Multi-Pose Face Recognition for Person Retrieval in Camera Networks

Martin Bäuml Keni Bernardin Mika Fischer Hazım Kemal Ekenel
Rainer Stiefelhagen

Institute for Anthropomatics, Karlsruhe Institute of Technology
{baeuml, keni.bernardin, mika.fischer, ekenel, rainer.stiefelhagen}@kit.edu

Abstract

In this paper, we study the use of facial appearance features for the re-identification of persons using distributed camera networks in a realistic surveillance scenario. In contrast to features commonly used for person re-identification, such as whole body appearance, facial features offer the advantage of remaining stable over much larger intervals of time. The challenge in using faces for such applications, apart from low captured face resolutions, is that their appearance across camera sightings is largely influenced by lighting and viewing pose. Here, a number of techniques to address these problems are presented and evaluated on a database of surveillance-type recordings. A system for online capture and interactive retrieval is presented that allows to search for sightings of particular persons in the video database. Evaluation results are presented on surveillance data recorded with four cameras over several days. A mean average precision of 0.60 was achieved for inter-camera retrieval using just a single track as query set, and up to 0.86 after relevance feedback by an operator.

1. Introduction

The objective is to find occurrences of a particular person, selected by an operator, within video footage captured by multiple cameras with possibly disjoint fields of view.

Possible questions that could be answered by querying such a system are: *Where was this person during the last 10 minutes?* or *How often did this person enter the building during the last 7 days?* These two questions already indicate two important requirements: First, the system should be able to run a query on very recently recorded video (”the last 10 minutes”), i.e. time-expensive preprocessing of the recorded videos is impractical. Second, the system should cope with changes in the appearance of persons over several days, hence it cannot depend for example on the color of a person's clothes. Additionally, it is desirable that such queries can be carried out fast, allowing for explorative investigation of the data, which typically means that results should be available within 5 to 10 seconds.

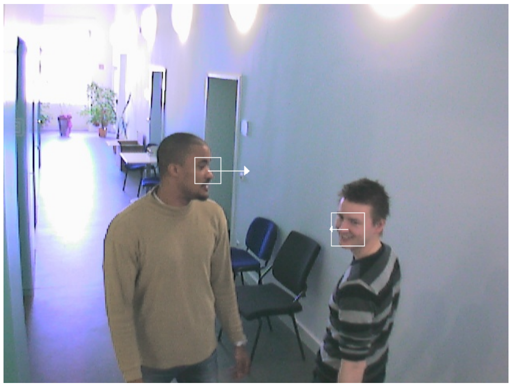

Figure 1: Surveillance image from one of our four cameras with tracked faces and estimated head pose. Tracking helps to associate difficult views such as the side-view of the left person with other views, allowing to gather diverse training data for retrieval.

Using a person's facial appearance for retrieval has one decisive advantage: It allows to search videos that have been recorded on different days, as it does not rely on the assumption that a person wears the same clothing throughout the observation period. It does, however, also pose a number of challenges, as in surveillance video faces are usually of low resolution and exhibit large variations in pose and illumination.

In our approach we can deal with face sizes as small as 18×18 pixels and out-of-plane rotations of up to 60 degrees. We track faces in real time across pose changes in order to increase the amount of available training data and also correctly identify persons by means of temporal association, even when recording conditions are unfavorable for multiple frames. The main contributions of this paper are: (i) It offers a quantitative evaluation of facial feature-driven person retrieval in realistic surveillance scenarios. (ii) It investigates the explicit incorporation of head pose information extracted during tracking in the retrieval task. (iii) It presents a real-time capable system to retrieve occurrences of persons in multiple surveillance camera views, independent of short-term cues such as color and texture of clothing.

We evaluate our approach on surveillance footage recorded over three whole days using four cameras in the corridors of our lab.

1.1. Related work

In security related research, usually color and/or texture of the clothes are the most important cues for person retrieval [6, 7, 8]. However, in general this does not allow to search in videos collected over several days since people tend to change clothes between days. Facial features on the other hand have mainly been used to identify persons from a closed set of possible choices [1, 13].

Research on facial feature-driven person retrieval has been conducted mainly in the area of multimedia analysis for applications such as fast video browsing and character search. The advantages of multimedia data are obvious: The quality of the images and faces is consistenly rather good because of professional lighting and close-up shots. Furthermore, even in movies the number of main characters is usually limited. Arandjelović and Zisserman [2] retrieve faces in feature-length films starting from one or more query images. A pose normalization step is included in order to deal with slightly non-frontal faces. Sivic *et al.* [12] employ a generic region tracker to associate faces into longer tracks and then match face tracks instead of single images. Each face track is described by a bag-of-words-like histogram which is later used for matching against other tracks. However, their tracker is computationally expensive and not suited for real-time face tracking. Similarly, Fischer *et al.* [5] use a face tracker in order generate longer face tracks. In an offline pre-processing step, frame-to-frame distances between all tracks are pre-computed. Using the global nearest-neighbour map, retrieval and enlarging of the query-set are performed simultaneously. This approach however inhibits real-time querying of recently added tracks because of the necessary pre-processing. Both [12] and [5] use a generic distance function for matching face sets, while in this work a discriminative classifier is trained.

2. Retrieval system

In this section we briefly describe the tracker which connects face appearances of the same person into consecutive tracks, and the facial feature extraction.

2.1. Face tracking

The tracker builds upon a generic face detector using the modified census transform (MCT) as feature. Our implementation of the face detector follows the approach in [9]. In order to associate face detections from the same person over time and to achieve real-time performance, we embed the detector in a particle filter framework.

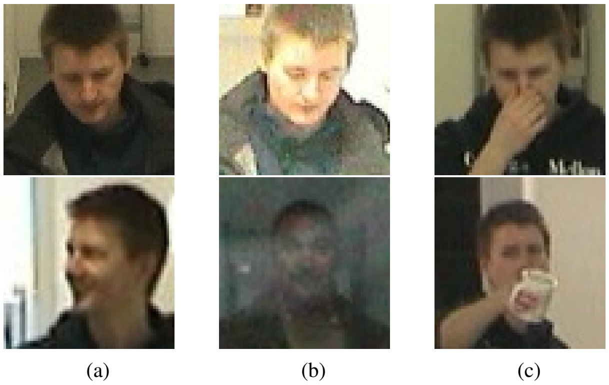

Figure 2: Surveillance data usually contains challenging image conditions for face recognition. Our collected data includes (a) unconstrained head poses (b) varying illumination and (c) occlusions of the face.

2.1.1 Face detectors

Following the approach in [9], our face detector is a cascade of boosted MCT feature histograms. There are two advantages in using this approach: First, the MCT features make the detector robust against illumination changes. Second, it is relatively fast, both in training and in actual detection. In order to detect faces at various yaw angles, we train multiple detectors at 0, 15, 30, 45, and 60 degrees out-of-plane rotation. Additionally, we mirror the detectors to the respective negative yaw angles. Each detector has four stages.

2.1.2 Pose-tracking particle filter

Running all detectors exhaustively on each frame would be too slow for achieving real-time performance. Thus, the detectors are integrated in a particle filter in order to evaluate them only at locations likely to contain a face, i.e. around the locations where a face has been in the last frame. We use one particle filter for each tracked face (consisting each of a weighted set of 2000 particles in our experiments). The state

$$\boldsymbol{x} = \{x, y, s, \alpha\}$$

of each particle consists of the 2D image location of the face (x, y), its size in image pixels s and its yaw angle α.

Propagation. The particles are propagated by means of independently drawn noise from normal distributions. We do not employ an explicit velocity model since our experiments showed that it is not needed in practice.

Observation models. For updating the particles' weights ω_i, we evaluate at each particle's location the detector that has the lowest angular distance between the particle's pose angle α and the detector's trained angular class γ. The detector provides a confidence value of the detection in form of its stage sums. These stage sums are directly used as weight update, but only if all stages of the detector cascade have been passed:

$$D_\gamma = \underset{\gamma}{\operatorname{argmin}}(\alpha - \gamma), \qquad \gamma \in \{-60, -45, \ldots, +60\} \tag{1}$$

$$\omega_i = \begin{cases} 0 & \text{if} \quad n < n_{max} \\ \sum_{k=1}^{n_{max}} H_{\gamma k}(\boldsymbol{x}) & \text{if} \quad n = n_{max} \end{cases} , \tag{2}$$

where $H_{\gamma k}(\boldsymbol{x})$ is the kth stage sum of the cascade of detector D_γ and n is the number of passed stages. We found that the performance of the tracker deteriorated significantly when we also used the confidence of detectors that did *not* pass all stages ($n < n_{max}$), most likely because of the small number of stages. By selecting the detector with the best matching yaw angle, the particles whose pose angles best describe the current face pose are assigned the highest weights.

Automatic track initialization and termination. Every k frames ($k = 5$ in our experiments), we scan the whole frame with the frontal, ±30 and ±60 degree face detectors to automatically initialize new tracks. The value of k trades off the average time until a new face is detected versus the speed of the tracker. A new track is initialized if more than three successive detections are spatially close to each other, but far enough from any known face track. A track is terminated when no detection was achieved during particle scoring (i.e. at none of the particle's locations the detector passed all stages) for more than 5 frames.

Occlusion handling. If there is more than one person in the video, care has to be taken that particles from different trackers do not coalesce onto one target. As mentioned above, in the case of multiple persons, one particle filter is used for each person/face. We ensure that particles from one track do not accidentally get scored on the face of another track by making the track centers ”repel” particles from other tracks: A particle's score is set to zero if its distance to another track's center $\overline{\boldsymbol{X}}_{\boldsymbol{i}}$ is smaller than a threshold:

$$\|\boldsymbol{x} - \overline{\boldsymbol{X}}_{\boldsymbol{i}}\| < \theta \, .$$

Again, this simple method works well in practice. However, in contrast to more elaborate occlusion models [10], a track cannot survive when largely occluded by another track. When an occluded face becomes visible again, it will be reinitialized as soon as it is far enough from the occluding face. This results in two disconnected tracks from the same person. At present, we do not explicitly merge those disconnected tracks.

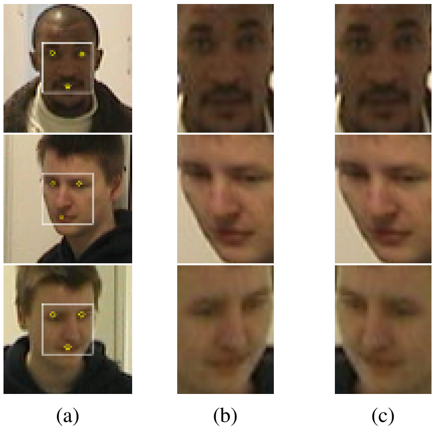

Figure 3: The effect of pose alignment and mirroring depending on head pose. (a) Original image with detected face and detected/estimated eye and mouth positions. (b) Aligned and cropped face w/o considering the head pose. (c) Aligned face after mirroring faces to positive yaw angles.

2.2. Feature extraction

Given the location of a face in a frame (as output of the tracker), we perform several steps to transform the face image into a feature vector.

Pose alignment. Pose alignment consists of three steps. In the first step, we use the pose information as determined by the tracker to mirror all non-frontal views with negative yaw angles to positive yaw angles. This assumes that a person's appearance is roughly symmetric.

Second, we try to localize the eyes and mouth, using eye and mouth detectors which are - similar to the face detectors - boosted cascades of MCT feature histograms. If the eyes cannot be detected, which might happen relatively often since the eye detectors were not specifically trained for this scenario, we again use the pose information provided by the tracker to roughly estimate the eye locations. In order to estimate the eye and mouth locations from the detector's pose class, we ran our pose-dependent face detectors on the FERET database [11] and computed the mean eye and mouth locations for each pose class from the labeled ground truth.

Third, given either the detected or estimated locations of eyes and mouth, we warp and crop the face with an affine transformation to a normalized pose of size 48 × 64 pixels such that the eyes and mouth are at fixed locations.

Figure 3 displays the effect of the alignment step. The third column shows the advantage of using the pose information in order to mirror all images to positive yaw angles. The variation of the images before feature extraction is reduced and thus a possibly more effective person model can be trained later.

Feature extraction. From the aligned image, we compute a feature vector according to the method in [4] which has proven to provide a robust representation of the facial appearance in real-world applications [3, 13]. In short, the aligned face is divided into non-overlapping blocks of 8×8 pixels resulting in 48 blocks. On each of these blocks, the 2-dimensional discrete cosine transform (DCT) is applied and the resulting DCT coefficients are ordered by zig-zag scanning (i.e. $c_{0,0}, c_{1,0}, c_{0,1}, c_{0,2}, c_{1,1}, c_{2,0}, \ldots$). From the ordered coefficients, the first is discarded for illumination normalization. The following 5 coefficients from all blocks, respectively, are concatenated to form the facial appearance feature vector ($5 \times 48 = 240$ dimensional). See [4] for details.

Both pose normalization and feature extraction are computationally cheap operations and can be performed online during tracking.

3. Retrieval using face tracks

A retrieval is started by selecting one or more tracks of one individual as *query set*. In a live system, this can be either a track from the track database, or even a currently running track, since pose estimation and feature extraction are conducted along-side with the tracking.

The retrieval is conducted by first assigning a matching score (or confidence value) to all possible tracks in the database, representing how well these tracks match the track(s) from the query set. The results are then ranked by score and all results with a score above a threshold θ are reported to the user.

In our approach, we compute the matching score by training a person-specific classifier using the features from the query set and then running the classifier on each frame of every possible track in the track database.

3.1. Model training

Given the track(s) from the query set, we train a person-specific classifier to perform the scoring. We use a Support Vector Machine (SVM) in our experiments, but in principle every discriminative classifier that provides a confidence value for the classification can be used.

In contrast to identification scenarios, we do not have a set of negative training examples from our track database against which we can train the SVM. After all, every track in the track database could be from the person we are looking for. Instead, we use a set of features from independently recorded persons of which we assume that they do not appear in the videos. This is our *unknown set* which is used as negative set for training the SVM. In our experiments we use as little as 1-2 tracks from 15 unknown persons, resulting in about 2000 negative features. We argue that for every scenario it should be possible to build such a small negative set of people that will certainly never be searched for in a retrieval operation.

In order to balance the amount of positive and negative training data, we subsample from the negative set so that we only have roughly twice as many negative examples as positive examples.

3.2. Track scoring

A track is scored by evaluating the previously trained classifier on each of the track's frames. The resulting confidence values of the classifiers (for the SVM: distance to hyperplane d_{SVM}) are averaged to a track score

$$s_{track} = \frac{1}{N} \sum_{i=1}^{N} d_{SVM}(\boldsymbol{x_i}) \tag{3}$$

where $\boldsymbol{x_i}$ is the facial feature vector of frame i.

3.3. Interactive feedback

Due to the challenging conditions in surveillance data, we cannot expect the results of fully automatic search to be perfect. After presenting the retrieval results, we allow the user to give feedback on the top three (with the highest scores) and bottom two results. This allows for fast retraining with labeling only 5 of the result tracks. We then retrain the model using these additional samples and run the retrieval again.

The feedback on the top results allows to update the classifier with (i) additional samples of the searched person and (ii) hard samples of a different person, which the classifier previously ranked much to high. We present the bottom results instead of additional top results to consistenly enlarge the set of negative training data as well. Remember that we initially started with around 2000 negative samples, which is fine for the initial retrieval, since one single track in our database typically consists of between 25 and 200 frames. However, with the interactive feedback, the number of positive samples increases. With the additional negative tracks we ensure that we keep enough negative samples to balance positive and negative training data.

This process can be continued for several rounds.

4. Evaluation and results

We evaluate our approach on a dataset of surveillance footage recorded by four cameras in the corridors of our lab

	# tracks	# persons	# persons with $\geq$ 10 tracks (# total tracks)
Cam 1	67	10	1 (24)
Cam 2	492	70	10 (348)
Cam 3	139	28	3 (49)
Cam 4	99	23	5 (64)
Total	797	92	18 (604)

Table 1: Statistics of our dataset. The large number of persons and tracks in camera 2 is due to an open house event in our lab where only camera 2 could capture the visitors. The number in brackets in the the third column denotes the number of tracks for which the persons with at least 10 tracks account for.

over a timespan of three days. In total, our tracker found 797 tracks of 92 different persons (cf. Table 1). There are 18 persons in our database who have at least 10 tracks each. These 18 persons account for a total of 604 tracks. For the experiments, we only use the tracks of these 18 persons as initial query set in order to be able to calculate meaningful recall-precision curves. Nevertheless, the whole set of tracks of all persons including those with $<$ 10 tracks (but excluding the initial query track) is used as retrieval dataset.

The performance of the retrieval is evaluated in form of mean average precision (MAP) and recall and precision. The mean average precision is calculated as the mean of the average precisions (AP) over all retrievals, where the average precision of a ranked list of tracks is

$$AP = \frac{1}{\sum_{i=1}^{N} r_i} \sum_{i=1}^{N} r_i \left(\frac{\sum_{j=1}^{i} r_j}{i} \right) \tag{4}$$

with r_i being 1 if the ith entry in the list is a track of the searched person and 0 otherwise. N is the total number of tracks.

Recall and precision are defined as

$$recall = \frac{\sum_{i=1}^{R_\theta} r_i}{\sum_{i=1}^{N} r_i} = \frac{\text{\# correctly retrieved tracks}}{\text{\# tracks of person}} \tag{5}$$

$$precision = \frac{\sum_{i=1}^{R_\theta} r_i}{R_\theta} = \frac{\text{\# correctly retrieved tracks}}{\text{\# retrieved tracks}} \tag{6}$$

where R_θ is the cut-off rank for a given score threshold θ.

We performed two sets of experiments. First, we evaluate the *intra-camera retrieval* performance on the data captured by camera 2. In this case, additional difficulties arising from different camera types and camera placement are avoided. The second set of experiments uses the whole dataset of all four cameras for evaluating *inter-camera retrieval*. In all experiments, we use a SVM with a polynomial kernel of degree 2 with $C = 0.03125$. The results in terms of mean average precision are given in Table 2. Although there is only a slight improvement by mirroring the training images according to their pose, the improvement is consistent over all experiments.

	intra-camera		inter-camera		
	w/o	w/	w/o	w/	mirroring
Initial	0.72	0.73	0.59	0.60	
Retrain Round 1	0.83	0.84	0.71	0.72	
Retrain Round 2	0.87	0.88	0.77	0.78	
Retrain Round 3	0.91	0.92	0.82	0.83	
Retrain Round 4	0.93	0.94	0.85	0.86	

Table 2: Mean average precisions for intra- and inter-camera retrieval.

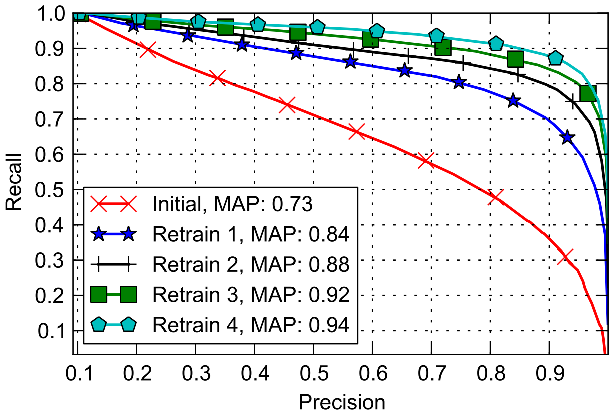

Figure 4: Intra-camera retrieval results. Relevance feedback for as little as 5 tracks per round can help to improve the results significantly.

4.1. Intra-camera retrieval

The intra-camera retrieval experiments are performed using the 348 tracks of the 10 persons with at least 10 tracks in camera 2 as initial query sets. The corresponding recall-precision curves are shown in Figure 4. At a precision of 90%, the recall is about 36% without retraining, i.e. just training the model with the initial track only. The mean average precision is 0.73.

When we present the operator of the system with as few as five result tracks per retraining round as described in Section 3.3 we can achieve a recall of 88% at a precision of 90% after only 4 retraining rounds. This suggests that the system can benefit a lot from the additional training samples and hard negative samples.

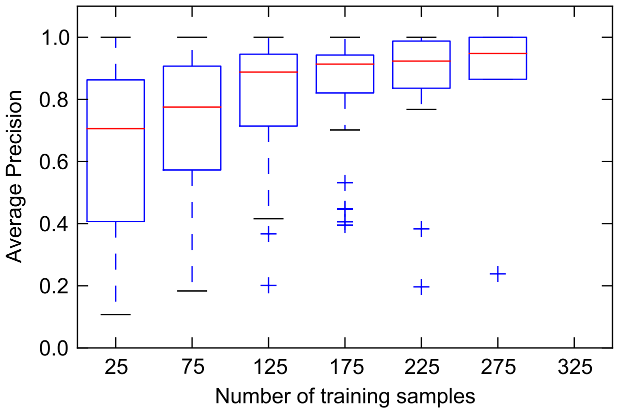

Figure 5: The retrieval performance is highly dependent on the number of training samples available to train the model. This boxplot shows the performance for intra-camera retrieval for the initial retrieval (no retraining). The boxes denote the median and lower and upper quartiles of the data in bins of size 50. The whiskers represent the full range of the data.

Actually, the performance of the initial retrieval is also highly dependent on the number of available training samples. The boxplot in Figure 5 shows, that the performance is on average significantly higher if the initial query set contains more training samples, i.e. the track is longer. This justifies the effort of building a good tracker which is capable of attaining long continuous tracks, which especially means tracking over non-frontal head poses.

4.2. Inter-camera retrieval

Inter-camera retrieval has to deal with additional challenges. We used two different kinds of cameras for the recordings, yielding images of different quality. For example, two of our cameras had an uncorrectable gain in the blue channel. In this way, our dataset probably reflects the reality in actual surveillance applications quite well, where no guarantees can be given that all cameras are of the same type or all yield good quality images.

As could be expected, the performance is negatively affected by these additional challenges. However, for the initial query we still achieve a mean average precision of 0.60 or 17% recall at 90% precision. After 4 retraining rounds, the recall at 90% is increased to 66% with a MAP of 0.86. The feedback helps in this case to overcome the inter-camera differences by adding additional training samples from all cameras.

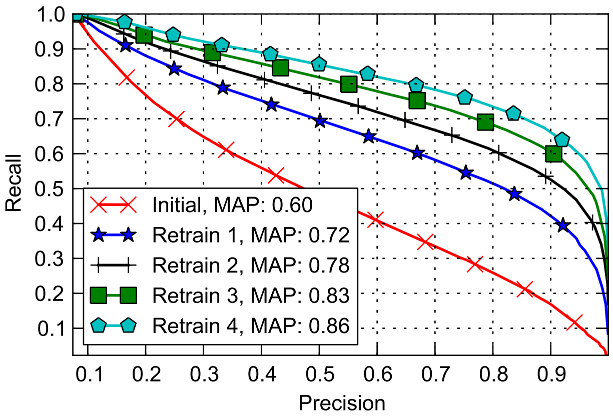

Figure 6: Precision-recall curve for the inter-camera retrieval experiment.

5. Conclusions and future work

We have demonstrated that person retrieval using facial features is feasible in realistic environments. This is possible by employing a robust face tracker to connect individual face instances over time and non-frontal head poses. We demonstrated how the explicity tracked head pose can be used for facial feature alignment and pose normalization. The retrieval is performed by using a discriminative classifier trained with the query set against a set of independently collected features. The retrieval results are improved by feedback by an operator on as few as five results.

On a realistic and difficult dataset recorded in our lab, we reported promising results. A mean average precision of 0.60 was achieved for inter-camera retrieval using just a single track as query set, and up to 0.86 after four rounds of relevance feedback by an operator.

In future work we plan to integrate the face tracker with a face-independent person tracker in order to further associate face tracks over head poses where the face is not present. Additionally, we will investigate how to improve the retrieval by combining short-term features, such as from clothing, with facial features while maintaining the possibility to search videos recorded on different days.

Acknowledgements This work was realized as part of the Quaero Programme, funded by OSEO, French State agency for innovation.

References

[1] N. Apostoloff and A. Zisserman. Who are you? Real-time person identification. In *Proceedings of the 18th British Machine Vision Conference*, pages 509–518, 2007. 2

[2] O. Arandjelovic and A. Zisserman. Automatic face recognition for film character retrieval in feature-length films. In

IEEE Conference on Computer Vision and Pattern Recognition, volume 1, page 860. IEEE, 2005. 2

[3] H. Ekenel, Q. Jin, M. Fischer, and R. Stiefelhagen. ISL Person Identification Systems in the CLEAR 2007 Evaluations. In *Proceedings of the International Evaluation Workshops CLEAR 2007*, page 256265, 2008. 4

[4] H. Ekenel and R. Stiefelhagen. Analysis of Local Appearance-Based Face Recognition: Effects of Feature Selection and Feature Normalization. *Conference on Computer Vision and Pattern Recognition Workshop*, pages 34–34, 2006. 4

[5] M. Fischer, H. Ekenel, and R. Stiefelhagen. Interactive Person Re-Identification in TV series. In *International Workshop on Content Based Multimedia Indexing, CBMI*, pages 219–224, Grenoble, France, 2010. 2

[6] T. Gandhi and M. M. Trivedi. Person tracking and reidentification: Introducing Panoramic Appearance Map (PAM) for feature representation. *Machine Vision and Applications*, 18(3-4):207–220, 2007. 2

[7] N. Gheissari, T. B. Sebastian, and R. Hartley. Person Reidentification Using Spatiotemporal Appearance. *IEEE Conference on Computer Vision and Pattern Recognition*, 2:1528–1535, 2006. 2

[8] O. Hamdoun, F. Moutarde, B. Stanciulescu, and B. Steux. Person re-identification in multi-camera system by signature based on interest point descriptors collected on short video sequences. *ACM/IEEE International Conference on Distributed Smart Cameras*, pages 1–6, 2008. 2

[9] C. Küblbeck and A. Ernst. Face detection and tracking in video sequences using the modified census transformation. *Image and Vision Computing*, 24(6):564–572, 2006. 2

[10] O. Lanz. Approximate Bayesian multibody tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1436–49, September 2006. 3

[11] P. Phillips, H. Moon, S. Rizvi, and P. Rauss. The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(10):10901104, 2000. 3

[12] J. Sivic, M. Everingham, and A. Zisserman. Person spotting: video shot retrieval for face sets. In *International Conference on Image and Video Retrieval*, page 226. Springer Verlag, 2005. 2

[13] J. Stallkamp, H. K. Ekenel, and R. Stiefelhagen. Video-based Face Recognition on Real-World Data. *IEEE International Conference on Computer Vision*, 206:1–8, Oktober 2007. 2, 4