, and R. Basri, "Contour-based joint clustering of multiple segmentations," in *CVPR*, 2011.

[69] S. Vitaladevuni and R. Basri, "Co-clustering of image segments using convex optimization applied to em neuronal reconstruction," in *CVPR*, 2010.

[70] T. Brox, C. Bregler, and J. Malik, "Large displacement optical flow," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, June 2009, pp. 41–48.

[71] R. E. Kass and A. E. Raftery, "Bayes factors," *Journal of the American Statistical Association*, vol. 90, pp. 773–795, 1995.

[72] F. Ge, S. Wang, and T. Liu, "Image-segmentation evaluation from the perspective of salient object extraction," in *CVPR*, 2006.

[73] J. Pont-Tuset and F. Marques, "Upper-bound region selection to evaluate image segmentation from a figure-ground perspective," *Submitted to ECCV*, 2012.

[74] S. Hongeng, R. Nevatia, and F. Bremond, "Video-based event recognition: Activity representation and probabilistic recognition methods," *Comput. Vis. Image Underst.*, vol. 96, no. 2, pp. 129–162, Nov. 2004. [Online]. Available: http://dx.doi.org/10.1016/j.cviu.2004.02.005

[75] S. Chen, A. Fern, and S. Todorovic, "Multi-object tracking via constrained sequential labeling," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*, 2014, pp. 1130–1137. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.148

[76] A. W. M. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content-based image retrieval at the end of the early years," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 12, pp. 1349–1380, Dec. 2000. [Online]. Available: http://dx.doi.org/10.1109/34.895972

[77] W. Brendel and S. Todorovic, "Video object segmentation by tracking regions," in *Computer Vision, 2009 IEEE 12th International Conference on*, Sept 2009, pp. 833–840.

[78] S. Paris, "Edge-preserving smoothing and mean-shift segmentation of video streams," in *Computer Vision – ECCV 2008*, ser. Lecture Notes in Computer Science, D. Forsyth, P. Torr, and A. Zisserman, Eds. Springer Berlin Heidelberg, 2008, vol. 5303, pp. 460–473. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-88688-4_34

[79] N. Bansal, A. Blum, and S. Chawla, "Correlation clustering," *Machine Learning*, vol. 56, no. 1-3, pp. 89–113, 2004. [Online]. Available: http://dx.doi.org/10.1023/B%3AMACH.0000033116.57574.95

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

[81] C. Xu, S. Whitt, and J. J. Corso, "Flattening supervoxel hierarchies by the uniform entropy slice," in *Proceedings of the IEEE International Conference on Computer Vision*, 2013. [Online]. Available: http://web.eecs.umich.edu/~jjcorso/pubs/jcorso_ICCV2013_hieflat.pdf

[82] F. Galasso, R. Cipolla, and B. Schiele, "Video segmentation with superpixels," in *Asian Conference on Computer Vision*, 2012.

[83] F. Galasso, M. Keuper, T. Brox, and B. Schiele, "Spectral graph reduction for efficient image and streaming video segmentation," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 2014.

[84] C. Rother, T. Minka, A. Blake, and V. Kolmogorov, "Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 1, June 2006, pp. 993–1000.

[85] G. Kim and E. P. Xing, "On multiple foreground cosegmentation," in *25th IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2012)*, 2012.

[86] J. Rubio, J. Serrat, and A. López, “Video co-segmentation,” in *Computer Vision – ACCV 2012*, ser. Lecture Notes in Computer Science, K. Lee, Y. Matsushita, J. Rehg, and Z. Hu, Eds. Springer Berlin Heidelberg, 2013, vol. 7725, pp. 13–24. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37444-9_2

[87] W.-C. Chiu and M. Fritz, “Multi-class video co-segmentation with a generative multi-video model,” in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 321–328.

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

[89] P. Salembier and L. Garrido, “Binary partition tree as an efficient representation for image processing, segmentation and information retrieval,” *IEEE transactions on image processing*, vol. 9, no. 4, p. 561–576, 2000.

[90] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik, “Using contours to detect and localize junctions in natural images,” in *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, June 2008, pp. 1–8.

[91] P. Arbelaez, J. Pont-Tuset, J. Barron, F. Marques, and J. Malik, “Multiscale combinatorial grouping,” in *Computer Vision and Pattern Recognition (CVPR)*, 2014.

[92] M. Everingham, L. Gool, C. K. Williams, J. Winn, and A. Zisserman, “The pascal visual object classes (voc) challenge,” *Int. J. Comput. Vision*, vol. 88, no. 2, pp. 303–338, Jun. 2010. [Online]. Available: http://dx.doi.org/10.1007/s11263-009-0275-4

[93] T. Radzik, “Newton’s method for fractional combinatorial optimization,” in *Foundations of Computer Science, 1992. Proceedings., 33rd Annual Symposium on*, Oct 1992, pp. 659–669.

[94] J. Pont-Tuset and F. Marques, “Supervised assessment of segmentation hierarchies,” in *European Conference on Computer Vision (ECCV)*, 01/2012 2012.

[95] A. Stein and M. Hebert, “Occlusion boundaries from motion: Low-level detection and mid-level reasoning,” *International Journal on Computer Vision*, vol. 82, no. 2, pp. 325–357, April 2009.

[96] A. Joulin, F. Bach, and J. Ponce, “Multi-class cosegmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 542–549.

[97] F. Galasso, N. S. Nagaraja, T. J. Cardenas, T. Brox, and B. Schiele, “A unified video segmentation benchmark: Annotation, metrics and analysis,” in *IEEE International Conference on Computer Vision*, 2013.

[98] F. Li, T. Kim, A. Humayun, D. Tsai, and J. M. Rehg, “Video segmentation by tracking many figure-ground segments,” in *ICCV*, 2013.

[99] K. N. Kutulakos and S. M. Seitz, “A theory of shape by space carving,” *International Journal of Computer Vision*, vol. 38, no. 3, pp. 199–218, 2000. [Online]. Available: http://dx.doi.org/10.1023/A:1008191222954

[100] R. Fergus, B. Singh, A. Hertzmann, S. T. Roweis, and W. T. Freeman, “Removing camera shake from a single photograph,” *ACM Trans. Graph.*, vol. 25, no. 3, pp. 787–794, Jul. 2006. [Online]. Available: http://doi.acm.org/10.1145/1141911.1141956

[101] H. Ling and K. Okada, “Diffusion distance for histogram comparison,” in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, ser. CVPR ’06. Washington, DC, USA: IEEE Computer Society, 2006, pp. 246–253. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2006.99

[102] A. Joulin, F. Bach, and J. Ponce, “Discriminative clustering for image co-segmentation,” in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1943–1950.