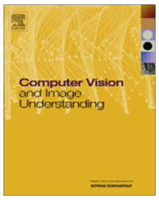
Computer Vision and Image Understanding 117 (2013) 229–237

Contents lists available at SciVerse ScienceDirect

Computer Vision and Image Understanding

journal homepage: www.elsevier.com/locate/cviu



Track based relevance feedback for tracing persons in surveillance videos ☆

Michael J. Metternich *, Marcel Worring

ISLA, University of Amsterdam, Science Park 904, 1098 XH Amsterdam, The Netherlands

ARTICLE INFO

Article history:
Received 26 March 2011
Accepted 18 November 2012
Available online 6 December 2012

Keywords:
Surveillance
Event reconstruction
Person tracing
Relevance feedback

ABSTRACT

While surveillance cameras are a powerful tool for the prevention, detection and resolving of crimes, for most cases the amount of video data has become unmanageable. To ease the analysis, various automatic methods have been proposed, focusing on data-management, detecting suspicious behavior, person recognition, or event reconstruction. In this paper we focus on event reconstruction, in particular on tracing the whereabouts of people. The standard approach for such event reconstruction is to first detect persons in single frames and then match a query to all detections to retrieve the same person in multiple cameras. However, since the number of detected persons is large and performance of matching techniques limited, this process is slow and prone to errors. Intelligent interactive techniques are urged for. We propose to represent detected persons by their complete track within a single camera instead of a single detection and thereby reduce the search-space. On these tracks we use Relevance Feedback to improve recall with only a small effort of the user. Testing the tracking method on a benchmark dataset and a real-life dataset led to a reduction of the search space of 90%, while tracing accuracy based on the distance between tracks improved recall by up to 110% when compared to random tracing. Adding Relevance Feedback led to an additional improvement in recall of up to 400% compared to sequential scanning using the same number of visual assessments.

1. Introduction

After the 2005 bombings in London, a total of 2500 items of CCTV footage were analyzed by police officers to find the persons responsible. Every officer had to manually look through hours of video material and describe all potential suspects in such a way that other officers were able to match them to the people they found. While eventually effective, this process was obviously immensely slow and cumbersome. To simplify the search through a vast amount of video data, two directions can be distinguished: (i) automating (parts of) the search process and (ii) providing additional real-time clues to help the observer [20,23,27]. In this paper we focus on the first direction, which is known as post-incident investigation. For this, all available information about an event is collected and combined to reconstruct that event. An important task in this reconstruction is to deduce the identity and whereabouts of victims, offenders and witnesses. Traditionally, this tedious task is done manually. With the advancement of research in computer vision, however, more automatic methods have become suited for event reconstruction.

Ideally, we need a fully automatic system capable of dealing with the challenges of real-life CCTV videos. When an investigator selects a person in such a system, all other instances of that person (in other cameras) are shown. The investigator can then see where that person has been before and if he met other persons. Additionally, the system would automatically recognize both persons from a database of wanted criminals. An investigator then only needs to bring in the person of interest for questioning.

In contrast to the previously described ideal system, truly automatic recognition of a person is only possible in high-resolution videos or under lab conditions. In practice many surveillance cameras work with limited resolution, suffer from compression, and seldom work in controlled conditions. The best we can do in this situation is comparing a selected person with all persons found. All other detected persons are ordered based on similarity to the person of interest. The user then decides if it is indeed the same person. Assuming a person is more similar to himself than to someone else, we can assume all other tracks of that person come up first in the resulting ordered list. The matching is done first in the same camera (tracking) and subsequently in other cameras (tracing). Formally, *tracking* is the following of a person in one camera until that person leaves the camera view. *Tracing* is the following of a person in different non-overlapping cameras, hence combining the tracks obtained with tracking in these cameras.

Examples of current matching and tracing based systems are described in [2,10] and [12]. A limitation of these methods is that they are based on images, ignoring an intrinsic characteristic of video namely time. When a person walks in clear view of the camera,

☆ This paper has been recommended for acceptance by Edwin Hancock.
* Corresponding author.
E-mail address: michael.j.metternich@gmail.com (M.J. Metternich).

1077-3142/$ - see front matter © 2012 Elsevier Inc. All rights reserved.
http://dx.doi.org/10.1016/j.cviu.2012.11.004

that person is likely to be detected multiple times. Since orientation towards the camera and general appearance hardly change in such a small time period, the feature descriptions of these detections are likely to be similar. Therefore, when searching for a person, large clusters of similar images are found in the ordered list of matching persons. Searching through such a list is then a boring and time consuming process. The method we propose is to first find tracks and order these tracks instead of detections in single images.

We define the distance between two detections as the euclidean distance between two points in a weighted feature space. The distance between two tracks is then an aggregation of all possible distances between the two tracks. While searching through a list of tracks ordered on distance to the query is much faster than temporally or randomly ordered lists, this process could still be greatly improved. An important observation in this regard is that every person has its specifics and therefore needs a specific focus on features that are descriptive for that person. For example, for one person the color of his jeans are distinctive, whereas for another person the texture of his jacket provides a valid cue. Due to this changing notion of distance in feature space, it is hard to pre-define a distance metric that works under all circumstances. In other words, user feedback is needed to adapt the distance metric for a specific query. Relevance Feedback (RF) is a method specifically designed for this type of interaction in text search and Content Based Image Retrieval (CBIR).

Surprisingly, few other frameworks incorporate RF in surveillance applications. To the best of our knowledge, only three suggestions have been made so far to use this type of information. Meessen et al. [17] and Zhang et al. [28] use RF to improve the retrieval of high level concepts such as "fighting", "robbery" or "similar scene". While this information is effective to speed up the search through large databases, the use for post-incident investigation is limited. In this setting we aim to understand a single scene as thoroughly as possible, instead of finding similar scenes. Ali et al. [3] use RF for an approach similar to the method we propose in this paper. They also aim to use RF for person matching over multiple cameras. In their approach the user gives a number between 0 and 1 indicating the distance to some query image. This approach is unfortunately ambiguous. Is a person wearing a similar jacket, but different trousers more or less similar to the query than a person wearing similar trousers and a different jacket? We develop a more intuitive and explicit feedback method in which the user identifies whether the query and the candidate detection are the same person.

In our surveillance RF scenario the first step is to let the system return a "naively" ordered list of tracks based only on the query track. In step two the user judges the currently displayed tracks on whether they show the same person as the query track or not. In step three the machine processes this feedback and gives an improved query set. Afterwards we return to step two. This process is continued until the user feels all instances of that person are found.

As RF is a method with a long history in both text retrieval and content-based image retrieval, many variations have been proposed [4,13,21]. These traditional schemes, however, do not directly generalize to our track based surveillance.

First of all, the *problem goal* is different. Traditional RF algorithms are designed to find objects of a similar class as the objects the user gives positive feedback on. Put differently, these methods aim to bridge the *semantic gap* [24]. In surveillance however, a single instance of a person is found and we want to retrieve other instances of that same person under different sensory conditions. In other words, we try to overcome the *sensory gap*.

The second difference is the *object of interaction*. In standard RF scenarios, all images in the training data are individually labeled to train a classifier capable of predicting the class of unseen objects. In track matching, it is rather tiresome for the user to label all elements of every observed track. We thus need an efficient approach to user interaction, allowing the user to provide feedback on complete tracks.

Fig. 1 shows an overview of the feedback loop embedded in a complete person tracing system. We describe the tracking and distance metrics between tracks in Section 2. Subsequently, the five most prevalent RF methods capable of employing user feedback are described in Section 3. We describe the experimental setup in Section 4 and give the results of these experiments on a benchmark dataset and a real-life surveillance dataset in Section 5.

The contributions of this paper are as follows:

1. Rather than using single images for interactive tracing of people we focus on complete tracks and provide methods for optimally comparing them.
2. We propose to use the user's feedback more effectively by incorporating relevance feedback in the search process, where relevance is based on the person's identity and not on appearance only.
3. We show the validity of the methods by comparing various standard RF methods and see how they can be adapted for this specific scenario.

2. Tracking and matching

Let us first describe more precisely the tracing process as needed for post-incident surveillance. To that end a bottom-up data analysis process is needed to provide the users with the basic elements for interaction and matching. From there interactive matching can proceed. We will now elaborate on these steps.

2.1. Problem definition

The first step to tracing is to identify the complete set of all candidate detections D. An automatic tracking method combines subsets of D from a single camera into tracks denoted by t. The full set of tracks is then denoted by T. When the user indicates a person of interest by defining a query track Q_0, we aim to retrieve the set of tracks R showing the same person. To do so, the complete set of tracks T is ordered based on their distance Δ to the query Q_0. The distance between tracks depends on the distance between the feature descriptions f of all detections comprising each of these two tracks. We define this distance between the feature descriptions of two detections as $\delta(f(t_1^k), f(t_2^k))$. Please note that the two

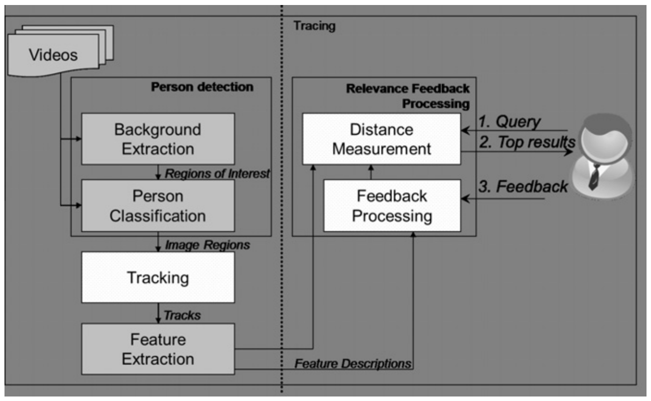


Fig. 1. Description of the event reconstruction system we propose for person tracking and tracing in surveillance videos. Grey boxes are discussed in previous work [18], while this paper focuses on the elements shown in white.

distance metrics Δ and δ are not the same. Δ gives the distance between two tracks whereas δ gives the distance between two detections. For δ any state-of-the-art distance metric can be used. With the resulting ordered set of tracks we can incorporate the user in the search process. When the user gives positive feedback on a track, that track is added to the set of positive tracks S^+. Likewise, when the user gives negative feedback that track is added to S^-. The combined set S grows after every feedback by the user.

2.2. Methods used

To obtain the set of detections and their subsequent tracks we use the methods discussed more thoroughly in [18]: We estimate the background in all frames of a single camera using the method described in [30]. All regions of movement are then obtained by subtracting this background from the original frames. We describe the regions of movement using Histograms of Oriented Gradients [7] and these are then classified as either “person” or “other”. For classification we use an SVM classifier with a Gaussian kernel, trained on the INRIA Person dataset [6]. The resulting detections D are the basis for tracking persons within a single camera. The background model we obtained after background extraction is used to mask the background in every detection.

Since in many real-life surveillance settings time-lapse data is used, most standard tracking methods will fail. We, therefore, use the method proposed in Koppen et al. [15] to combine multiple detections of a single person in consecutive frames. This method uses hysteresis thresholding to allow for bridging gaps in the tracks due to low quality or occlusions and A^* search to find an optimal assignment of the detections to tracks for each person.

In this work we compute the feature description of a detection $f(t^k)$ using the Multi Color-Height Histogram (MCHH) [9]. This method uses two four-dimensional histograms of six bins per dimension. For the first histogram we put every non-masked pixel in its appropriate bin, using the three channels of the Opponent Color space [25] and its relative vertical position. For the second histogram the approach is similar, but instead we use ranked Opponent Color space, a variation on RankedRGB [16]. While the two histograms are similar in many ways, they are designed to capture different elements of the target detection. The first histogram is designed to be a global structure and color descriptor. The second histogram focuses more on the local structure and colors of the detection. Both histograms are normalized using L2 normalization. While any distance measure can be used to obtain δ, we use the sum of the two histogram intersections for its simplicity and performance.

2.3. Automatic track matching methods

For track matching to be successful, the distance measure Δ should cluster the feature descriptions of tracks showing the same person and scatter the feature descriptions of tracks showing different persons. Ideally this distance function is independent of the direction towards the camera, ignores tracking errors and is invariant to the position of the person within a frame. We now introduce a number of options to obtain this distance measure. The first method is to aggregate the descriptions of all detections in a track. This way different views of a single person are contained in a single representation. The most straightforward approach to obtain this aggregation is to average all feature descriptions:

$$\Delta_{Average}(t_1, t_2) = \delta(\overline{t_1}, \overline{t_2}) \tag{1}$$

where $\bar{t}$ is the average descriptor of track t. A second approach is to match the two detections that show the person(s) in those two tracks best. To do so, we need to find the best representative for each track. Since a person is best visible when he or she is closest to the camera, we represent the complete path using the feature description of the single largest detection. We describe this as:

$$\Delta_{Largest}(t_1, t_2) = \delta\left(f\left(t_1^{\sigma_1}\right), f\left(t_2^{\sigma_2}\right)\right) \tag{2}$$

where σ_1 and σ_2 give the index of the largest bounding box in their respective tracks.

Lastly, we observe that the appearance of people is dependent on their orientation towards the camera. To obtain a partly invariant distance measure we match the two detections most likely to have a similar orientation. Obviously, this orientation is unknown, but we observe that the visually most similar detections of two tracks often have similar orientation towards the camera. We therefore match the most similar detections between two tracks:

$$\Delta_{Minimal}(t_1, t_2) = \min_{m,n} \delta\left(f\left(t_1^m\right), f\left(t_2^n\right)\right) \tag{3}$$

Fig. 2 shows a visual representation of these three matching methods.

2.4. Method comparison

When using these methods, we have to consider that in real-life situations the quality of automatically detected persons is very poor. In these situations, the resulting tracks contain many false positives, i.e. “other” classified as “person”. Additionally, the tracked person might be lost, resulting in the track following a different person. For the three methods introduced in this section, this has different consequences on performance.

For average track matching, a single false detection has a negligible effect on performance. In this situation the average description is almost similar to the average description of only positive detections. When the number of false positives is large though, the average feature description of both positive and negative detections no longer describes the tracked person and thus becomes meaningless.

For Largest Detection Matching and Minimal Distance Matching a similar observation can be made since both methods use only a single detection to describe a track. If that detection does not show the person of interest, matching that track to other tracks gives no information about the presence of that person. As long as the number of false detections is low, the chance of this happening is low. But when it happens the matching is meaningless. The two methods also have differences though. Largest Detection Matching always uses the same detection to represent a track whereas this changes for Minimal Distance Matching. Since the distance between detections showing the same person is most likely smaller than between two random detections, it is more likely that the person of interest is visible in the matched detections. Additionally, it is unlikely that false positives from different cameras are similar. It is therefore likely that this method probably performs better than Largest Detection Matching.

In the next section we describe how to let the system learn from user feedback to improve all distances Δ.

3. Interactive matching

In content based image retrieval, user feedback has been used for many years with great success [29]. The simple idea here is to let a user give feedback on a set of retrieved images and use this feedback to improve the results of the next set of images. This feedback process has gained popularity as Relevance Feedback (RF). In literature various variations on RF have been proposed, which can roughly be distinguished as Borda Count, Feature Weighting, Query Optimization, Density Estimation or Classification. Fig. 8 gives a visualization of these five methods and the baseline approach

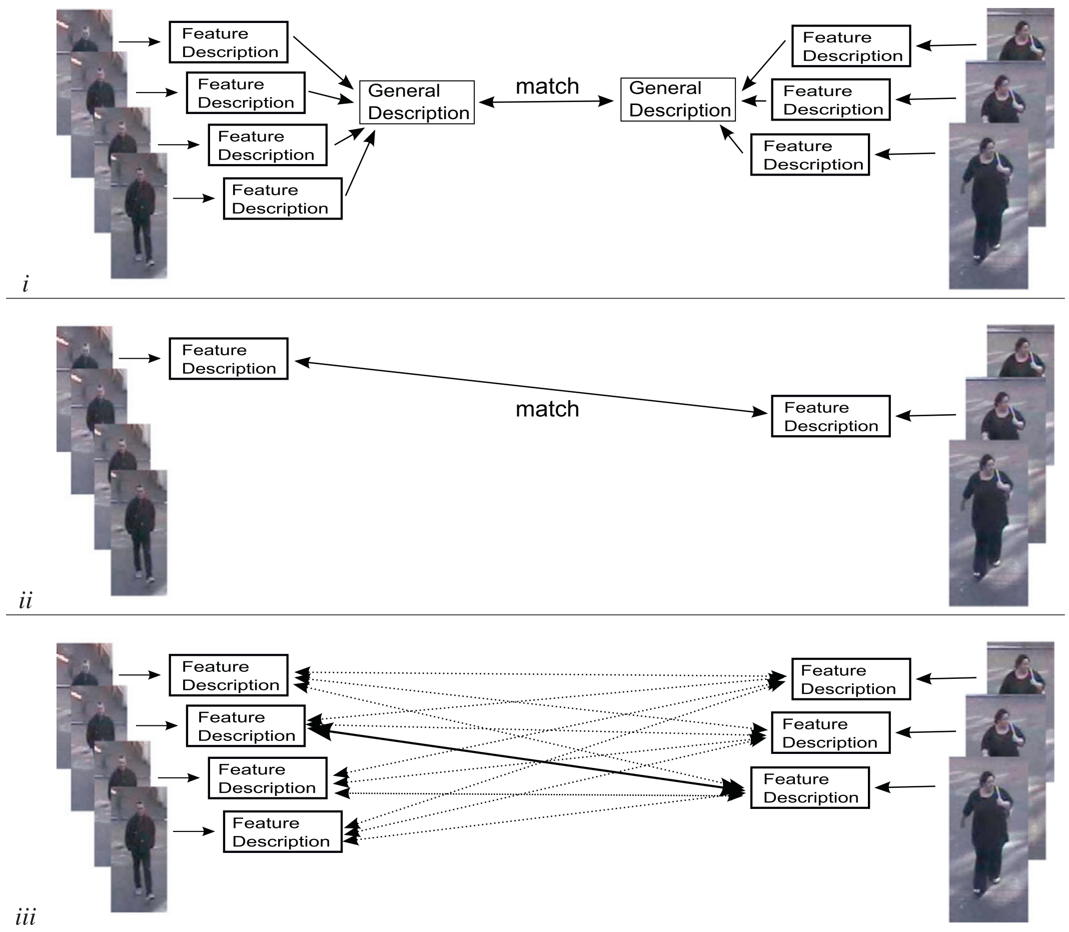


Fig. 2. Comparison of two tracks using an average description (i), the largest detection (ii) or the minimal distance between all detections (iii).

"no interaction". Before describing these methods though, we first formalize the interaction as used for RF and all elements needed to formalize this interaction.

To more easily show the differences between all RF methods, we describe the complete query space at any moment during interaction as a goal dependent 4-tuple $\{\mathcal{Q},\mathcal{I},\Delta,\mathcal{Z}\}$, similar to Smeulders et al. [24]. Here, $\mathcal{Q}$ is the feature description of the query and $\mathcal{I}$ the active set of tracks. The parameterized function giving the distance between tracks is denoted by Δ and finally $\mathcal{Z}$ gives for all tracks the probability that they show the same person as the query track. All RF methods in this paper are described by the manner in which this query space is updated.

$$\{\mathcal{Q}_i,\mathcal{I}_i,\Delta_i,\mathcal{Z}_i\}\overset{RF_i}{\rightarrow}\{\mathcal{Q}_{i+1},\mathcal{I}_{i+1},\Delta_{i+1},\mathcal{Z}_{i+1}\} \tag{4}$$

The difference between methods lies in the components kept constant and the way they change the other elements.

The distance function Δ described in the previous section gives the system's point of view of distance. However, the user makes the final decision by indicating whether the query track Q and a similar track indeed show the same person. To make this interaction explicit we define

$$\Delta_{user}(Q,t)=\begin{cases}0, & \text{if same person}\\ \infty, & \text{otherwise}\end{cases} \tag{5}$$

In our RF setting we let V_i be the set of tracks visualized for display to the user at iteration i of user feedback. For all these tracks the user tells the system if that track shows the same person as the query track. If the user gives negative feedback, we can assume all individual detections in that track are negative. Therefore, the feature descriptions of all detections in that track are added to the set of negative elements S_i^-. Here, i again denotes the iteration of user feedback. Unfortunately, due to errors in either the detection or tracking phase, this approach cannot be used for positive feedback. For example, when the track starts with one person but loses him or her after a few frames, not all detections of that track show the same person. As a result we can only use the detections selected by the user to increase the set of positive elements S_i^+. The complete set of elements the user gave feedback on is then the combined set of all positive and negative tracks over all iterations.

After feedback, all active elements are removed from the active set I, i.e. $I_{i+1}=T-\widehat{I}_i$. The remaining set I_{i+1} is then reordered using one of the following interaction methods.

3.1. Baseline

In the baseline, the complete set of tracks is ranked based on distance to the query. In this situation, user feedback is not used to reorder the set of active tracks.

3.2. Borda Count

Instead of a naive linear search, we can reorder the original list after every round of feedback. The simplest method to do so is to use all tracks with positive feedback as new queries. By combining all orderings, performance will hopefully improve when compared to the initial ordering. A well-known method to obtain such an aggregated list is Borda Count [8]. This method assigns scores to all tracks based on their position in the orderings. Reordering based on the total sum of these scores then leads to a combined ranking:

$$\Delta^{BC}(Q_i,t)=\sum_{s\in S^+}\Delta(s,t) \tag{6}$$

$$Q^{BC}_{i+1}=S^+ \tag{7}$$

While Borda Count can effectively implement user feedback in the reordering of tracks, it ignores differences between the feature descriptions of positive and negative elements.

3.3. Feature Weighting

To manage the influence of feature dimensions on the distance measure, Feature Weighting [22] adds a weight to all dimensions. Ideally, this method sets the weights in such a way that tracks containing the same person are moved towards the query. Simultaneously, the feature representations of tracks containing different persons are moved away from the query. In this process, the query point itself does not change. In the case of Histogram Intersection this updated distance would become:

$$\Delta^{FW}(Q_i,t)=\sqrt{\sum_{k=1}^{K}w^k\cdot\left(t^k-Q_i^k\right)^2} \tag{8}$$

where Q_i^k is feature k of the query after interaction round i. The weight w^k is set inversely proportional to the variance over the kth feature values of all known relevant items.

3.4. Query Optimization

For Query Optimization the query point in the feature space is adapted in such a way that it is directed towards relevant elements and moved away from non-relevant elements. When using Rocchio's formula [19], the updated query becomes:

$$Q^{QO}_{i+1}=\alpha\cdot Q^{QO}_i+\beta\cdot\overline{S^+}-\chi\cdot\overline{S^-} \tag{9}$$

where $\overline{S^+}$ and $\overline{S^-}$ are the average feature description of all elements in S^+ and S^- respectively and α, β and χ are parameters steering the influence of the different components.

3.5. Density Estimation

Density Estimation combines the weighting of dimensions of Feature Weighting with the movement of the query point of Query Optimization. Additionally this method allows correlations between feature dimensions to be expressed via a weight matrix M. We use the query optimization method and distance function proposed in [14]:

$$\Delta^{DE}(Q_i,t)=\sum_{k=0}^{n}\sum_{l=0}^{n}M^{kl}\left(Q_i^k-t^k\right)\left(t^l-Q_i^l\right) \tag{10}$$

$$Q_{i+1}=\overline{S^+} \tag{11}$$

where the element $M_{k,l}$ is set inversely proportional to the covariance of features k and l. Similar to Feature Weighting this method uses only relevant tracks.

3.6. Classification

In contrast to the previously discussed RF methods, Classification is not based on optimizing the distance function between elements. Instead, this method focuses on obtaining a discrimination function capable of distinguishing relevant tracks and irrelevant tracks. Any state-of-the-art classifier can be used for this purpose. We propose to train a single Support Vector Machine (SVM) [26], as it has been proven to perform well in CBIR and text retrieval and is particularly suitable for two-class classification problems [29]. This classification method constructs a hyperplane that has the largest functional margin to minimize its generalization error. To train the hyperplane for track based RF we use the complete set of S^+ and an equally sized random subset of S^-.

When using an SVM for track classification a distance function is needed. However, in our setting we do not have a single vector to classify, but a complete track composed of individual detections. To that end we measure the distance C^{CL} of all detections of that track to the hyperplane. The distance to the element most certain to be positive is then considered as the distance between the query and the complete track:

$$\Delta^{CL}(Q_i,t)=\min_k C^{CL}(t^k) \tag{12}$$

$$\mathcal{Z}_{i+1}=C^{CL}(Q_i,T) \tag{13}$$

here, the operator C^{CL} is overloaded, classifying either a detection or the complete set of tracks T.

In summary, we distinguish five different RF methods which can be used to improve the retrieval of tracks in a surveillance setting, most likely with different degrees of success.

	Q_{i+1}	Δ_{i+1}	$\mathcal{Z}_{i+1}$	Ranking
Baseline				Δ to the query
BC	S_i			Δ to the query set
FW		Δ^{FW}		Δ to the query
QO	Q^{qo}_{i+1}			Δ to the query
DE.	$\overline{S^+}$	Δ^{de}		Δ to the query
CL			$\mathcal{Z}_{i+1}$	Δ to the hyperplane

4. Experiments

In this section we describe the experiments to (i) evaluate the matching methods and (ii) compare the RF methods proposed.

4.1. User interaction

For all experiments in this paper the system employs Δ_{user} using ground truth data. In other words, the person shown in Fig. 1 is replaced by an automated "perfect user". We assume the user is

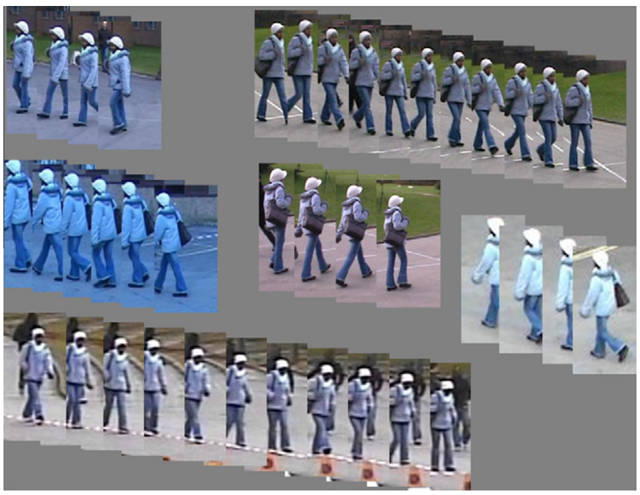


Fig. 3. All detected tracks of one person after detection and tracking methods are applied to the PETS dataset.

always capable of matching a track to the query if the same person is indeed present in both tracks. Due to large variations in appearance this assumption will not always hold in real-life situation. However, if the user would be willing to carefully observe every element in every track, it is a fair assumption.

4.2. Evaluation

Performance of all methods is measured using recall, the percentage of similar tracks found. More formally:

$$\text{Recall}_i = \frac{|S_i^+|}{|R|} \tag{14}$$

Based on recall, we compare all methods using the Cumulative Matching Curve (CMC) [11]. A CMC gives for every rank the average recall. An increase in rank number therefore always means a similar or higher recall score. For example, if on average 50% of the positive results are found after 10 tracks, the recall is 0.5 at rank 10. At rank 15 at least 50% is found as well, since these were already matched after 10 tracks. Better performing methods show a CMC curve leaning towards the top left corner. This measure is similar to an ROC curve, but focuses on ordered lists instead of classification:

$$CMC(k) = \frac{1}{n}\sum_{n=1}^{N} \frac{\left|\{R_n\} \cap \left\{\text{Top}_n^k\right\}\right|}{|\{R_n\}|} \tag{15}$$

where N is the complete set of queries and Top_n^k the top k tracks for query n.

4.3. Datasets

We evaluate on two different datasets. The first dataset is the S2.L1 subset of PETS 2009 [1]. We consider this dataset our benchmark since it is recorded under constrained conditions; the videos show limited occlusion, a small number of persons is simultaneously visible and the weather conditions are stable during the recordings. This dataset is publicly available and well documented since it is used as part of the PETS tracking challenge. Applying the previously described person detection resulted in a total of 5455 detections. After applying the tracking algorithm we manually labeled the resulting 712 tracks to obtain a ground truth. For this dataset information about overlapping camera views could be used to match detections. However, since this information is in general not available in public surveillance settings we ignore this information here. Sample tracks are given in Fig. 3.

In addition to the PETS dataset, we recorded a real-life dataset with the assistance of the Dutch police: the Amsterdam dataset. This dataset consists of simultaneous recordings of five cameras in the red light district of Amsterdam. The cameras have no overlap in field-of-view and each recording lasts one hour. The recordings were made as part of the regular surveillance process for that area. A ground-truth is obtained by manually labeling the positions of seven persons who were asked to walk around in the area under surveillance. The dataset contains several sources of variation. Most notable are the changes in weather, camera angle, apparent colors and texture of clothing and reflections in windows. Furthermore, the visual appearance of these seven persons varied greatly; some were wearing distinctive colors whereas others were less characteristic. Sample tracks of applying the detection and tracking methods are shown in Fig. 4. The persons cooperating in the experiment were present in at least three and at most five cameras. Applying the detection and tracking methods resulted in a total of 98,260 detections and 2433 tracks. The seven participants are clearly visible in 51 of those tracks.

To compare all methods in a fair manner we assume:

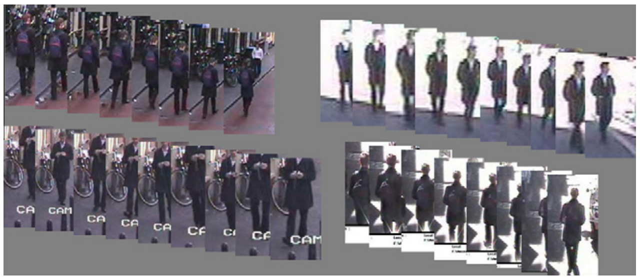


Fig. 4. A subset of all detected tracks of a single person after detection and tracking methods are applied to the Amsterdam dataset.

1. Great discernibility based on color information.
2. No overlap in field of view.
3. Simultaneous recordings.
4. Little occlusion and persons are visible from head to toe.

4.4. Experiments

To measure the effectiveness of tracing, we first compare the discussed track matching methods. Afterwards we focus on the different RF methods described in Section 3.

4.4.1. Track matching

To evaluate the different matching methods we repeatedly take a random track from the ground truth as query. As mentioned in Section 2, we order all tracks in other cameras than the query based on their distance to the query track. For every query we thus obtain a CMC curve and average those to obtain an overall result.

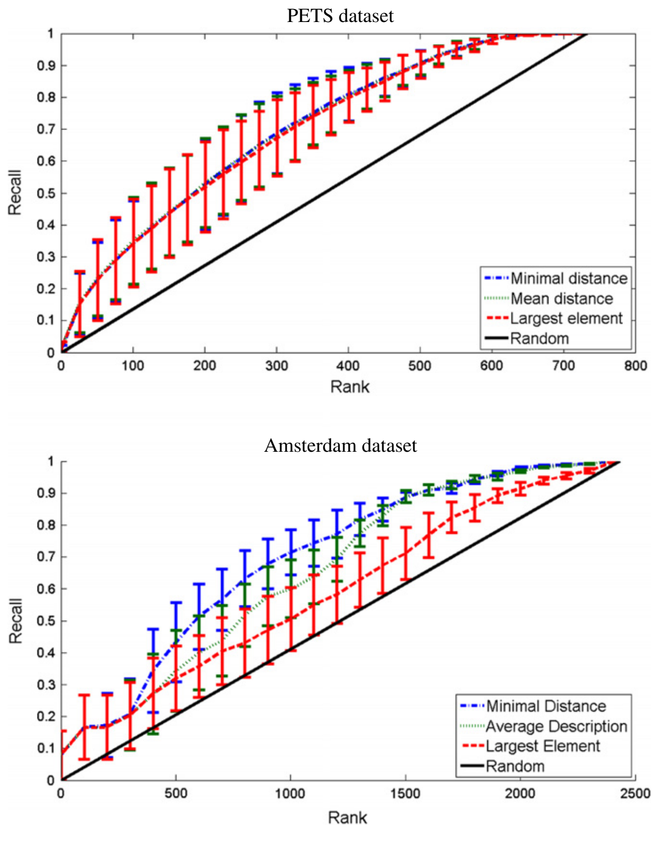


Fig. 5. Comparison of the three track matching methods on the PETS dataset and the Amsterdam dataset with variance as error bars. For both datasets we compared minimal distance, mean distance and largest element matching to a random baseline.

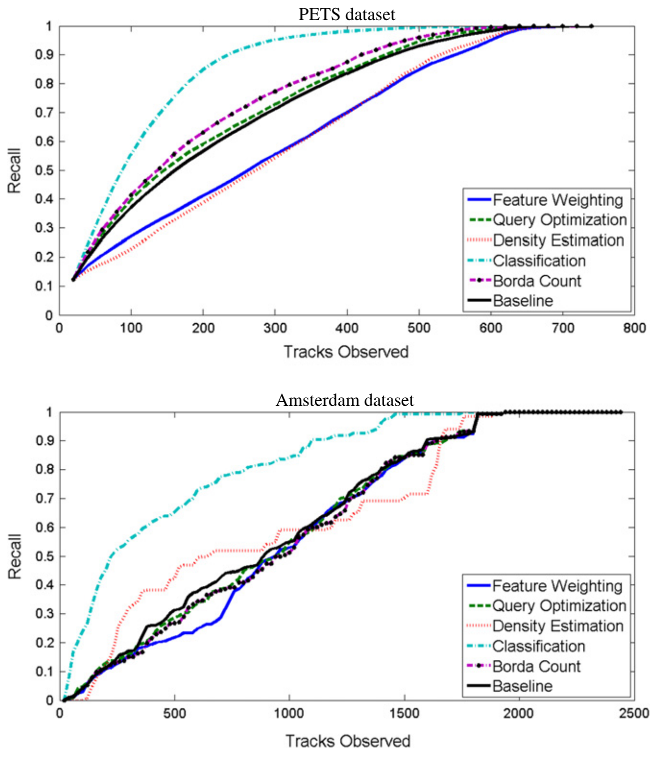


Fig. 6. Comparison of the five relevance feedback methods on both the PETS dataset and the Amsterdam dataset. For both datasets we compared dimension weighting, query point optimization, density estimation, classification and Borda count to the baseline performance of not using any user feedback.

4.4.2. Relevance Feedback

For RF the start is similar as for track matching. We again sample a random track from the ground truth and order all tracks not from the same camera based on their distance to the track. We present 20 images to the user in every iteration, i.e. $|V_i| = 20$ and simulate feedback on those 20 tracks. Using one of the RF methods discussed in Section 3 all tracks not previously observed, i.e. I_i, are re-ordered. The top 20 tracks of the newly ordered list are then given to the simulated user. This process is continued until all tracks are observed. For every query we measure the increase in recall after every round of interaction and average this result for all queries.

To compare the various RF methods we use the best performing feature descriptors and track matching method of experiment one. The baseline performance and initial ordering of all RF methods is then the result of ordering all tracks based on their distance to the query track. After manual optimization on a small number of sample queries on the PETS dataset we fix the parameters of Section 3.4 at $\alpha = 0.5$, $\beta = 0.3$ and $\chi = 0.2$ respectively. For classification we used the standard libSVM parameter settings for C, γ and ϵ [5].

5. Results

5.1. Track matching

In Fig. 5, we show the results of comparing the three track matching methods. As can easily be observed, on the PETS dataset, the results of all matching methods are similar. For the Amsterdam dataset, however, Minimal Distance Matching significantly outperforms all other methods. Since the resolution and frame-rate of camera footage is lower in the Amsterdam dataset than in the PETS dataset, the quality of detections is similarly worse. The difference in performance between methods is thus likely due to the decrease in detection quality.

As mentioned in Section 2, the quality of detections has a different impact on matching performance for each of the three methods. When using Average Description Matching a large number of errors results in a meaningless average descriptor. Both Largest Detection Matching and Minimal Distance Matching use a single detection to represent a track. This means that if that detection is not showing the person of interest the same person will probably not be found in other tracks either. However, in contrast to Largest Detection Matching, Minimal Distance Matching has less problems

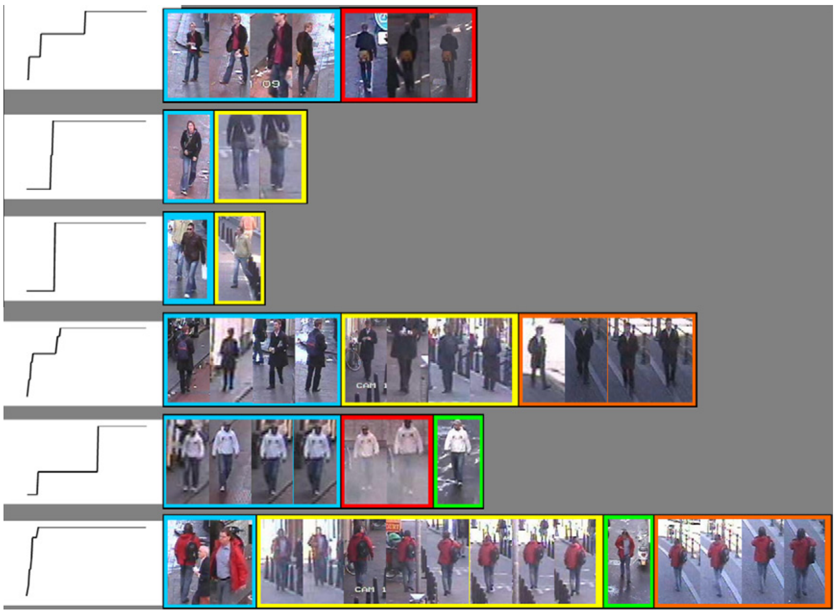


Fig. 7. Personalized relevance feedback results on the amsterdam dataset using classification. We show the results as spark lines together with a representative image region of every track that person is visible in. Detections from the same camera are shown with a similarly colored bounding box. For all persons the first track is used as query track, where all tracks in other cameras are to retrieved as fast as possible. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

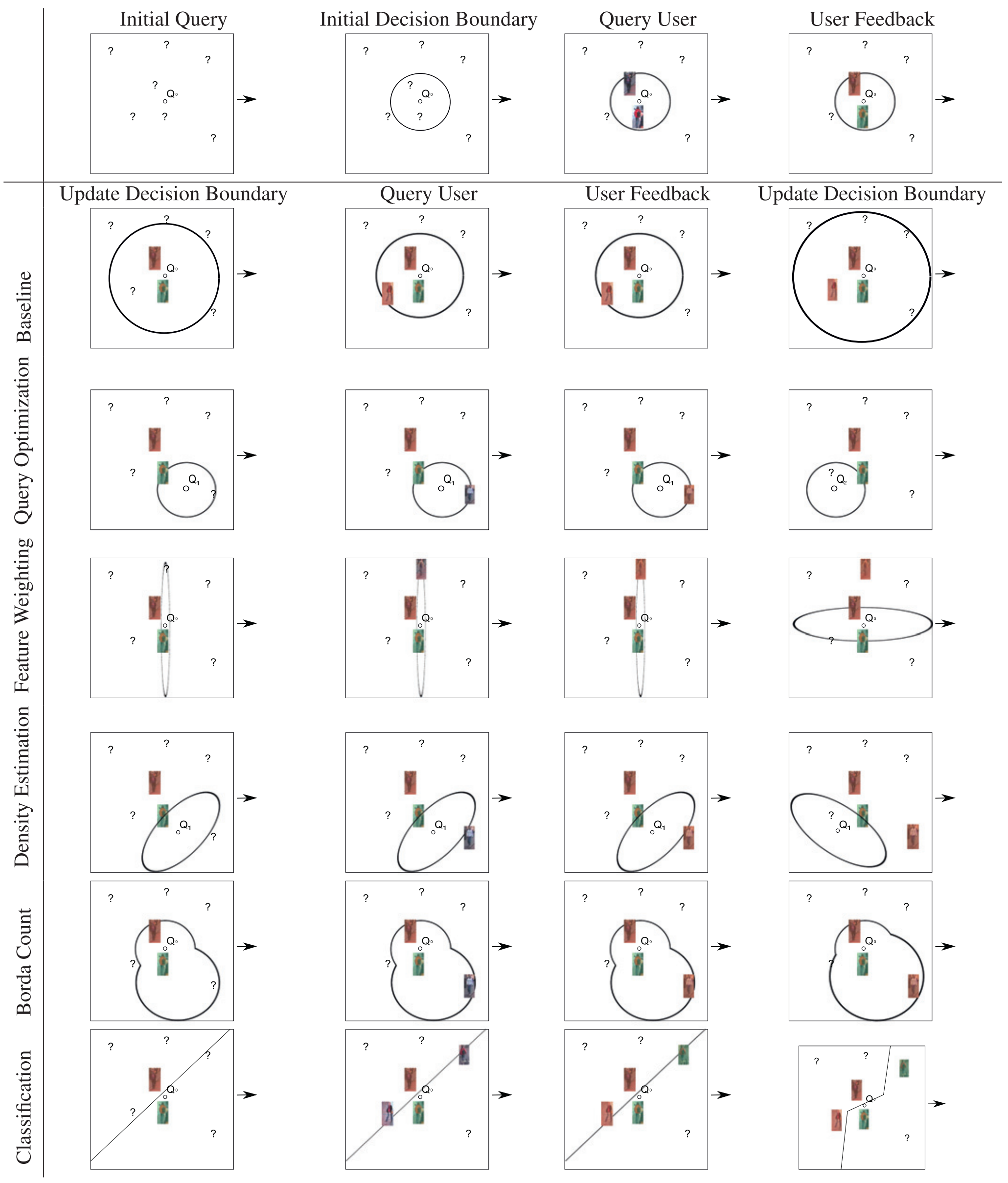


Fig. 8. Visualization of the five standard relevance feedback methods we propose in this paper. The top four images show the initial steps all methods use: Step one is to let the user define a query point which is visualized in a 2-dimensional feature space together with a set of other detections. In the second image we show the initial decision boundary based on this query where all detections within the circle are "classified" as similar to the query. In the third image we show these detections to the user and in Fig. 4 the user provides on them. In this sense a red shade indicates negative feedback (not the same person) and a green shade indicates positive feedback (same person). Afterwards, we show feedback loop for each method in its respective row. Here, "update decision boundary" means the last decision boundary is altered based on the user feedback and the method used. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

with these situations. The distance between detections showing the same person is most likely smaller than between other detections. Additionally, it is unlikely that false positives from different cameras are similar.

5.2. Relevance Feedback

We show the results of comparing the five RF methods discussed in Section 3 in Fig. 6. For Feature Weighting and Density Estimation, performance is worse than when feedback is not used. Since the feature descriptors of detections in a track are rather sparse, it is likely that some dimensions are exactly alike for all elements in S^+. In this case the variance over that dimension is zero, which leads to a disproportionately high weight. Since the distance between tracks is then effectively based on only a small subset of all dimensions, the discriminative value of the features is severely decreased. With a more balanced weighting scheme results might improve, but finding an ideal weighting scheme is dataset dependent. Query Optimization is slightly better than not using feedback, but not when compared to Borda Count. Inspection of the results shows that query optimization works well if more than one person is present in the query track. The feedback is then used to distinguish between these persons. If only a single person is present however, not much is gained or lost. Since all queries used for this experiment show only a single person, this potential improvement is not observable. In contrast, the ordering Borda Count proposes always performs better than not using any user feedback. Here, performance is especially improved when new tracks have been found that look dissimilar to the initial query. Clearly the best performing method overall for RF is Classification.

Classification based Relevance Feedback has a much higher recall than all other methods due to its completely different adaptation of the feature space. Where all other methods try to minimize the distance for tracks of interest to a single point, Classification measures the distance to a hyperplane. This means that the distance between elements is of no concern, as long as the plane is optimally positioned. In other words, RF does not need to be used to change the complete feature space such that all tracks of the same person are positioned together and all other tracks are far away. We only need to find a plane that maximizes the distance between tracks of the same persons and tracks of other persons. This simpler task then leads to a better overall tracing performance.

The RF results on the Amsterdam dataset are more difficult to interpret than the results on the PETS dataset. However, we can observe that Feature Weighting, Query Optimization and Borda Count perform similar to not using feedback. Due to the small number of positive elements and sparse feature descriptions, the weight matrix used for Density Estimation is zero. In these cases the resulting ordering is random and can therefore not be compared to the other methods. Analogous to the results on the PETS dataset, Classification based RF clearly outperforms all other RF methods. Directly after the first round of interaction, performance is better than for the baseline and keeps improving after every round of feedback.

Not surprisingly, both the matching and RF methods are dependent on the visual characteristics of the person traced. We thus show the personalized results for Classification in Fig. 7. As can be seen by the matching results for the person wearing a red jacket, this method works well if the person traced is visually distinctive. Additionally, matching benefits greatly from RF if some tracks are similar to other tracks showing the same person, but not to the query track.

6. Conclusions

In this paper we developed a track-based relevance feedback system. For this, we used an automatic tracking method to speed up and improve person tracing. To match the resulting tracks we identified three possible distance measures and compared these methods on both a benchmark dataset and a real-life dataset. We showed that if the quality of tracks is high, all methods show a similar performance. There is a difference in computational costs though, so if time is a concern we recommend either Largest Element Matching or Average Distance Matching. If the quality of tracks is low, matching based on the minimal distance clearly outperforms other methods.

To evaluate Relevance Feedback we compared five representative feedback methods; Borda Count, Feature Weighting, Density Estimation and Classification. We showed that Classification based RF clearly outperforms all other methods, improving recall by up to 400% when compared to not using RF or any other method. We also proposed an interaction loop capable of dealing with a different object of interaction when compared to traditional RF methods. In our system, the user gives feedback on complete tracks instead of on single detections which simplifies the interaction significantly.

In conclusion, track based Relevance Feedback can considerably improve the efficiency of tracing people in surveillance settings.

Acknowledgments

This work is partially sponsored by FES Camera 3D. The used real-life surveillance dataset was provided by the Netherlands Forensic Institute (NFI) in conjunction with the Dutch police, for which we would like to thank Jurrien Bijhold in particular.

References

[1] PETS Benchmark Data, 2009. <www.cvg.rdg.ac.uk/WINTERPETS09/a.html>.
[2] A. Alahi, P. Vandergheynst, M. Bierlaire, M. Kunt, Cascade of descriptors to detect and track objects across any network of cameras, CVIU '10 114 (6) (2010) 624–640.
[3] S. Ali, O. Javed, N. Haering, T. Kanade, Interactive retrieval of targets for wide area surveillance, in: ACM Multimedia, MM '10, 2010, pp. 895–898.
[4] C. Buckley, G. Salton, J. Allan, The effect of adding relevance information in a relevance feedback environment, in: ACM SIGIR Conf. on Res. and Dev. in IR, 1994, pp. 292–300.
[5] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, 2001. <http://www.csie.ntu.edu.tw/cjlin/libsvm>.
[6] N. Dalal, INRIA Person Dataset, 2005. <http://pascal.inrialpes.fr/data/human/>.
[7] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, CVPR '05 2 (2005) 886–893.
[8] J.C. de Borda, Memoire sur les Elections au Scrutin, Histoire de l'Academie Royale des Sciences, Paris, 1781.
[9] R. den Hollander, S. Landsmeer, A. van Eekeren, H. Bouma, Appearance based retrieval of persons in digital images of multiple cameras, Technical Report TNO-DV-2009-D651, TNO, 2010.
[10] M. Farenzena, L. Bazzani, A. Perina, V. Murino, M. Cristani, Person re-identification by symmetry-driven accumulation of local features, in: CVPR '10, 2010.
[11] D. Gray, S. Brennan, H. Tao, Evaluating appearance models for recognition, reacquisition, and tracking, in: PETS '07, 2007.
[12] D. Gray, H. Tao, Viewpoint invariant pedestrian recognition with an ensemble of localized features, in: ECCV '08, 2008.
[13] C. Harman, Relevance feedback revisited, in: ACM SIGIR Conf. on Res. and Dev. in IR, 1992, pp. 1–10.
[14] Y. Ishikawa, R. Subramanya, C. Faloutsos, Mindreader: querying databases through multiple examples, in: VLDB, 1998, pp. 218–227.
[15] P. Koppen, M. Worring, Multi-target tracking in time-lapse video forensics, in: MiFor '09, 2009, pp. 61–66.
[16] Z. Lin, L.S. Davis, Learning pairwise dissimilarity profiles for appearance recognition in visual surveillance, in: ISVC '08, 2008, pp. 23–34.
[17] J. Meessen, X. Desurmont, J.-F. Delaigle, C. De Vleeschouwer, B. Macq, Progressive learning for interactive surveillance scenes retrieval, in: CVPR'07, 2007.
[18] M.J. Metternich, M. Worring, A.W.M. Smeulders, Color based tracing in real-life surveillance data, in: Transactions on Data Hiding and Multimedia Security V, 2010, pp. 18–33.
[19] J. Rocchio, Salton: the Smart Retrieval System: Experiments in Automatic Document Processing, Chapter: Relevance Feedback in Information Retrieval, Prentice-Hall, 1971, pp. 313–323.
[20] T. Rose, J. Fiscus, P. Over, J. Garofolo, M. Michel, The trecvid 2008 event detection evaluation, in: WACV, 2009, pp. 1–8.
[21] G. Salton, C. Buckley, Improving retrieval performance by relevance feedback, Journal of the American Society for Information Science 41 (1990) 288–297.
[22] R. Schettini, G. Ciocca, I. Gagliardi, Content-based color image retrieval with relevance feedback, in: ICIP (3), 1999, pp. 75–79.
[23] R. Schuster, R. Mörzinger, W. Haas, H. Grabner, L. Van Gool, Real-time detection of unusual regions in image streams, in: ACM Multimedia, MM '10, 2010, pp. 1307–1310.
[24] A.W.M. Smeulders, M. Worring, S. Santini, A. Gupta, R. Jain, Content-based image retrieval at the end of the early years, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (2000) 1349–1380.
[25] K.E.A. van de Sande, T. Gevers, C.G.M. Snoek, Evaluating color descriptors for object and scene recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 32 (9) (2010) 1582–1596.
[26] V.N. Vapnik, The Nature of Statistical Learning Theory, Springer-Verlag New York, Inc.,, 1995.
[27] W. Wang, Z. Liu, Automatic event detection of abandoned and stolen objects in real-time surveillance, in: International Conference on Image Processing and Pattern Recognition in Industrial Engineering, 2010.
[28] C. Zhang, W.-B. Chen, X. Chen, L. Yang, J. Johnstone, A multiple instance learning and relevance feedback framework for retrieving abnormal incidents in surveillance videos, Journal of Multimedia 5 (4) (2010).
[29] X.S. Zhou, T.S. Huang, Relevance feedback in image retrieval: a comprehensive review, Multimedia Systems 8 (2003) 536–544.
[30] Z. Zivkovic, F. van der Heijden, Efficient adaptive density estimation per image pixel for the task of background subtraction, Pattern Recognition Letters 27 (2006) 773–780.