This CVPR2014 paper is the Open Access version, provided by the Computer Vision Foundation.
The authoritative version of this paper is available in IEEE Xplore.

Region-based particle filter for video object segmentation

David Varas and Ferran Marques
Universitat Politecnica de Catalunya Barcelona Tech
{david.varas,ferran.marques}@upc.edu [*]

Abstract

We present a video object segmentation approach that extends the particle filter 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 re-formulation of the particle filter. The prediction step uses 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. Particles are defined as unions of regions in the current image partition and their propagation is computed through a single co-clustering. The proposed technique is assessed on the SegTrack dataset, leading to satisfactory perceptual results and obtaining very competitive pixel error rates compared with the state-of-the-art methods.

1. Introduction

In image object tracking, difficulties such as fast object motion, changes of the object patterns or the scene, non-rigid object structures, occlusions and camera motion are common problems that must be handled by the system [28]. In this context, an object trajectory is generated by estimating the object location in each video frame.

Such an estimation has a crucial role in video editing, postprocessing and interactive applications in which the shape of the object should be considered. Objects are usually represented by a geometrical shape (e.g., an ellipse). However, a fixed shape is a too simple representation of real objects and applications using object's shape to extract information about the scene (i.e.: gesture recognition) cannot make use of these trackers and require video object segmentation. 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 [7] solve that problem by including an object segment in 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 value of its applications relies on obtaining accurate segmentations ([24], [27]).

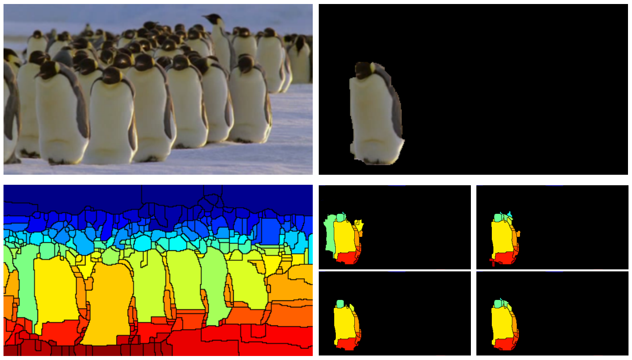


Figure 1. Video object segmentation using a region-based particle filter. From left to right and top to bottom: original image, final object segmentation, image partition and particles of the filter formed with regions from the partition.

In this paper, we propose a video object segmentation based on a novel formulation of the particle filter algorithm using a region-based image representation. The proposed algorithm performs off-line object tracking in a reliable manner and provides an accurate segmentation of the target along the sequence. The inclusion of regions requires the re-formulation of some aspects of the particle filter (Section 4), leading to satisfactory perceptual results and obtaining a very competitive results compared with the state-of-the-art methods even on sequences with rapid movement and drastic object changes (Section 5).

2. Related work

Object tracking and segmentation is addressed in [5] 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 4.2.2). Motion estimation is used

[*] This work has been partially supported by the Spanish Ministerio de Ciencia e Innovación, under project TEC2010-18094.

to obtain video object segmentation in [9]. 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 [29], 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 4.2.1) to predict the position and the shape of the object. In [17], 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 5).

Object tracking is modeled as a Maximum Weight Cliques problem in [20] 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 [3] 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 [15], 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 deformation.

Image-based features for particle filters were introduced by [22]. 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 [18] instead of the Bhattacharyya distance [4] for histogram comparison since it leads to better perceptual performance (Section 4.2.2). As the color of an object can vary through time, the target model is adapted during temporally stable image observations in [23]. Note that [22], [23] 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 [10] is presented in [25]. 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 4). In [21], 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.

3. Particle filters in object tracking

3.1. 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:

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

$$z_k = h_k(x_k, n_k) \tag{2}$$

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, $\{v_{k-1}, k \in \mathbb{N}\}$ and $\{n_k, k \in \mathbb{N}\}$ are i.i.d. noise sequences, and n_x, n_z, n_v, n_n are the dimensions of the state, measurement, state noise vector and measurement noise vector.

3.2. Particle filters

A common way to solve this problem without imposing any constraint is particle filters. 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:

$$p(x_{1:k}|z_{1:k}) \approx \sum_{i=1}^{N_s} w_k^{(i)} \delta(x_{1:k} - x_{1:k}^{(i)}) \tag{3}$$

where the weights $w_k^{(i)}$ are chosen using *importance sampling* [12]. As this posterior is computed using a sequential procedure, weights can be expressed as [2]:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \frac{p(z_k|x_k^{(i)})p(x_k^{(i)}|x_{k-1}^{(i)})}{q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)} \tag{4}$$

where $q(x_k^{(i)}|x_{k-1}^{(i)}, z_k)$ is called *importance density*. Particles are drawn from this function that is a priori not defined.

3.3. Color-based particle filters

The color-based particle is presented here for further comparisons with our region-based approach. This technique estimates the state of an object in a video sequence. 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):

$$x_k = \{x, y, H_x, H_y, \dot{x}, \dot{y}\} \tag{5}$$

$$z_k = I_k \tag{6}$$

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

The filter which is most commonly used for video object tracking in the literature [15], [23], [16] is the *Sampling Importance Resampling* (SIR) filter proposed by [14].

In this SIR 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 4 yields to:

$$w_k^{(i)} \propto w_{k-1}^{(i)} p(z_k|x_k^{(i)}) \tag{7}$$

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 3. Thus, the weights are reset to $w_{k-1}^i = 1/N_s$ and the expression to compute the new weights at k becomes:

$$w_k^{(i)} \propto p(z_k|x_k^{(i)}) \tag{8}$$

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 [14]. 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 1. Usually, this evolution is modeled by a linear stochastic differential equation:

$$x_k^{(i)} = Ax'^{(i)}_{k-1} + Bv_{k-1}^{(i)} \tag{9}$$

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.

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.

3.3.1 Estimation

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

$$E[S_k] = \sum_{i=1}^{N_s} w_k^{(i)} x_k^{(i)} \tag{10}$$

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.

4. Region-based particle filter

In this section, we propose a region-based particle filter to perform video object tracking and segmentation. The algorithm is presented using the previous section structure to allow comparing 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:

$$x_k = \bigcup_{r}^{n_k^o} R_k^r \tag{11}$$

$$z_k = [I_k, P_k] \tag{12}$$

where $P_k = \{R_k^1, R_k^2, ..., R_k^{n_k}\}$ is a partition of the image I_k defined on a domain $\Psi \in \mathbb{R}^2$, n_k is the number of regions that form the partition, n_k^o is the number of regions that characterize the object with $n_k^o \leq n_k$, and $\Psi = \bigcup_{r=1}^{n_k} R_k^r$.

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. Moreover, particle propagation is no longer only dependent on their previous state, it also depends on P_k (measurement). Thus, the new importance density is:

$$q(x_k|x_{k-1}^{(i)}, z_k) = p(x_k|x_{k-1}^{(i)}, z_k) \tag{13}$$

As shown in [12], 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} [19] is maximized. Replacing Equation 13 into Equation 4:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \int p(z_k|x_k')p(x_k'|x_{k-1}^{(i)})dx_k' \tag{14}$$

This integral over Ω involves the states represented by all the regions of the partition and all their possible combinations, which can be represented by a summation:

$$w_k^{(i)} \propto w_{k-1}^{(i)} \sum_c p(z_k|x_k^c)p(x_k^c|x_{k-1}^{(i)}) \tag{15}$$

This summation becomes intractable using a brute force approach. For each $w_{k-1}^{(i)}$, its probability of being represented by all the possible solutions in Ω (regions and combinations of regions of P_k) should be computed $p(x_k^c|x_{k-1}^{(i)})$ and evaluated $p(z_k|x_k^c)$ (Section 4.3).

4.1. 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 15 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* [2].

4.2. Propagation

The propagation of particles between consecutive time instants is usually calculated in two steps ([15], [23], [6]).

4.2.1 Prediction

Prediction is performed to ensure a minimum quality of the particles estimation. We propose to perform particle prediction as a global process using the information provided by all particles. 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. The co-clustering proposed in [13] is adapted to be used in a tracking scheme. Its keypoints for the case of two partitions are briefly described to motivate our choice and modifications, as well as to present some results (Section 4.3).

Co-clustering: Let us consider a pair of images $\{I_j\}_{j=k-1,k}$ and their associated partitions $\{P_j\}_{j=k-1,k}$. Each of these partitions is formed by a group of n_j regions $P_j = \{R_j^1, R_j^2, ..., R_j^{n_i}\}$ where $\Psi \in \mathbb{R}^2$ and $\Psi = \cup_{r=1}^{n_j} R_j^r$.

A co-clustering between partitions is defined by $X \in \{1,0\}^{n \times c}$, where $n = n_{k-1} + n_k$ and c is the number of clusters. Each column x_l with $l \in \{1, ..., c\}$ corresponds to a single cluster. Regions from partitions are assigned to only one cluster if matrix X is constrained to have unitary rows. The score associated with X is computed as:

$$tr(X^T Q X) = \sum_{l=1}^{c} x_l^T Q x_l \tag{16}$$

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 the elements of region contours [13].

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 [13], intra image similarity is proportional to the number of contour elements that share both regions and their color similarity. In turn, inter image similarities are captured comparing the HOG descriptor of their contour elements that are closer than 20 pixels. Then, co-clustering of both partitions becomes an optimization problem:

$$\begin{aligned} &\max_X tr(X^T Q X) \\ &X_{i,j} \in \{0,1\} \forall i,j \quad \sum_j X_{i,j} = 1 \forall i \end{aligned} \tag{17}$$

This is a Quadratic Semi-Assignment Problem (QSAP) [26] for which a Linear Programming relaxation was presented by [8] imposing distances between regions based on the triangular inequality. Further relaxation approaches ([26],[13]) make use of distances defined over cliques in a region adjacency graph. Considering these relaxations, this optimization can be stated as:

$$\begin{aligned}&\min_{D} \sum_{i,j} q_{i,j} d_{i,j}\\ &0 \leq d_{i,j} \leq 1 \quad d_{i,i} = 0 \, \forall i \quad d_{i,j} = d_{j,i} \, \forall i,j\\ &d_{i,j} \leq d_{i,k} + d_{k,j} \quad \forall e_{i,j}, e_{i,k}, e_{k,j} \in G \end{aligned} \tag{18}$$

where G is the previous adjacency graph and $d_{i,j} \in \{0,1\}$ is the distance between regions i and j. Regions that belong to the same cluster have distances equal to 0.

Prediction using co-clustering: Let us now extend the previous co-clustering [13] to our tracking problem. At time k, let us suppose that the tracked object has been estimated at the previous time instant and that an *object partition* P^o_{k-1} has been generated clustering regions from a fine partition. In the ideal case, only two clusters (object and background) form this partition. As we can consider these clusters as regions in a different scale, we will denote them as regions. Finally, we consider that an image I_k and a *fine partition* P_k (measurement) are available at the present time instant.

Before the optimization, similarities expected between regions from both partitions must be analyzed. Intra image similarities are computed as in [13]. Inter image similarity between regions l and m from partitions P_k and P^o_{k-1} respectively should be proportional to $p^{l,m}_{(k,k-1)} = p(R_{l,k}|R_{m,k-1})$. Three types of changes are considered in our algorithm to model differences between regions from consecutive partitions: *changes of color/illumination*, *deformations* and *changes of position*. In terms of probability, we consider these processes to be independent:

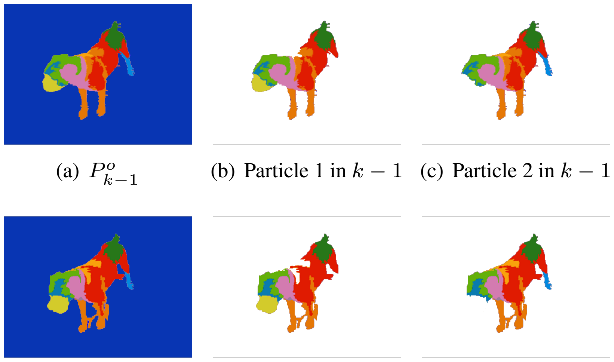
$$p^{l,m}_{(k,k-1)} = p^{C_{l,m}}_{(k,k-1)} p^{D_{l,m}}_{(k,k-1)} p^{P_{l,m}}_{(k,k-1)} \tag{19}$$

The color information is obtained from a local histogram of pixels from the image in a neighborhood of 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 centered on the pixel of the region which is closer to the boundary in that direction and they are averaged. To compute deformations, the information about the shape of the boundary around each contour element is captured computing a HOG descriptor [11]. Finally, changes of position are computed using the Euclidean distance between elements.

Similarity between contour elements is computed as $W_{(k,k-1)}(l,m) = e^{(f_{l,k}-f_{m,k-1})^T \Sigma^{-1} (f_{l,k}-f_{m,k-1})}$, where $f_{q,j}$ is the feature vector of element q that belongs to the partition at time j. This vector is formed as the concatenation of the three types of descriptors that have been previously described. In order to model fast movement and deformations, we allow contour elements which are closer than 100 pixels to be matched. Otherwise, $W_{(k,k-1)}(l,m) = 0$.

In order to obtain a consistent co-clustering, similarity information is propagated using an adjacency graph. In [13], regions from P_k and P^o_{k-1} are considered adjacent if at least one pixel of each region shares the same position at both partitions. As in a tracking problem this constrain may not hold because of object and camera motion, we define a tracking-oriented adjacency graph. In it, we allow any region from P^o_{k-1} to potentially be represented by any region in P_k. We model this defining that each region from P^o_{k-1} is adjacent to all the regions from P_k. Partition P^o_{k-1} is formed by the minimum number of regions necessary to define all particles in $k-1$. This allows us to perform all particle propagations with a single co-clustering. As regions in P^o_{k-1} have been previously labeled as object or background, those having different labels are considered to be disconnected. In turn, adjacency between regions in P_k is defined using spatial connectivity.

Once the optimization problem is solved, a vector d of distances between regions is obtained. This vector assigns to each region of P_k a label from P^o_{k-1}. Accordingly, each particle at $k-1$ can be propagated to k by selecting those regions of P_k that have been assigned the same label than the regions that formed the particle in P^o_{k-1} (Figure 2). As a consequence, a global estimation of the movement for all particles is achieved using a single optimization operation.



(a) P^o_{k-1} (b) Particle 1 in $k-1$ (c) Particle 2 in $k-1$

(d) Co-cluster in k (e) Prediction 1 in k (f) Prediction 2 in k

Figure 2. In (a) and (d) two clustered partitions at $k-1$ and k are presented. Images (b) and (c) show two different particles at $k-1$ and their propagation can be observed in (e) and (f) respectively.

4.2.2 Perturbation

The second step of the propagation process is to perturb 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 a 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^C_k after the estimation step. As the optimization has been globally performed for all particles, P_k^C also defines the union of regions that form each particle. Thus, N_s points of Ω are sampled to analyze. 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 13, 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 15. 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:

$$p_C(R_j) = \begin{cases} 1 - L'(R_j), & \text{if } j \text{ is part of the anchor point} \\ L'(R_j), & \text{otherwise} \end{cases} \tag{20}$$

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.

4.3. Evaluation

In the evaluation step, particles are weighted according to Equation 15. 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 [13]. 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:

$$\begin{aligned} w_k^{(i)} &\propto w_{k-1}^{(i)} \sum_c p(z_k|x_k^c) p(x_k^c|x_{k-1}^{(i)}) = \\ &= w_{k-1}^{(i)} \sum_c \left(m^T Z x_k^c\right)\left((x_k^c)^T Q' x_{k-1}^{(i)}\right) = \\ &= w_{k-1}^{(i)} m^T Z X Q' x_{k-1}^{(i)} \end{aligned} \tag{21}$$

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 4.2.1.

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.

4.4. Estimation

The object is estimated averaging the states of the particles. In the color-based approach, as all samples have the same geometrical shape, the average of the particles 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 step (Section 4.2). Thus, the object shape is estimated combining the masks associated with all particles.

Let $M^{(i)}$ be the binary mask associated with the ith par-

	Ours	[29]	[3]	[20]	[24]	[17]	[9]
Birdfall	243	**155**	166	189	252	288	454
Cheetah	**391**	633	661	806	1142	905	1217
Girl	1935	1488	**1214**	1698	1304	1785	1755
Monkeydog	497	**365**	394	472	563	521	683
Parachute	**187**	220	218	221	235	201	502
Penguin	**903**	-	-	-	1705	136285	6627

Table 1. Segmentation results

	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. Precision and recall analysis.

ticle. The average mask A_M is computed as:

$$A_M = \sum_{i=1}^{N_s} w^{(i)} M^{(i)} \tag{22}$$

where N_s is the total number of particles and $w^{(i)}$ is the weight of the particle after the *Evaluation* step (see Section 4.3). 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.

5. Experiments

In this section, we present both qualitative and quantitative assessment of our region-based particle filter. The SegTrack dataset [24] is used for the evaluation and comparison with other state of the art methods due to its accurate frame annotation. In all experiments, segmentations have been performed using [1] 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 [17], [24], [9], [29], [3], [20]. As it can be observed in Table 1, our method outperforms the results of these methods in three out of six sequences of the SegTrack database (*cheetah*, *parachute* and *penguin*), in two sequences it is ranked the fourth (*birdfall* and *monkeydog*) and in one sequence it is ranked the seventh (*girl*).

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 we present mean and variance of precision and recall values for each sequence.

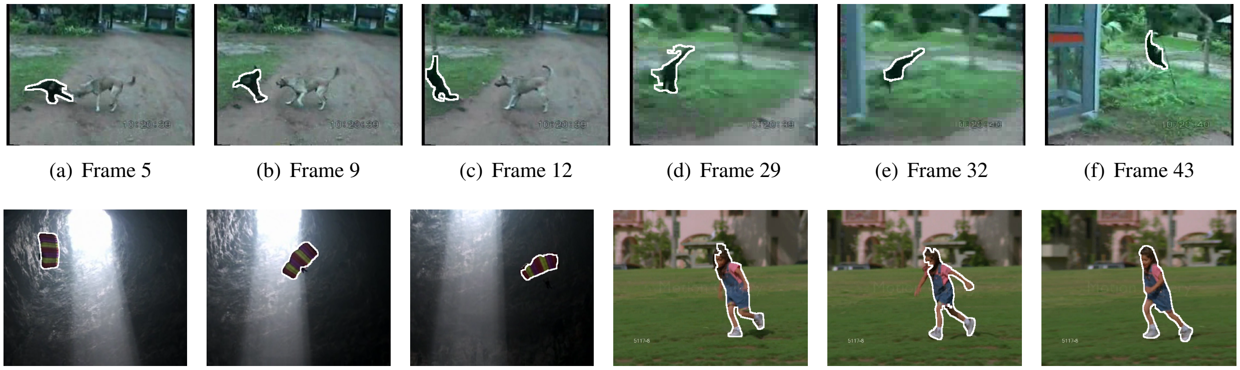
From the qualitative point of view, the results on the *monkeydog* video are particularly significant in two main aspects. First, the correct prediction of the object in a sequence with rapid movement is performed thanks to a tracking-oriented graph and a co-clustering scheme oriented to this task as presented in Section 4.2. This estimation would not be possible considering adjacency between regions as in [13]. Second, considering color information improves the result of the co-clustering when the shape of the object suffers strong deformations. At the first row of Figure 3, several qualitative results are presented. As it can be observed, our method produces robust object segmentations along the sequence. Images (c) and (d) show the frames with lower and higher average pixel error, respectively. The error introduced in (d) 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 (Image (e)) 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 segmenting the object and correcting 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 process is used by the *parachute* sequence to achieve 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.

6. Conclusions

We present a novel technique for video object segmentation based on a formulation of the particle filter in terms of region-based image representation. Our approach is assessed over the SegTrack database producing robust object segmentations and leading to competitive results compared with the state-of-the-art. The code used in this work will be publicly available to encourage reproducible research.



(a) Frame 5 (b) Frame 9 (c) Frame 12 (d) Frame 29 (e) Frame 32 (f) Frame 43

(g) Frame 1 (h) Frame 27 (i) Frame 47 (j) (k) (l)

Figure 3. Qualitative results.

References

[1] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. Pattern Anal. Mach. Intell.*, 33(5):898–916, May 2011.

[2] 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*, 50(2), 2002.

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

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

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

[6] 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*, 115(6), 2011.

[7] A. Bugeau and P. Pérez. Track and cut: simultaneous tracking and segmentation of multiple objects with graph cuts. *J. Image Video Process.*, 2008:3:1–3:14, Jan. 2008.

[8] M. Charikar, V. Guruswami, and A. Wirth. Clustering with qualitative information. In *Foundations of Computer Science, 2003.*, pages 524–533.

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

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

[11] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005.

[12] A. Doucet, S. Godsill, and C. Andrieu. On sequential Monte Carlo sampling methods for Bayesian filtering. *STATISTICS AND COMPUTING*, 10(3):197–208, 2000.

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

[14] N. Gordon, D. Salmond, and A. Smith. Novel approach to nonlinear/non-gaussian bayesian state estimation. *Radar and Signal Processing, IEEE Proceedings F*, 140(2), apr 1993.

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

[16] J. Kwon and F. Park. Visual tracking via particle filtering on the affine group. In *ICIA 2008.*, pages 997–1002, 2008.

[17] Y. J. Lee, J. Kim, and K. Grauman. Key-segments for video object segmentation. ICCV, 2011.

[18] H. Ling and K. Okada. Diffusion distance for histogram comparison. In *CVPR*, 2006.

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

[20] T. Ma and L. J. Latecki. Maximum weight cliques with mutex constraints for video object segmentation. In *CVPR*, pages 670–677. IEEE, 2012.

[21] A. Nakhmani and A. Tannenbaum. Particle filtering with region-based matching for tracking of partially occluded and scaled targets. *SIAM Journal on Imaging Sciences*, 4(1):220–242, 2011.

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

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

[24] D. Tsai, M. Flagg, A. Nakazawa, and J. Rehg. Motion coherent tracking using multi-label MRF optimization. *International Journal of Computer Vision*, 100(2), 2012.

[25] V. Vilaplana and F. Marques. Region-based mean shift tracking: Application to face tracking. In *ICIP*, 2008.

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

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

[28] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Computing Surveys*, 38:no 4, 2006.

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