People appearance tracing in video by spectral graph transduction

Dalia Coppi, Simone Calderara, Rita Cucchiara
DII - University of Modena and Reggio Emilia
Via Vignolese 905 - Modena - Italy
{name.surname}@unimore.it

Abstract

*Following people in different video sources is a challenging task: variations in the type of camera, in the lighting conditions, in the scene settings (*e.g. *crowd or occlusions) and in the point of view must be accounted. In this paper we propose a system based only on appearance information that, disregarding temporal and spatial information, can be flexibly applied on both moving and static cameras. We exploit the joint use of transductive learning and spectral properties of graph Laplacians proposing a formulation of the people tracing problem as a semi-supervised classification. The knowledge encoded in two labeled input sets of positive and negative samples of the target person and the continuous spectral update of these models allow us to obtain a robust approach for people tracing in surveillance video sequences. Experiments on publicly available datasets show satisfactory results and exhibit a good robustness in dealing with short and long term occlusions.*

1. Introduction

To ensure public security, an increasing number of surveillance cameras have been placed in public areas over the recent years. As a result, the increased amount of data has lead to the adoption of automatic systems of video analysis both for real time surveillance and for the a-posteriori mining and reasoning on the extracted security information. In this way, the great variety of video data provided by installed cameras are automatically analyzed for event detection, object and people tracking, action understanding offering a valid support to investigations and crime detection. Among others, people tracking has reached the most attention: it is the first necessary task to keep a consistent identification of detected people and, at the same time, it exhibits an objective difficulty in modeling the human body appearance and motion. Despite numerous algorithms have been proposed and reached good results in constrained environments, the problem is still not completely solved in complex scenarios, which may include moving cameras, illumination variations or crowded scenes.

A comprehensive survey on object tracking has been drawn by Yilmaz *et al* [15] where a categorization of the different state-of-the-art methods is provided with a description of the most representative approaches for each category.

Conventional tracking systems [17, 9, 8] exploit temporal and spatial information to estimate the optimal target position in the image given its position in the previous frame and search for the best match using appearance information. However, when data are time-lapsed, *i.e*. the frame rate is not constant due to bandwidth saving, or when data is provided from different or moving cameras, motion estimation is invalidated and tracking systems tend to fail or to show decreasing performances. Security data footages often exhibit these characteristics and therefore it is of growing importance to consider tracing instead of tracking. *Tracing algorithms* [12, 7], usually based on color and texture information of the person, have the purpose to follow the traces of the target when no motion and position estimation is available.

We propose a tracing system where only the people appearance is used allowing to deal with different types of camera (*e.g*. fixed or PTZ) and various background conditions, since no motion information is used and no background-foreground segmentation is performed. The bounding boxes of people in the scene are extracted by a people detector and provided to the system as a set of labeled and unlabeled elements reinterpreting tracing as a *Semi Supervised Graph based problem*.

Semi-supervised learning (SSL) refers to a subset of learning techniques that exploit both labeled and unlabeled data for training a classifier. In the recent years SSL has been used in many areas including computer vision where several applications *i.e*. image segmentation [4] or object recognition and tracking [16] are solved with a semi-supervised classification. The reason for this choice can be explained observing that in many real world applications it is relatively easy to acquire a large amount of unlabeled (unclassified) data whereas it is quite difficult and expensive to have labeled, or classified, data. In surveillance and foren-

2011 IEEE International Conference on Computer Vision Workshops
978-1-4673-0063-6/11/$26.00 ©2011 IEEE

sic tasks visual data is easily acquired by installed cameras but their classification and annotation require a slow human intervention. For this reason we propose to provide the system with two initial sets of labeled elements classified as *positive* when belonging to the target person and as *negative* otherwise. Thus, the tracing objective on the new snapshots (iteratively extracted on subsequent frames) is automatically reached by the classification algorithm.

The main novelty of this proposal is that we do not define a complete automatic tracking system with given initial constraints. Instead, we propose a semi-automatic approach where users analyzing the people snapshots detected in initial video frames provide some (very few) initial examples of correspondence allowing the transductive learning system to be trained and to continue in the current video and other available video to trace the same person, also learning different appearances due to light, scene and point of view changes. Working without motion prediction it can be used in very general contexts; when only few frames (time unrelated) are available, when large occlusions and crowd make detection a difficult task so that the person is extracted in few frames only; when camera is moving or video are acquired by different cameras. Differently from tracking approaches, where single or multiple target positions are predicted and then predictions and observations are compared, in this SSL approach prediction is not exploited but the single target is evaluated against all the available targets detected at each frame, by looking for appearance similarity. This is neither an approach of content-based retrieval where some examples are given and similar ones are searched in the dataset, since here the possibility of analyzing continuous frames (even with some discontinuities) allows the system to learn and remember the changes in appearance.

Our proposal includes some innovative points: (i) adoption of people detection and transductive learning with covariance matrix descriptors for appearance similarity; (ii) definition of a new transductive models that accounts both positive and negative examples; (iii) definition of a model updating mechanism based on spectral graph theory that avoids drifts due to classification errors.

2. System overview

Fig. 1 depicts the overall scheme of the proposed system. The first operation is devoted to the **detection engine**. Since we designed the system to be flexibly applied on both, moving and static camera scenarios, we decided to discard conventional Background Subtraction methods for detecting moving people. Consequently, we adopt in every frame a people detector to extract the snapshots of people in the scene. Once extracted, snapshots are described in a Riemannian fashion using the covariance matrix descriptor as detailed in the next section. Therefore, an initial data association of few frames is needed. This can be made manually as in typical forensics applications or, if possible, automatically using a conventional tracking system (*e.g.* [14]). Once some examples of the target have been provided, the **tracing engine** searches for the target in the subsequent frames. The tracing engine exploits semi supervised learning (SSL) to detect among the candidates the positive samples of the target objects, if present. Differently from previously proposed methods [16, 1], this transductive learning approach based on Riemannian space uses both positive and negative examples to classify new elements.
Since the target appearance can evolve in time, we designed the **update engine** to retain the best target images and capture most of its appearance variability. The presented system can be employed both for on-line surveillance tasks and off-line crime detection in forensics. In the first case the method is performed in real-time as the visual data is provided by surveillance cameras, while in the second possibility a posteriori for the analysis of previously stored video sequences is performed.

3. Detection Engine for people extraction

The system we propose is independent from the specific people detection method and relies only on the set of people snapshots extracted from each frame of the video and on the initial models of labeled elements. We decide to extract people on the scene using one of the state-of-the-art people detector based on Histograms of Oriented Gradient [2], which have been demonstrated in [3] to be one of the most reliable method to detect people in surveillance scenarios. Many others can be applied without limitations.

To represent each snapshot and match different image region of the same person a predefined feature should be used with a specific metric capable to provide a reliable comparison between snapshots. We implemented a snapshot similarity measure based on covariance matrix features descriptor, [13]. The same metric has been previously adopted in [12] or [10] because of its robustness in capturing shape, location and color information. The reason of covariance matrix usage in object representation is based on the fact that generally a matrix extracted from a single region is enough to match the region in different views and poses, because the noise corrupting individual samples is largely filtered out with the average filter during covariance computation. Moreover, covariance matrices have scale and rotation invariance property and are independent to the mean changes such as identical shifting of color values.
The covariance matrix is a square symmetric matrix $d \times d$, where d is the number of selected features independently from the size of the image window, carrying the advantage of being a low dimensional data representation. Given the covariance matrix C its diagonal entries represent the variance of each feature and the non-diagonal entries represent

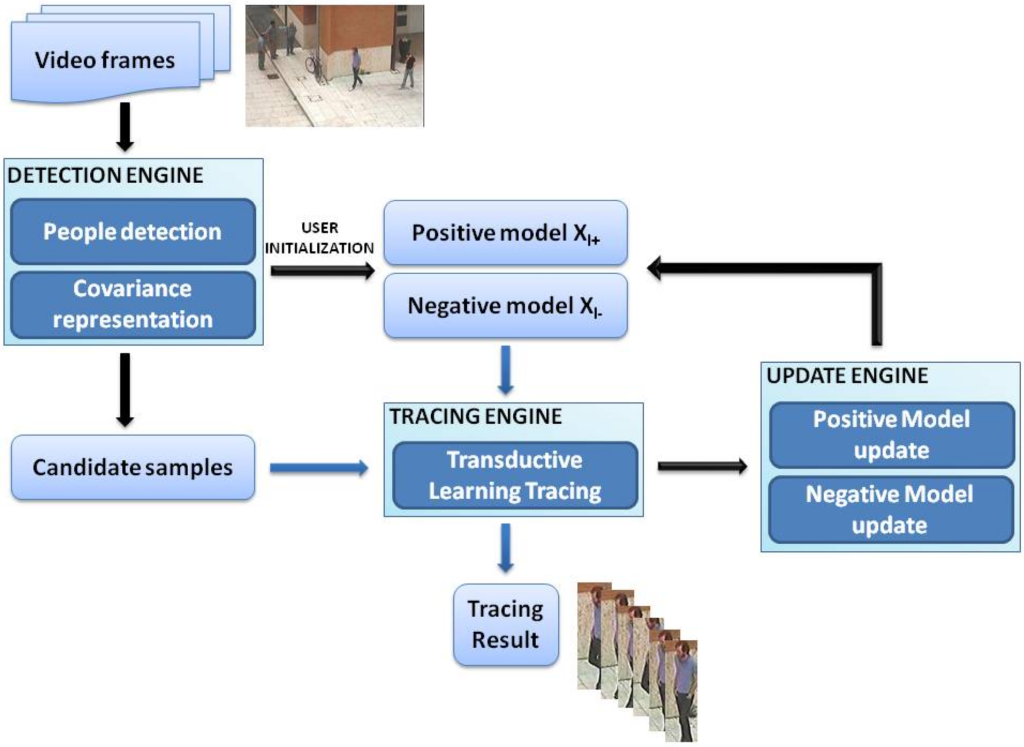

Figure 1. System overview

the correlations.

Considering I as a three-dimensional color image and F as the $W \times H \times d$ dimensional feature image extracted from I,

$$F(x, y) = \Phi(I, x, y) \quad (1)$$

where the function Φ can be any mapping such as intensity, color, gradients, filter responses, etc. Let $\{z_i\}_{i=1...N}$ be the d-dimensional feature points inside F, with $N = W \times H$. The image I is represented with the $d \times d$ covariance matrix of the feature points:

$$C_R = \frac{1}{N-1} \sum_{i=1}^{n} (z_i - \mu)(z_i - \mu)^T \quad (2)$$

where μ is the vector of the means of the corresponding features for the points within the region R.

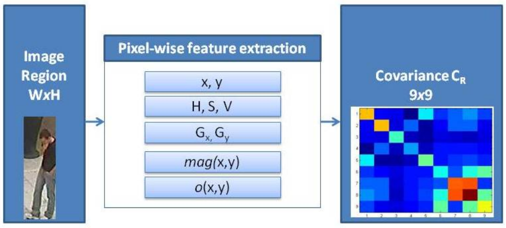

Figure 2. Covariance matrix computation.

In our case z_i is the feature vector composed for each pixel by its spatial, color and edge information. We use x and y pixel location in the image grid, HSV color values, G_x and G_y first order derivatives of the intensities calculated through Sobel operator w.r.t. x and y, and the magnitude $mag(x, y) = \sqrt{G_x^2 + G_y^2}$ and the angle $o(x, y) = arctan\left(\frac{G_y}{G_x}\right)$ of the first order derivatives. Therefore each pixel of the image is mapped to a nine-dimensional feature vector

$$z_i = [\, x \;\; y \;\; H \;\; S \;\; V \;\; G_x \;\; G_y \;\; mag(x, y) \;\; o(x, y) \,]^T \quad (3)$$

Based on this features vector the covariance of a region is a 9×9 matrix.

It should be noted that we use HSV color space instead of the basic RGB color space because we experimented an higher invariance to scale and light changes of the HSV components with respect to the RGB ones.

A part from the adopted feature vector, an adequate distance between covariance matrices must be defined to assess the appearance similarity between candidates regions and the target. However, the covariance matrices do not lie on the Euclidean space and arithmetic subtractions or simple operations between matrices are not correct. A robust distance metric between the covariance matrices is proposed in

[5] as the sum of the squared logarithms of the generalized eigenvalues:

$$\rho(C_i, C_j) = \sqrt{\sum_{i=1}^{d} \ln^2 \lambda_k(C_i, C_j)} \quad (4)$$

where $\lambda_k(C_i, C_j)_{k=1\dots d}$ are the generalized eigenvalues of C_i and C_j computed as:

$$\lambda_k C_1 x_k - C_j x_k = 0 \;\; k = 0 \dots d \quad (5)$$

where x_k are the generalized eigenvectors. The distance measure ρ satisfies the metric axioms, positivity, symmetry, triangle inequality, for positive definite symmetric matrices.

4. Tracing Engine based on graph transduction

As introduced above we exploit a Semi Supervised classification algorithm to classify people snapshots as belonging or not to the current target. The general configuration of a SSL algorithm provides a data set $X = (x_i)_{i \in n}$ that can be divided in two parts: the elements $X_l = (x_1, \dots, x_l)$ with associated labels $Y_i = (y_1, \dots, y_l)$, and the elements $X_u = (x_{l+u}, \dots, x_u)$ the labels of which are not known and should be derived by the learner.

Similarly to [1, 6], we adopt a transductive learning algorithm based on spectral graph theory, but defined in conjunction with covariance matrices and, differently from the previous works, we propose the exploitation of both positive and negative examples.

Considering to have a set of labeled instances X_l subdivided in two sets, namely the positive labeled model X_{l+} with $y_i = +1$ built with multiple instances of the target object we want to trace, and the negative labeled model X_{l-} with $y_i = -1$ corresponding to the labeled istances that differs from the target. The complete dataset comprises both the model X_l and the candidates samples X_u and their associated label function y_i that takes non-zero values for the elements belonging to the model and zero value otherwise.

$$D(X, Y) = \{X_l \cup X_u, Y : y_i = \pm 1 \text{ iff } x_i \in X_l\} \quad (6)$$

By setting the problem in this form we aim at propagating for every frame the knowledge encoded into the model that can be equivalently interpreted as the problem of estimating the missing label function values.

The transductive learning is performed iteratively on each frame. At each iteration the input of the algorithm is constituted by the training labels Y_l , the two models X_{l+} and X_{l-} and the samples extracted in the current frame in the form of an affinity undirected graph $G = (V, E)$ where the nodes represent the samples and the edges are proportional to the affinity between them. Conceptually, exploiting the spectral properties of the graph and finding a cut of the graph that separates positive and negative elements, the algorithm aims at finding a classification for the new elements. The result is in the form of labels for unlabeled elements and their values are representative of the confidence of each element to belong to the traced object.

4.1. The algorithm for a spectral transducer on graphs

Let $G = (V, E)$ be an undirected similarity-weighted k nearest-neighbour graph over the input space X, and let A be its adjacency matrix. A is simmetricized by $A = A' + A'^T$ with:

$$A'_{i,j} = \begin{cases} \dfrac{w_{i,j}}{\sum\limits_{k \in KNN(x_i)} w_{i,k}} & \text{if } x_j \in KNN(x_i) \\ 0 \text{ otherwise} \end{cases} \quad (7)$$

where $w_{i,j}$ is an exponential symmetric function proportional to the distance $\rho(x_i, x_j)$ between samples x_i and x_j, $w_{ij} = \exp\left(-\dfrac{\rho(x_i, x_j)}{\sigma^2}\right)$ with σ a regularization parameter. We also include some constraints on A by setting to 0 the affinities among unlabeled elements, i.e. , the candidates patches extracted from the current frame. Indeed for each frame only one of the elements can belong to the positive samples and by making their subgraph non-connected we avoid self-loops that could lead to misclassification.

Following the postulates in [6] for the design of a transductive learner, the problem of finding a labeling for the test examples X_u can be solved by minimizing a classification error function or equivalently optimizing the problem:

$$\begin{aligned} &\max_y \sum_{i=1}^{n} \sum_{j \in kNN(x_i)} y_i y_j A_{ij} \text{ s.t.} \\ &y_i = +1 \text{ if } x_i \in X_{l+} \\ &y_i = -1 \text{ if } x_i \in X_{l-} \\ &y_j \in \{0, 1\} \end{aligned} \quad (8)$$

When A is considered as the adjacency matrix of a graph G, the problem in Eq.8 is equivalent to finding the cut of the graph G that separates the two subgraphs G^+ constituted by the set of examples (*i.e.* vertices) with $y_i = +1$ and G^- constituted by the set of examples with $y_i = -1$. Although this problem can be solved using both, the s-t mincut algorithm or the transductive SVM, they easily lead to degenerate cuts when the number of labeled and unlabeled samples is not well balanced. Including the cut size in the objective function to be maximized the problem can be equivalently interpreted as a ratio-cut problem that can be efficiently solved exploiting the spectral properties of the graph Laplacian.

Let D be the diagonal degree matrix $D_{ii} = \sum_j A_{ji}$, we can compute the Laplacian graph as $L = D - A$ and L is symmetric and positive semi-definite thus indirected. Graph Laplacians have recently been successfully adopted in image segmentation, [4], spectral clustering and dimensionality reduction, [11], since they represent a powerful manifold learning tool.

If we include the constraints in Eq.8 in the ratio-cut problem, the supervised optimization problem becomes:

$$\begin{aligned} &\min_{\overrightarrow{z}} \overrightarrow{z}^T L \overrightarrow{z} + c(\overrightarrow{z} - \gamma)^T C(\overrightarrow{z} - \overrightarrow{\gamma})\text{s.t.} \\ &\overrightarrow{z}^T \overrightarrow{z} = n \text{ and } \overrightarrow{z}^T 1 = 0 \end{aligned} \tag{9}$$

For each labeled example, the corresponding element of $\overrightarrow{\gamma}$ is respectively equal to $\gamma_+ = \sqrt{\frac{l_-}{l_+}}$ and $\gamma_- = \sqrt{\frac{l_+}{l_-}}$ for positive and negative examples, and it is zero for test examples and where l_+ (l_-) is the number of the positive (negative) labeled training example as in [6]. In Eq.9 c is a parameter that trades off training errors versus cut-values, and C is a diagonal cost matrix that accounts different misclassification costs for each example. Taking the eigendecomposition $L = U\Sigma U^T$ of the Laplacian, one can introduce a new parameter vector $\overrightarrow{w}$ and substitute $\overrightarrow{z} = U\overrightarrow{w}$. Since the eigenvector of the smallest eigenvalue of a Laplacian is always $\overrightarrow{1}$, the constraint in Eq.9 becomes equivalent to setting $w1 = 0$. Letting Ev be the matrix with all eigenvectors U and EV the matrix with all eigenvalues Σ except the smallest one, the optimization problem can equivalently be written as

$$\begin{aligned} &\min_{\overrightarrow{w}} \overrightarrow{w}^T D \overrightarrow{w} + c(Ev\overrightarrow{w} - \gamma)^T C(Ev\overrightarrow{w} - \overrightarrow{\gamma}) \text{ s.t.} \\ &\overrightarrow{w}^T \overrightarrow{w} = n \end{aligned} \tag{10}$$

Defining $G = (EV + cEv^T CEv)$ and $\overrightarrow{b} = cEv^T C\overrightarrow{\gamma}$ the objective function can also be written as $\overrightarrow{w}^T G\overrightarrow{w} - 2\overrightarrow{b}^T \overrightarrow{w} + c\overrightarrow{\gamma}^T C\overrightarrow{\gamma}$, where the last term can be dropped since it is constant. Problem of Eq. 10 is then minimized for $\overrightarrow{w}^* = (G - \lambda^* I)^{-1}\overrightarrow{b}$ where λ^* is the smallest eigenvalue of

$$\begin{bmatrix} G & -I \\ -\frac{1}{n}\overrightarrow{b}\overrightarrow{b}^T & G \end{bmatrix} \tag{11}$$

I is the identity matrix. The optimal value of Eq.9 is computed as

$$\overrightarrow{z}^* = Ev\overrightarrow{w}^* \tag{12}$$

producing a predicted value for each example in the test set. To make a hard class assignment on the predicted value a threshold can be used that we set to the fixed value of 0.5.

5. Model Update Engine

Transductive learning is a powerful approach to learn classification by adding, step by step, positive and negative examples in the training set. Of course, the main risk is that an error in classification of positive or negative examples can propagate errors making the successive labeling unreliable. To this aim a strategy for model updating must be adopted to avoid the injection into the models of elements that may cause classification errors. Thus the positive and negative models update is a very critical aspect. Here we explain some different strategies, specifically analysing our *spectral update* method.

Denoting $X_{l+} = \{x_i, 1\}_{i=1,\ldots,p}$ as the positive model constituted by p elements and $X_{l-} = \{x_i, -1\}_{i=p+1,\ldots,l}$ as the negative model containing n elements, we want to obtain the updated models X_{l+}^{new} and X_{l-}^{new}. The simplest way to choose both the models of positive and negative elements is *no update* which consists of using the same initial models given by the user, thus $X_{l+}^{new} = X_{l+}$ and $X_{l-}^{new} = X_{l-}$. Although in this way we can prevent the injection of wrong labeled elements, with no update the models do not carry information about changes in people appearance in time, therefore after a certain number of frames the system is unable to recognize the target person, leading to a short term tracing only.

A second strategy is called *last k results*, where the models are updated iteratively adding the last results and removing the oldest elements. The underlying hypothesis is that the elements closer in time are more likely to be similar to the current appearance of the object. The shortcoming with this solution is that the models tend to drift. At each iteration there is no control on the new added elements and small errors could be introduced. With the accumulation of these errors the models and consequently the classification of the unlabeled elements drift away from the initial target.

A smarter strategy, the one we adopted, is a *spectral update* where X_{l+}^{new} and X_{l-}^{new} are built using a clustering performed over the previous models X_{l+} and X_{l-} and the obtained results. The basic idea is to cluster the samples and maintain a set of elements for each cluster. In this way we can assure that both, the target and the negative elements are reliably represented considering changes in appearance, because with a good approximation each cluster is representative of a cohesive set of samples.

Here we consider separately the update of the positive and negative model.

5.1. Positive model update

Using a set of positive sample of the traced person, instead of a single element, we want to model both illumination variations and more general variations like the pose. A good way to achieve this goal is to have a certain number of items for each of the past variants of the target. At this aim we propose to update the model every m frames exploiting a clustering algorithm over the previous results retrieved by the system.

At frame t multiple of m, we call $R_{t-1} = (r_1, \dots, r_{t-1})$ the result retrieved from the frame 1 to the frame $t-1$, and A^* the affinity graph built over R_{t-1} where nodes correspond to the covariance matrix of each element computed in the same way of Eq.3, and the edges among them represent the distance in Eq.4. We exploit the spectral properties of the Laplacian $L = D - A^*$ performing the Unnormalized Spectral Clustering algorithm [11] in order to obtain k clusters $C_1, \dots, C_k$, where k is the number of clusters chosen employing the eigengap analysis. Finally we update the new model X^{new}_{l+} maintaining the elements closest to the centroid for each cluster, with an upper bound n^{l+}_{max} for the dimension of the model.

$$\begin{aligned} & X^{new}_{l+} = \{x_j, 1\}_{j=1,\dots,n^{l+}} \text{ s.t.} \\ & n^{l+} < n^{l+}_{max} \\ & x_j \in C_i,\ i = 1, \dots, k\ ,\ \bigcup_i C_i = R_{t-1} \\ & \sharp(C_i) > minCard \end{aligned} \tag{13}$$

Clusters with a really low number of elements ($\sharp(C_i) < minCard$) represent elements that differs from all the others, thus can be considered as outliers or misclassified elements and are discarded in the new model X^{new}_{l+}. In this manner (see Fig. 3 the positive model becomes an ensemble of more clusters representing different aspects of the pose.

5.2. Negative model update

Referring to the update of the negative model the problem is quite different. Indeed in this case we do not need to model the variation in appearance of a target, rather we need to refresh the samples that do not depict the target but other people detected in the scene. For this reason we propose to continuously update X_{l-} by adding for each frame the elements with a negative value of the resulting label and fixing a maximum threshold n^{l-}_{max} for the total number of elements in the model.

When the size of the negative model is greater than n^{l-}_{max} we cluster the elements of X_{l-} with the same technique explained in Sec.5.1 and randomly select elements from each cluster C_i.

$$\begin{aligned} & X^{new}_{l-} = \{x_j, 1\}_{j=1,\dots,n^{l-}} \text{ s.t.} \\ & n^{l-} < n^{l-}_{max} \\ & x_j \in C_i,\ i = 1, \dots, k\ ,\ \bigcup_i C_i = X_{l-} \end{aligned} \tag{14}$$

6. Experiments

In this section we assess the performance of the proposed system. To evaluate the degree of reliability in tracing a target person we collect video from the publicly available datasets THIS [1], PETS2004 [2] and PETS2009 [3], moreover we acquire a set of sequence from the camera installed in our campus specifically focused to test the quality of re-acquisition after occlusion and long term tracing.

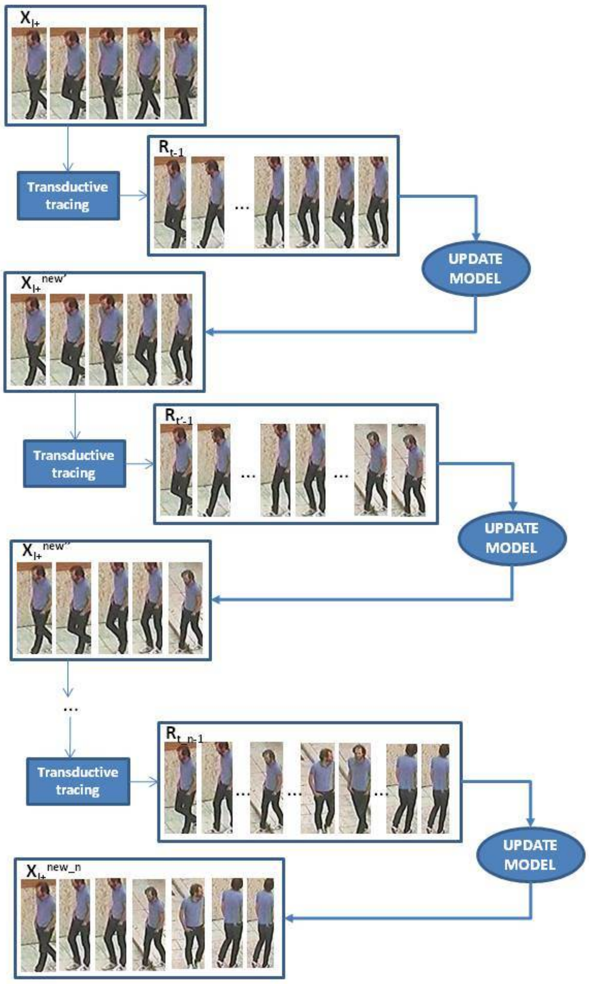

Figure 3. Example of consecutive iterations of the Positive Model Update

THIS and PETS examined videos in Fig.4, show people walking in metro stations and public areas with different viewpoints. Notably are the sample sequences depicted in the first and the third row of Fig. 4, displaying difficult tracking conditions due to the lack of a fixed background and to the crowded conditions.

The dataset acquired in our campus as depicted in Fig. 5, shows images with several sources of variation and many people entering and exiting the scene more than one time; particularly the sample frames in the figure show an exam-

[1] http://www.openvisor.org

[2] http://www-prima.inrialpes.fr/PETS04/caviardata.html

[3] http://www.cvg.rdg.ac.uk/PETS2009/a.html

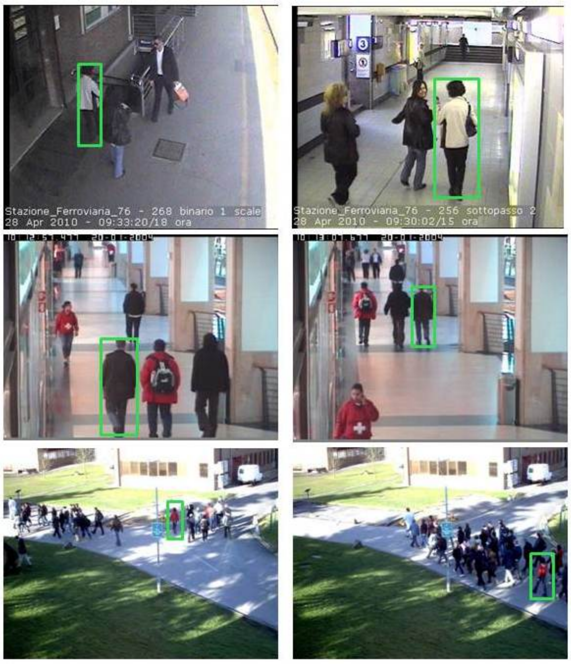

Figure 4. Sample frames from datasets THIS, first row, PETS2004, second row and PETS2009, last row.

ple of long term occlusion with the target person disappearing behind the pillar and then re-appearing. Always referring to this set of videos the different lighting conditions, the low image quality and the distance of the camera from the people do not make the task of detecting and tracing an object easy.

We use an approximately number of 50 video sequences, each one consisting of an average number of 500-600 frames and we tested our system with 2-4 different target persons for each sequence.

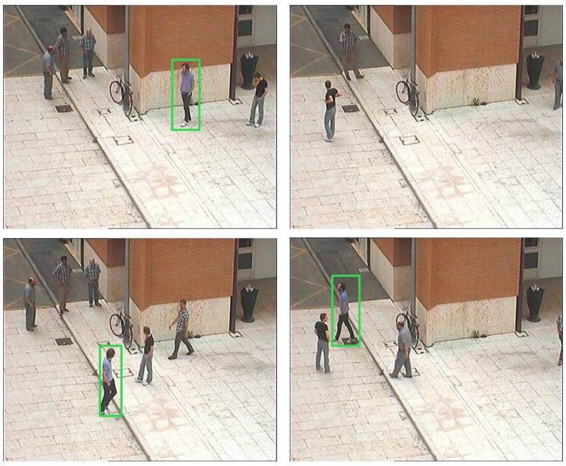

Figure 5. Sample frames from our campus videos. The sequence highlights a typical re-acquisition after occlusion case.

We measure the quality of our proposal in terms of precision and recall comparing the system with the results in the similar work of Coppi et al. [1] where only the positive labeled model was exploited. We also compare our method with the baseline method of Zha et al. [16], that recently proposed the use of graph-based transductive learning for visual tracking without any update strategy.

Tab. 1 and 2 contain respectively average recall and precision results obtained from THIS, PETS2004, PETS2009, and CampusVideo using our Transductive Tracer with Positive and Negative examples, TTPN, performed both with and without the proposed update strategy and compares these values to the results obtained with the tracing system proposed in [1] and with the baseline method in [16].

	THIS	PETS04	PETS09	Campus
TTPN NoUpd	0.93	0.90	0.62	0.66
TTPN SpUpd	**0.95**	**0.94**	**0.79**	**0.83**
TTP Upd	0.91	0.87	0.33	0.47
TLT Baseline	0.92	0.91	0.30	0.44

Table 1. Average Recall values obtained with our Transductive Tracing system with Positive and Negative labeled elements with No Update strategy (TTPN NoUpD) and with the proposed Spectral Update Strategy (TTPN SpUpd), compared with the Transductive Tracing with only Positive labeled elements continuously Updated in time(TTP Upd) [1] and finally with the baseline method of Transductive Learning Tracking (TLT Baseline) proposed in [16]

	THIS	PETS04	PETS09	Campus
TTPN NoUpd	0.80	0.78	0.65	0.63
TTPN SpUpd	**0.97**	**0.96**	**0.81**	**0.86**
TTP Upd	0.95	0.90	0.40	0.51
TLT Baseline	0.76	0.68	0.37	0.45

Table 2. Average Precision values obtained with our Transductive Tracing system with Positive and Negative labeled elements with No Update strategy (TTPN NoUpD) and with the proposed Spectral Update Strategy (TTPN SpUpd), compared with the Transductive Tracing with only Positive labeled elements continuously Updated in time(TTP Upd) [1] and finally with the baseline method of Transductive Learning Tracking (TLT Baseline) proposed in [16]

We point out how the presented method outperforms the baseline algorithm and method with only the positive model. Both precision and recall are significantly improved by the negative model exploitation instead of the single positive model. This improvement is particularly noticeable for the PETS2009 and our campus video sequences, which exhibits more difficult conditions, while is slight for THIS and PETS2004. Regarding PETS2009 videos, the method presented in [1] was not able to reliably distinguish and trace the target among the large number of persons on the scene. Similarly, for what concerns the set of video recording on our campus, only exploiting both, the positive and negative labeled models lead to a robust long term tracing capable to deal with occlusions and with large changes in pose and appearance of the target. Results in Tab.1 and 2 also underline how the spectral update is effective in improving the tracing performances.

As stated, our tracing method does not exploit motion information and strongly depends on the efficacy of people detection. In our experiments, especially in crowds and in complex scenarios, the HOG based people detection leads to a large number of false positive results, which are used to initialize the negative model improving the results.

7. Conclusions

In this paper we propose a system to trace a single person in real video surveillance context. While the individual tools and methods (*i.e*. graph Laplacians and transductive learning) are not new, their combination with a strong and mathematically well founded update strategy has led to a powerful tool especially for surveillance and forensic applications.

Concluding the proposed system represents a robust tracing system and differs from tracking systems, since no spatial and temporal information are used and no motion estimation is computed. Exploiting appearance information of the detected person and the knowledge encoded in the positive and negative labeled models the system is able to follow a person in videos and has showed to be robust to appearance changes, motion challenges and occlusions, situations where convention tracking system tend to have impressive degradation in performance.

Experiments on public and ad-hoc dataset exhibit good and encouraging values for precision and recall.

Future extensions of the work may consider improvements in time consumption using an iterative scheme to update the graph Laplacian without recomputing it for every frame. Moreover, another extension may stronger involve crowded and particularly occluded scenes.

References

[1] D. Coppi, S. Calderara, and R. Cucchiara. Appearance tracking by transduction in surveillance scenarios. In *8th Intl. Conf. on Advanced Video and Signal-Based Surveillance (AVSS)*. IEEE Computer Society, 2011. 2, 4, 7

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 886–893. IEEE Computer Society, 2005. 2

[3] P. Dollr, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: A benchmark. In *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 304–311. IEEE Computer Society, 2009. 2

[4] O. Duchenne, J. Audibert, R. Keriven, J. Ponce, and F. Segonne. Segmentation by transduction. In *Proc. of Int'l. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8. IEEE Computer Society, 2008. 1, 5

[5] W. Forstner, B. B. Moonen, and C. Gauss. A metric for covariance matrices. Technical report, Dept.Geodesy Geoinform., Stuttgart Univ., Stuttgart, Germany, 1999. 4

[6] T. Joachims. Transductive learning via spectral graph partitioning. In *Proc. of Intl. Conf. on Machine Learning (ICML)*, pages 290–297, 2003. 4, 5

[7] P. Koppen and M. Worring. Multi-target tracking in time-lapse video forensics. In *Proceedings of the First ACM workshop on Multimedia in forensics*, MiFor '09, pages 61–66. ACM, 2009. 1

[8] I. Leichter, M. Lindenbaum, and E. Rivlin. Tracking by affine kernel transformations using color and boundary cues. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 31:164–171, 2009. 1

[9] M. Li, W. C. 0012, K. Huang, and T. Tan. Visual tracking via incremental self-tuning particle filtering on the affine group. In *Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 1315–1322. IEEE Computer Society, 2010. 1

[10] Y. Liu, G. Li, and Z. Shi. Covariance tracking via geometric particle filtering. *EURASIP J. Adv. Signal Process*, 2010:1–22, 2010. 2

[11] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 5, 6

[12] M. J. Metternich, M. Worring, and A. W. M. Smeulders. Color based tracing in real-life surveillance data. In Y. Q. Shi, editor, *Transactions on Data Hiding and Multimedia Security V*, Lecture Notes in Computer Science, pages 18–33. Springer Berlin / Heidelberg, 2010. 1, 2

[13] F. Porikli, O. Tuzel, and P. Meer. Covariance tracking using model update based on lie algebra. In *Proc. of Intl. Conf. on Computer Vision and Pattern Recognition (CVPR)*, pages 728–735. IEEE Computer Society, 2005. 2

[14] R. Vezzani, C. Grana, and R. Cucchiara. Probabilistic people tracking with appearance models and occlusion classification: The ad-hoc system. *Pattern Recognition Letters*, 32(6):867–877, Apr. 2011. 2

[15] A. Yilmaz, O. Javed, and M. Shah. Object tracking: A survey. *ACM Comput. Surv.*, 38(4):1–45, December 2006. 1

[16] Y. Zha, Y. Yang, and D. Bi. Graph-based transductive learning for robust visual tracking. *Pattern Recognition*, 43(1):187–196, 2010. 1, 2, 7

[17] K. Zimmermann, J. Matas, and T. Svoboda. Tracking by an optimal sequence of linear predictors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, pages 677–692, 2009. 1