A region covariance embedded in a particle filter for multi-objects tracking

Hélio Palaio and Jorge Batista*
ISR-Institute of Systems and Robotics
DEEC-FCTUC, University of Coimbra, Portugal

Abstract

This paper present an approach for detection, labelling and tracking multiple objects through both temporally and spatially significant occlusions. The proposed method builds on the idea of object permanence to reason about occlusion. To this end, tracking is performed at both the region level and the object level. At the region level, a particle filter is used to search for optimal region tracks. This limits the scope of object trajectories. At the object level, each object is located based on adaptive appearance models, spatial distributions and inter-occlusion relationships. Region covariance matrices are used to model objects appearance and the dissimilarity between region covariance matrices is used as a new measurement for the particles weight. The regions covariance matrices are updated using a novel approach in a Riemannian space. The proposed architecture is capable of tracking multiple objects even in the presence of periods of full occlusions using a simple and efficient solution for group handling and occlusion reasoning. The results shows the effectiveness of the approach hereby proposed.

1. Introduction

Multiple object tracking with interactions still one of the major challenging tasks to be accomplished on Visual Surveillance Systems (VSS). This task has the purpose of searching possible targets to be tracked in a given scene. To achieve this goal, the system needs to detect the objects, represent and discriminate those objects and robustly track them. This task can be very complicated in realistic scenes, which may contain cluttered backgrounds, unknown number of objects, multiple interactions and occluded objects.

The first issue to solve is the object detection, which can be accomplished through various methods. Yilmaz, et al [15] presents an overview of the object tracking and categorizes the object detection problem in four categories: Point detection - like a Harris Detector; Segmentation - like Graph Cut method; Background Modelling - like Dynamic texture background; Supervised classifiers - like Boosting. For the purpose of this work, and since our goal is to track moving objects, no matter what kind of object, the solution proposed by F. Porikli [7] was adopted, which fits in the third category. That method, which is based on intrinsic images, has the advantages of robustly performing a foreground/background segmentation, even with sudden illumination changes, detects all kind of moving objects and has great final results for our purpose.

A visual-based multi-target tracking system should be able to track a variable number of objects in a dynamic scene and maintain the correct identities of the targets regardless of occlusions or any other visual perturbations. For the tracking system, in order to distinguish different targets and keep their identity over time, it is necessary to treat each individual as a specific object. For specific object representation, salience and uniqueness are the most important characteristics. Objects can be represented by different cues. In Bramble[4] the objects are represented by an appearance base model. A person is modelled as a generalized cylinder whose axis is vertical in the world coordinate frame. In [10] it is used a probability densities of the object appearance. Porikli, et al in [8] purposes an object representation through a region covariance descriptor. The covariance matrices present several advantages as region descriptors, providing a natural way of fusing multiple features.

For the purpose of tracking, particle filtering has been widely used in computer vision and robotics. The particle filter gained its popularity because of its ability to handle highly nonlinear and non-Gaussian models in Bayesian filtering with a clear and neat numerical approximation. The key idea is to approximate the posterior distribution with a set of randomly sampled particles that have weights associated to them. In a standard Bayesian filtering framework, data association is performed to pair the observations and tracks for the evaluation of the likelihood function.

The particle filter framework in our approach handles this level of data association in an implicit way because the covariance matrices are extracted from the regions specified by the particles and the particles weights are updated based on the dissimilarity between those regions covariance matrices and the covariance matrix that represent the last known data of the object. Note that each object has a particle filter associated. It means observations and tracks are no longer independent because observations are conditioned on the particles.

In a scene with multiple objects that interact, the handling of group formation and occlusion reasoning is crucial and challenging. In [9] and [5] it is proposed a method to manage the merge and split of the players in a football match,

*This was funded by FCT Project POSC/EEA-SRI/61150/2004

through a construction of the target interaction graph, linking the players to tracks and then clustering those tracks to obtain the position of each player. This method has assumptions of a fixed number of objects. Our approach to manage this problem is also based in events like merge and split, but without any constraint regarding the number of objects.

Our goal was to develop a complete system which gathered the advantages of the tracking based of the particle filter (PF) with those of the regions covariance descriptors, resulting in a very effective system. In order to improve the system's performance in the merge/split situations which lead to occlusions, we introduce a different approach to manage them. Moreover we introduced a new updated method to the regions covariance matrices in a Riemannian space. This new update method is more effective and computationally less expensive.

So, we can state that this paper presents two major contribution: in a concept level the fusion of the particle filter with the region covariance descriptors is introduced, which gives an effective tracking system even in a very clutter scenarios; and in a more specific level a novel and computational more efficient updated method to the regions covariance matrices is presented.

2. Paper Overview

The paper is organized as follows. Section 3 shows the method for background/foreground segmentation. In section 4 it is presented the object regions descriptor and the new update solution for those descriptors. After that, in section 5 it is introduced the particle filter that will be used in association with the region descriptors. The group formation and occlusion reasoning is shown in section 6. Section 7 presents the results of the whole system hereby proposed. The conclusions are discussed in section 8.

Figure 1 shows a diagram of the method proposed in the present paper. The process start with the segmentation task that feed the image detected objects to particle filter (SIR-s). The particle filter projects the objects particles and compute its weights through the proposed region covariance dissimilarity metric. Then the correspondence matrix of the detected objects and the particle filter estimate is built after that the three managers are applied to handle grouping and occlusions. Finally the objects are updated and a new frame is processed.

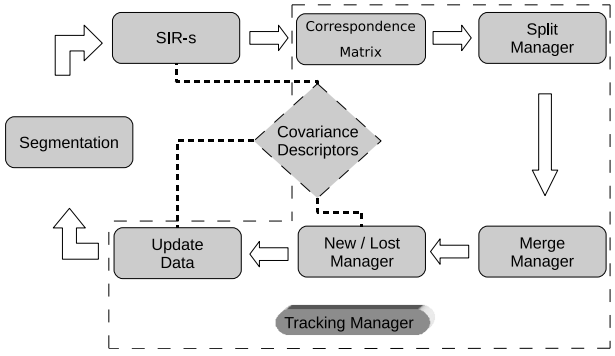

Figure 1. System Diagram

3. Background/Foreground Analysis

To a visual surveillance system, one of the most important step is to have a good process for detection of foreground regions. This process has to deal with changes of lighting and must be computational efficient. The approach here presented is based on the use of intrinsic images following the method proposed by Weiss in [13]. In this process a scene is described as a composition of a static reflectance and a varying illumination,

$$I_t = \mathrm{R}.L_t \tag{1}$$

where R is the reflectance values and L_t is the illumination intensities. It is performed an equivalent formulation in log domain, $i_t = r + l_t$. The lower letters represent the log domain. Following the solution proposed by Porikli in [7] that proposed a similar decomposition that enables evaluation of the motion properties and detection of moving objects in the scene, the image can be decomposed like,

$$I_t = B_t.C_t \tag{2}$$

where B_t is the background which is associated to the static and C_t is the foreground which is associated to the dynamic constituents of the scene. The last formula is mapped to the logarithm domain as $i_t = b_t + c_t$. In real scenes, the static and the dynamic constituents change with time. So this approach considers time-varying intrinsic images, in opposition to the proposed in [13] that assumes the reflectance image has to be independent from the illumination changes.

Let $\{I_{t-kN}, ..., I_{t-k}, I_t\}$ be a set of images from an input set with N images, where k is the sampling period that is adjustable depending on the object motion characteristics. The value of k is set to avoid the overlapping regions between moving object appearances within the consecutive images. Two sets of spatial derivative filters f_n are applied to the images in order to compute the intensity gradients $f_n * i_t$. It is proposed two derivative filters $f_0 = [1{-}1]$ and $f_1 = [1{-}1]^T$. To obtain the maximal likelihood (ML) estimate of static constituent in the transform domain,$\hat{b}_{tn}$ we compute,

$$\hat{b}_{tn} = median_t\{f_n * i_t\}. \tag{3}$$

The dynamic constituent in the transform domain $\hat{c}_{tn}$ is given by

$$\hat{c}_{tn} = (f_n * i_t) - \hat{b}_{tn} \tag{4}$$

At last to recover the time-varying background and foreground images, it is solved the system

$$\begin{array}{l} \hat{b}_t = g * (\sum_n f_n^r * \hat{b}_{tn}) \\ \hat{c}_t = g * (\sum_n f_n^r * \hat{c}_{tn}) \end{array} \tag{5}$$

where f_n^r is the reversed filter of f_n, and g is a filter which satisfies $G = (\mathrm{F}_n.\mathrm{F}_n^r)^{-1}$ in the Fourier transform domain. At this stage it is computed the inversion of the log domain, $B_t = exp(\hat{b}_t)$, $C_t = exp(\hat{c}_t)$.

Figure 2. Segmentation process results - upper left is the original image, upper right is the background image, lower left is the foreground image and the lower right is the mask image.

Now we can define a mask of the foreground region. This is accomplished through a varying threshold, defined as $th_m = 2.5\sigma$, where σ_t^2 and μ_t are the variance and mean of the difference between the Background and Foreground images, $D_t = B_t - C_t$.

$$M_t = \{ \begin{array}{cc} 1 & |D - \mu_t| > 2.5\sigma_t \\ 0 & otherwise \end{array} \quad (6)$$

In figure 2 it is shown the result of this process.

4. Region Covariance Descriptor

The output of the segmentation process gives only the foreground regions. So it is necessary to represent those detected region. To accomplish that task the selection of the right object feature is crucial. Generally, the most desirable property of a visual feature is its uniqueness so that the objects can be easily distinguished in the feature space. Feature selection is closely related to the object representation. The region covariance matrices present several advantages as region descriptors, providing a natural way of fusing multiple features.

Let us start by presenting a method proposed O. Tuzel and F. Porikli in [11] and [8], the region covariance descriptor. Let I be a three dimensional color image, and J a $w \times h \times d$ dimensional feature image, extracted from I

$$J(x, y) = \Phi(I, x, y) \quad (7)$$

where Φ represents the mapping such as intensity, color, gradients, etc. If we define a region R in image J, so that, $R \subset J$ and assuming $\{r_i\}$ be the feature points of R, then this region could be represented by a covariance matrix

$$\mathbf{C}_R = \frac{1}{S-1} \sum_{i=1}^{S} (r_i - \mu)(r_i - \mu)^T \quad (8)$$

where C_R is a $d \times d$ matrix, S the number of points of R and μ the mean of these points.

To have a correct object representation we need to choose what features to extract from I. We define Φ as

$$[x\ y\ I_r\ I_g\ I_b\ I_x\ I_y\ I_{xy}] \quad (9)$$

where x and y are the pixel location in R; I_r, I_g and I_b are the red, green and blue color components of I; I_x and I_y are first derivatives of luminance image of I; and I_{xy} are the laplacian of the the luminance image of I. In this way the region R is mapped into a 8×8 covariance matrix.

4.1. Descriptors Dissimilarity and Update

In a tracking process, the objects appearance changes over time. This dynamic behaviour requires a robust temporal update of the region covariance descriptors and the definition of a dissimilarity metric for th regions covariances. The important question here is: how to measure the dissimilarity between two regions covariance matrices and how to update the regions covariance matrix in the next time slot. Note that the covariance matrices do not lie on Euclidean space. For example, the space is not closed under multiplication with negative scalars. So it is necessary to get the dissimilarity between two covariances in a different space. To overcome this problem a Riemannian Manifold is used.

4.1.1 Riemannian Manifolds

Before continuing our approach, let us introduce a few notions of Riemannian Geometry. A Manifold is a topological space which locally can be seen as an Euclidean space. A Riemannian manifold is a manifold with a Riemannian metric. This allows to generalize notions from Euclidean geometry. The Riemannian metric is a continuous collection of inner products at each tangent space at a point of the Manifold. In general Riemannian Manifolds invariance properties lead to a natural choice for the metric. In the present work we use a metric proposed in [8] which is an invariant metric for the tangent space for symmetric positive definite matrices (e.g. covariance matrices) and is given by

$$< \mathbf{y}, \mathbf{k} >_{\mathbf{X}} = tr(\mathbf{X}^{-\frac{1}{2}} \mathbf{y} \mathbf{X}^{-1} \mathbf{k} \mathbf{X}^{-\frac{1}{2}}) \quad (10)$$

where capital letters denote the points on the Manifold and small letters correspond to vectors on the tangent space, which are also matrices. We refer the readers to [6] for a detailed discussion on Riemannian Geometry with this metric.

4.1.2 Dissimilarity Metric

The dissimilarity between two regions covariance matrices can be given by the distance between two points of the manifold M, considering that those points are the two regions covariance matrices.

That distance on a Manifold M is the length of the curve with the minimum length which connects them. This curve lives on a geodesic. Let $\mathbf{y} \in T_{\mathbf{x}}M$, where $T_{\mathbf{x}}M$ is the tangent space at point $\mathbf{X} \in M$. There is a unique geodesic starting at $\mathbf{X}$ with tangent vector $\mathbf{y}$. The exponential map, $exp_{\mathbf{X}}$:

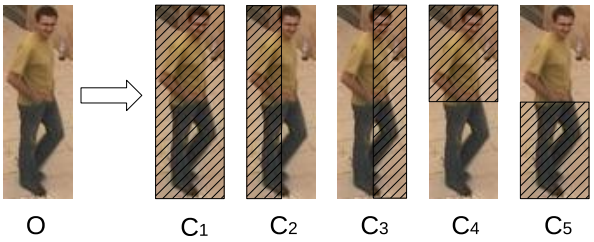

Figure 3. Object representation. Construction of the five covariance matrices from overlapping regions of an object feature image.

$T_{\mathbf{X}}M \mapsto M$, maps the vector $\mathbf{y}$ to a point $\mathbf{Y}$ belonging to the previous geodesic. We denote by $log_{\mathbf{X}}$ its inverse. The distance between $\mathbf{X}$ and $\mathbf{Y}$ is given by $d^2(X,Y) = \|y\|^2_{\mathbf{X}}$. Like above we use the exponential map proposed in [8] with the same metric,

$$\mathbf{Y} = exp_{\mathbf{X}}(\mathbf{y}) = \mathbf{X}^{\frac{1}{2}} exp(\mathbf{X}^{-\frac{1}{2}}\mathbf{y}\mathbf{X}^{-\frac{1}{2}})\mathbf{X}^{\frac{1}{2}} \tag{11}$$

$$\mathbf{y} = log_{\mathbf{X}}(\mathbf{Y}) = \mathbf{X}^{\frac{1}{2}} log(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}})\mathbf{X}^{\frac{1}{2}} \tag{12}$$

If we use the definition of the geodesic distance and substituting (12) into (10) we have,

$$\begin{aligned} d^2(\mathbf{X},\mathbf{Y}) &= < log_{\mathbf{X}}(\mathbf{Y}), log_{\mathbf{X}}(\mathbf{Y}) >_{\mathbf{X}} \\ &= tr(log^2(\mathbf{X}^{-\frac{1}{2}}\mathbf{Y}\mathbf{X}^{-\frac{1}{2}})) \end{aligned} \tag{13}$$

4.1.3 Regions Covariance Matrix Update

A solution for the covariance matrices update was proposed in [8], that is based on the estimation of the points mean on a Riemannian Manifold, where each point corresponds to a covariance matrix. This mean estimation is obtained using a gradient descent approach. In this paper, we propose a novel solution for the covariance matrix update, that is based on the mean of the new covariance matrix and the last covariance updated. If $\mathbf{y}$ is the velocity that takes us from X to Y, $\mathbf{y}/2$ will take us half the distance to point $\overline{\mathbf{C}}$. Using equations (11) and (12), we have

$$\begin{aligned} \overline{\mathbf{C}} &= \mathbf{X}^{\frac{1}{2}} exp(\mathbf{X}^{-\frac{1}{2}}(\tfrac{1}{2}\mathbf{y})\mathbf{X}^{-\frac{1}{2}})\mathbf{X}^{\frac{1}{2}} \\ &= \mathbf{X}^{\frac{1}{2}} exp(\tfrac{1}{2} log(\mathbf{X}^{-\frac{1}{2}})\mathbf{Y}\mathbf{X}^{-\frac{1}{2}}))\mathbf{X}^{\frac{1}{2}} \end{aligned} \tag{14}$$

which after some mathematical simplification turns into,

$$\overline{\mathbf{C}} = (\mathbf{X}^{\frac{1}{2}}\mathbf{Y}\mathbf{X}^{\frac{1}{2}})^{\frac{1}{2}} \tag{15}$$

where $\overline{\mathbf{C}}$ is the average distance between two points on a Riemannian Manifold (the updated covariance matrix). This update means that the present covariance is more important than the previous covariances. Since we are tracking objects that can change over time, the last information about them is more reliable.

4.2. Improvement to Occlusion

One way to improve the capacity of matching even with occlusions is proposed by Tuzel, et al [11] and consists in representing the object region with five regions covariances matrices. Figure 3 it is shown the regions per each covariance. If one half of the image is occluded and has a bad match, the integration of the others regions will produce better results. The covariance matrices are computed as described above and the dissimilarity between two objects is, now, defined by

$$\rho(O_1, O_2) = \sum_{i=1}^{5} d^2(\mathbf{C}_i^{O_1}, \mathbf{C}_i^{O_2}) \tag{16}$$

where $\mathbf{C}_i^{O_1}$ and $\mathbf{C}_i^{O_2}$ are the five regions covariance matrices of object 1 and object 2, respectively.

5. Particle Filter

Defined the objects descriptor, it is necessary to find an approach to track such objects. So, in order to reduce the search areas and to have an accurate estimate of the object location, it is used a filtering with prediction approach. The Extended Kalman Filter (EKF) is one of the most known approaches to the non-linear filtering problem, as we can see by Welch and Bishop in [14]. However, this method does a linearization through a Taylor expansion and using the first term. In opposition, we have the particle filter (PF) that is a truly nonlinear filter.

The particle filtering method allows Bayesian estimation to be carried out approximately but in a structured manner. The objective is to compute the posterior PDF when the situation does not yield an analytical form. Particle filtering has been a successful numerical approximation technique for Bayesian sequential estimation with non-linear, non-Gaussian models. Our approach is based on the bootstrap filter [3] [2].

Let $\{\mathbf{x}_t^{(1)}, ..., \mathbf{x}_t^{(N_s)}\}$ and $\{\mathbf{z}_t, ..., \mathbf{z}_t\}$ be, respectively, the samples and the observations up to time. The particle filter approximates the posterior distribution $p(\mathbf{x}_t|\mathbf{z}_{1:t})$ by a set of weighted samples $\{\mathbf{x}^{(i)}, w^{(i)}\}_1^{N_s}$. In our solution, the state estimate, $\hat{\mathbf{x}}_t$, is approximate by the sample with the higher weight,

$$\hat{\mathbf{x}} = \mathbf{x}^j|_{w^j = max(w)}. \tag{17}$$

The weights are computed based on the dissimilarity equation (16),

$$w^{(i)} = exp(-\rho(C(\mathbf{x}^{(i)}), \overline{\mathbf{C}})). \tag{18}$$

where $C(\mathbf{x}^{(i)})$ represents the five covariance matrices in region centred at $\mathbf{x}^{(i)}$ and $\overline{\mathbf{C}}$ the mean covariance matrix that represents the tracking target. To simplify the notation we will refer to the five covariance matrices only as $\mathbf{C}$. Since the object may change rapidly, no posterior information is available about the size of the region object, so it is performed a

search in five different scales based in the last size known. Therefore it is necessary to change equation 18, that will be given by

$$w^{(i)} = max_j(exp(-\rho(\mathbf{C}^j(\mathbf{x}^{(i)}), \overline{\mathbf{C}}))). \quad (19)$$

where $\mathbf{C}^j$ is the covariance matrices at scale j which is the index of scale, $Scale = \{100\%, 109\%, 118\%, 92\%, 86\%\}$.

The motion between two instants is described by the transition model. The ideal model has the exact kinematics of the object movement. However, that is not possible in practice, so approximated models are used. The most common is a fixed constant-velocity model with a fixed noise variance: $\mathbf{x}_t^{(i)} = \mathbf{x}_{t-1}^{(i)} + \mathbf{v} + \xi$, where $\mathbf{v}$ is the velocity of the system and $\xi : \mathbf{N}(0, \Sigma)$ is a Gaussian noise. This kind of model presents some problems: the difficulty of a correct first velocity estimate; the dynamics usually don't follow a constant velocity model; and the trajectory changes during time.

In [16] by Zhou, et al it is proposed an adaptive model for a particle filter. Our approach is based on a adaptive velocity model. As time runs it is possible to compute a more accurate estimate for the true velocity of the system as the average velocity of the last k instants,

$$\mathbf{v}_t = \frac{1}{k} \sum_{n=t-k}^{t} |\mathbf{x}_n - \mathbf{x}_{n-1}|. \quad (20)$$

6. Tracking Manager

At this stage it is important to clearly define the concept of an object and how it is integrated with the particle filter. An object can be of a single nature or a group object and represents an image tracked target. It is represented by the descriptor $O_n = [X, Y, S_n, N, L, Id]$, where X and Y are the target centre of mass coordinates, S_n is the PF filter parameters, N the number of targets associated to the object n, L is a list of pointers to the N object descriptors that form the group object ($N > 1$) and Id is the target label.

The descriptor $S_n = [Sz, \overline{\mathbf{C}}, \{w^i\}_1^{N_s}, \{\mathbf{x}^i\}_1^{N_s}, N_s, \mathbf{v}]$ is composed by: Sz the dimension of the target bounding box; $\overline{\mathbf{C}}$ the mean covariance matrices of the target; w^i the weights of particle i; $\mathbf{x}^i$, which is the vector with the coordinates of the particle i; the number of particles N_s; and the velocity of the model, $\mathbf{v}$.

6.1. Single Object Tracking

Let us consider the simple situation of one object tracked with the object descriptor $O_1(t)$ at time frame t. The estimate of the object mass centre coordinates in the next instant is given by $[\hat{X}, \hat{Y}]^T = \mathbf{x}^j|_{w^j = max(w)}$. Then the object is associated to the detected blob given by the segmentation process. If the estimated coordinates are inside that blob the object descriptor is updated and $O_1(t+1) = \hat{O}_1(t)$.

	F_1	F_2	...	F_M	CM_l
$\hat{O}_1$	1	0	0	1	2
$\hat{O}_2$	0	1	0	0	1
...	...	...	...	...	...
$\hat{O}_N$	0	1	0	0	1
CM_c	1	2	0	0	

Signal	Event
$CM_l > 1$	split
$CM_l = 0$	lost
$CM_c > 1$	merge
$CM_c = 0$	new
$CM_l = CM_c = 1$	stable

Table 1. Left- Correspondence Matrix; Right- Event table

6.2. Grouping and Occlusion Manager

When we have multiple objects to track, it is possible that there exist various candidates for a single blob, which results in a merge, or group objects that split. Our solution to handle this problem is based in a event manager but without constraint regarding to the number of objects, unlike the method proposed by P. Nillius, et al in [5].

Let assume that we have N objects $\hat{O}_i$ with the position estimate given by equation (17) and M detected blobs F_j which are the foreground images. The detected blobs will also be called as targets. At a frame time t, to disambiguate the problem, the first step is to build a correspondence matrix between $\hat{O}_i$ and F_j. The correspondence matrix (CM) is a $N \times M$ matrix, defined as follows

$$CM(i,j) = Blgs(\hat{O}_i, F_j), \ \forall_{i \in 1...N, j \in 1...M} \quad (21)$$

where $Blgs$ returns 1 if $[\hat{X}_i \hat{Y}_i]^T \subset F_j$ and returns 0 otherwise. Defining $CM_l(i) = \sum_{j=1}^{M} CM(i,j)$, $CM_c(j) = \sum_{i=1}^{N} CM(i,j)$ our CM will be in the form

Now, we define events based on the cardinality of CM_l and CM_c. Table 1 shows how the event appears. As far as the merging is concerned, the algorithm has to deal with occlusion and grouping. Its goal is to manage the paths of group objects and single objects, applying the dynamic model if occluded or updating otherwise. For splitting events, the algorithm has to disambiguate which objects are associated to different targets.

Based on the CM, four managers, running in cascade, were used to handle the image objects: split manager, merge manager, new/lost manager and update manager.

Three lists of objects were considered, the active list, occluded list and lost list corresponding, respectively, to the visibly objects, occluded objects and objects that were not detected for a while.

6.2.1 The Split manager

When a split event is detected, there are two possible situations that may occur: a group object split or a single object

split.

The first case is the most common one. The single objects that compound the group object are detected in more than one target (F_j). Then the split manager pass the group object to an inactive state (lost list) and those single objects passes to the active list. The CM is rebuilt.

To handle the combined event split/merge it is necessary to ensure that for a group object $\hat{O}_i$ that was associated to F_j, the single objects referenced in L, must be inside the target F_j. Otherwise, it occurs a split and merge. If it is the case, the group object passes to the lost list and the single objects referenced in L return to the active list. The CM is rebuilt.

Regarding a single object split, new objects are created and added the active list.

6.2.2 The Merge manager

When a merge situation is detected, a group object is created in the active list. In this situation the single objects that make the group object will go to the occluded list, the parameter L is filled with the Id of the single objects and the value N is set up.

6.2.3 New/Lost manager

When a new event is detected, it may take place one of two situations: there is a new object or it is a miss split. In the second case, it is performed a search for a group object near the new possibility. If a group object is detected, it is performed a match test between the single objects that compound it and the region of the new possibility. If a single object matches, it is associated to that region and the CM is rebuilt. Otherwise it is considered a new single object.

If it is detected a lost event, it means that an object could have really disappeared or it may be a single object that belongs to a group with a bad position estimation. If so, it means that the position estimation was made based only on the dynamic model of the filter. When that happens the position is set equal to the last known position and the group object passes to the active list. The CM is rebuilt and the split and merge manager are called again. If this is not the case, the object descriptor goes to the lost list.

6.2.4 Updating manager

The update manager just updates the state of the objects. The single objects are updated directly with the estimation $\hat{O}_i$. The group objects, before doing their own update, search for all single objects that compose them and, if not occluded, perform the single object update based on the position estimate, otherwise the update is performed based only on the equation of the dynamic model. The object is considered occluded when $max(w) < th$, where w are the particles weight of the PF associated to the object and th is a defined threshold.

7. Results

In order to correctly evaluate the new update method and the fusion of the particle filter with the region covariance descriptors, we start by evaluating the proposed update method with the one previously proposed in the literature. After that we showed the performance of the system in different scenarios. We started by testing it in a two people crossing scenario, with excellent results, followed by a four people interacting scenario, and finishing with a five people scenario with multiple crosses. Figure 6 shows the results of the five people scenario. In this case the human figures pass by one another several times occluding each other. It is a very challenging scenario. Nevertheless, the method here proposed is ultimately able to correctly track and label all persons. We also tested the system when tracking people's faces and vehicles in a highway.

7.1. Update method

To evaluate the robustness of the proposed update solution we compare the result of it and the ones obtained by the Porikli update proposed in [8] with a ground-truth dataset. The objects position of this dataset was manually set. The results were obtained by just changing the update method, so the two methods were tested in the same conditions.

In graphic 4 and table 2 we compare the RMS error and the percentage of correct labelling, respectively, of the new update method and the one proposed by Porikli, et al in [8]. These results were obtained in the five people scenario. Concerning to the RMS error, the two updates are equivalent. Relatively to the percentage of correct labelling, the new update method presents better results with a correct labelling when grouped of 75% against the 66% obtained by the Porikli update method.

Another advantage of this new update method is the time execution. In table 3 we show the results in milliseconds of the two updates methods. The Porikli updates time execution was measured considering a stack of five regions covariance matrices. Both methods were implemented and tested in *Matlab*. In this chapter the new update is much faster than the one proposed in [8], with an average performance of $3.6ms$.

	Labelling when Single	Labelling when Grouped	Correct Grouping	Correct Splitting
Porikli Update	100 %	66 %	9/12	11/12
New Update	100 %	75 %	9 / 10	9 / 10

Table 2. Correct labelling results - Five people scenario

	Execution time (ms)
Porikli Update	837.5
New Update	3.6

Table 3. Update methods execution time in milliseconds

7.2. System Performance

To evaluate the performance of the system hereby proposed we tested it in different scenarios. So we evaluate the

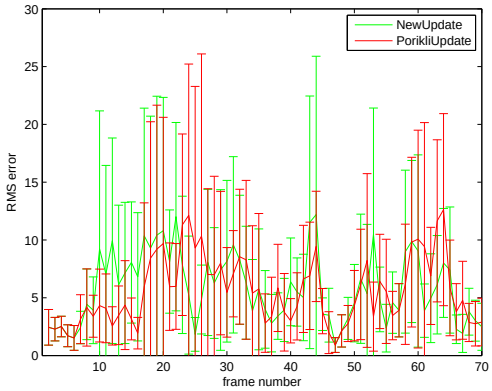

Figure 4. RMS error and Confidence Intervals: Green - New Update; Red - Porikli Update

system in a five people scenario, a highway scenario and its behaviour when tracking people's faces. To detect the faces we used the face detect approach based on the haar-like features initially proposed by P. Viola and M. Jones in [12], instead of the segmentation approach of section 3. We had to do that because we do not want the whole of the moving objects, but only a part of them, the face. In figure 7 it is shown a four people scenario (data set provided in AVSS 2007 [1]). Once more we can see that the system ultimately is able to label the objects correctly. We can also perceive that in some frames of this scenario occurs a bad estimate for the occluded objects. However, when the object reappears that estimate is corrected. Between frame 55 and 65 we can see how the system leads with the exit of an object and in frame 75 a new object is detected. The object is the same but it was long time out of the image and was considered lost.

Table 4 shows the system's performance in the different scenarios. When tracking faces the system presents worse results than in the other scenarios. Nevertheless it is capable of doing a correct labelling when the object splits. We can also see that no matter which scenario, our approach always does a correct labelling when the objects are not grouped. Figure 5 shows the trajectories of the five people scene objects. Each object is represented by a color where the dashed lines mean a labelling when the object was grouped, the dots means that it was occluded (trajectories are based only in the dynamic model) and the continuous line means a labelling when the object was alone. In the black object trajectory it is possible to verify that when grouped it had some bad labellings.

Figure 5. Object trajectories.

Scene	Labelling when Single	Labelling when Grouped	# Frame
Highway	100 %	71 %	410
Five people	100 %	75 %	90
Faces	100 %	55 %	150

Table 4. Correct labelling in different scenarios

8. Conclusions

We proposed a complete system for detection, labelling and tracking multiple objects. A particle filter was introduced with the dissimilarity between covariance matrices as a new measurement for the particle weights. The performance of the system proved it effectiveness even in a very clutter scene with multiple occlusions and in different scenarios with different objects to track. So, it has the advantage of tracking all kind of moving objects in scene.

In order to update those matrices, a novel solution was proposed in a Riemannian space. The results prove that this update method is computational less expensive than the one proposed by F. Porikli [8] and obtain with a similar performance.

References

[1] Advanced video and signal based surveillance 2007 - datasets.
[2] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for on-line non-linear/non-gaussian bayesian tracking. In *IEEE Transactions of Signal Processing*.
[3] A. Doucet, N. De Freitas, and N. Gordon, editors. *Sequential Monte Carlo methods in practice*. 2001.
[4] M. Isard and J. MacCormick. Bramble: a bayesian multiple-blob tracker. In *ICCV*, 2001.
[5] P. Nillius, J. Sullivan, and S. Carlsson. Multi-target tracking - linking identities using bayesian network inference. In *CVPR*, 2006.
[6] X. Pennec, P. Fillard, and N. Ayache. A riemannian framework for tensor computing. In *IJCV*, 2006.
[7] F. Porikli. Multiplicative background-foreground estimation under uncontrolled illumination using intrinsic images. In *IEEE Computer Society Workshop on Motion and Video Computing*, 2005.
[8] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on means on riemannian. In *CVPR*, 2006.
[9] J. Sullivan and S. Carlsson. Tracking and labelling of interacting multiple targets. In *9th European Conf. Comput. Vision ECCV*. IEEE Computer Society, 2006.
[10] H. Tao, H. S. Sawhney, and R. Kumar. A sampling algorithm for tracking multiple objects. In *Workshop on Vision Algorithms*, 1999.
[11] O. Tuzel, F. Porikli, and P. Meer. Region covariance: A fast descriptor for detection and classification. In *ECCV*, 2006.
[12] P. Viola and M. Jones. Robust real-time object detection. 2002.
[13] Y. Weiss. Deriving intrinsic images from image sequences. In *ICCV*, 2001.
[14] G. Welch and G. Bishop. An introduction to the kalman filter. Technical report, 1995.
[15] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4), 2006.
[16] S. K. Zhou, R. Chellappa, and B. Moghaddam. Visual tracking and recognition using appearance-adaptive models in particle filters. In *Image Processing, IEEE Transactions*, 2004.

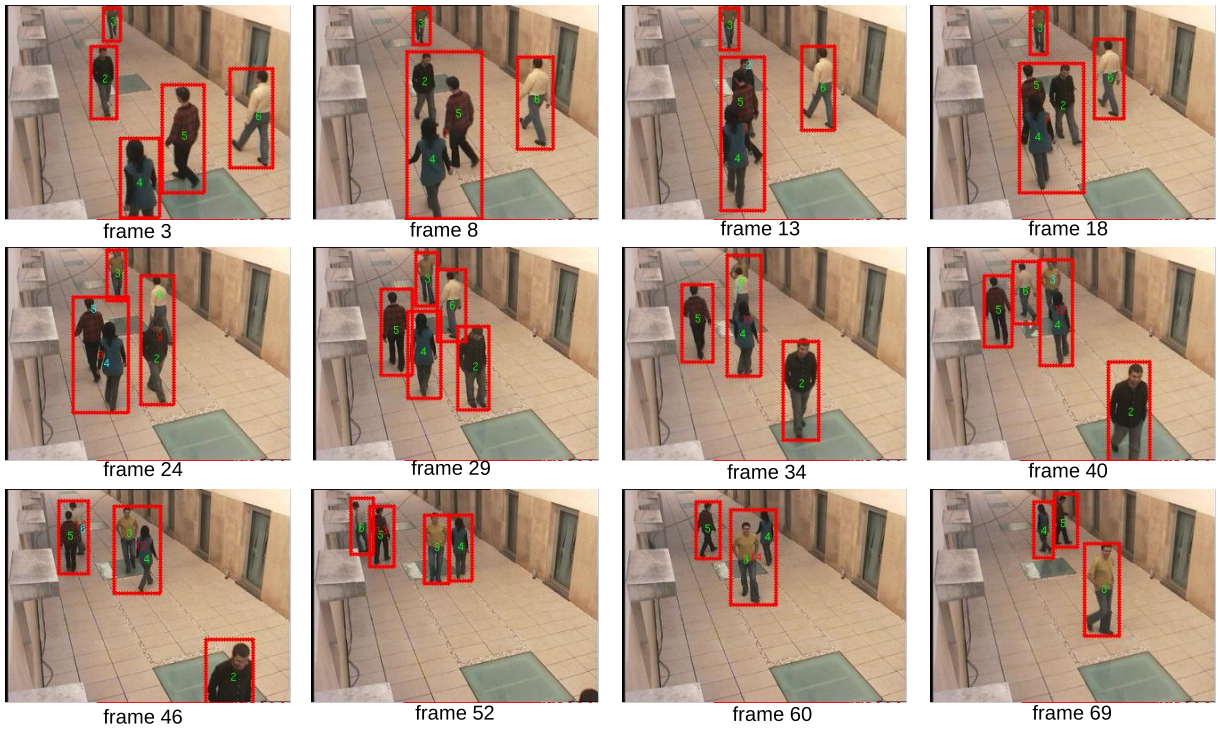

Figure 6. Results of a scene with five people that occludes each others; the red numbers means a group object; the single objects are represented with the green numbers; the cyan numbers means a estimate based only on the model dynamics, the red rectangles shows the blob region.

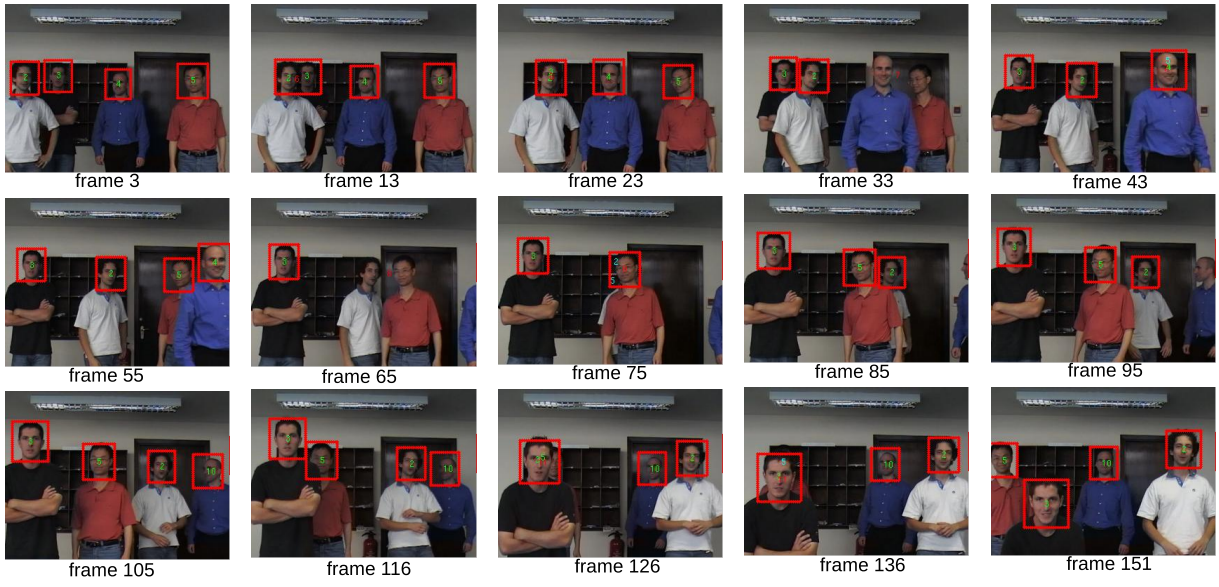

Figure 7. Results of a face tracking in a scene with four people that occludes each others. The color code is the same as in figure 6. Data set available in [1]