How to Describe Face Sequences for Fast Person Recognition

Christian Herrmann

Vision and Fusion Laboratory
Institute for Anthropomatics
Karlsruhe Institute of Technology (KIT), Germany
christian.herrmann@kit.edu

Technical Report IES-2013-02

Abstract: The evaluation of video material for forensic purposes is a time intensive and complex work. A common task is to identify or find persons in video footage. Computer vision based methods can help to reduce the manual effort. However, video databases in forensic applications are often rather large. This poses harsh requirements with respect to the processing speed of any automated recognition approach. Specifically, searching for persons needs to be much faster than real time. An analysis and evaluation of existing face recognition techniques is performed with respect to this requirement. Based on this result a promising approach is presented. The key concept is to use a cascade of the existing techniques and combine them in a way that the advantages of each one are used. This results in a significant speedup in processing time and additionally in a slight improvement in the recognition performance. Using this approach promises to help at the forensic search in video footage.

1 Introduction

With the increasing availability of video data in all kinds of shape, the interest in automatic analysis grows. Content-based video search is relevant in a wide area of applications. Ranging from sorting private holiday videos to professional analysis of surveillance material. A key interest is the search for known persons in the video data. As the human face is a discriminative feature for identity, the use of automated face recognition is useful for this task. The focus of this report is the forensic analysis of surveillance footage. Computer vision support promises to speed up investigations which are based on video material. Compared to usual au-

Figure 1.1: Typical images from surveillance video. Containing several challenges like low resolution, different head poses, motion blur and noise.

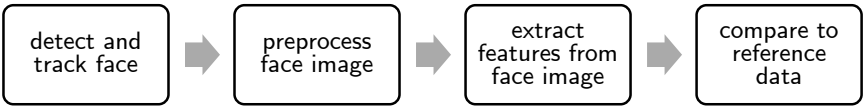


Figure 1.2: Flowchart showing the basic steps of a face recognition system.

tomated face recognition the main challenges for surveillance videos (see Fig. 1.1 for some sample images) are:

- *Unconstrained environment* – Head position, illumination and facial accessories may vary from video to video.
- *Large database* – The database may contain hours or days of video footage.
- *Low resolution* – Face sizes are typically well below 50 pixels.

In this report, the focus is on the large database while the two other challenges remain as side conditions. A research about the processing speed of different existing approaches is performed, and possibilities to address the problem are discussed.

The typical workflow for a face recognition system is shown in Fig. 1.2. In this report the first two steps are not considered. We assume that the face detection and tracking as well as a possible preprocessing is already done. The analysis will concentrate on the last two steps. Namely the feature extraction and the comparison of the extracted features. The wide variety of existing approaches will be discussed with respect to the main goal: fast recognition.

2 Problem Definition

Usually, in the field of automated face recognition two basic scenarios are known [PGM11]:

- *Verification / Authentication* – A reference identity is claimed and a sample face is provided. The task is to check if the sample face belongs to the

claimed identity. This task requires a binary answer. A typical scenario is at border control where the sample face should be compared to a reference identity given by the passport.

- *Identification / Recognition* – A sample face is given and the task is to determine the most likely identity out of a predefined set of identities. This set is usually represented by a gallery which contains reference data for the face of each identity. This task requires an integer answer. A typical scenario is the recognition of a character in a movie.

Usually, automated face recognition compares the sample face to the reference face and calculates a score which measures the similarity between the faces. In the identification scenario the identity with the highest score is the result. For the verification a comparison of the score to a threshold is necessary and the verification is accepted if the score exceeds this threshold.

For the analysis of the introduced scenario of forensic analysis a few definitions are necessary. A video will be denoted by V and contains a sequence of F frames f: $V = (f_1, ..., f_F)$. One frame is an image vector of dimension d: $f \in [0,1]^d$. A collection of B videos is denoted by $C = \{V_1, ..., V_B\}$. Each video V shows the face of exactly one person. Thus there exists a mapping $M : V \mapsto id$, where id is one identity in the set of M identities $I = \{id_1, ..., id_M\}$.

If the scenario of forensic analysis must be matched to one of the two previously defined scenarios, it can be understood as identification task (Fig. 2.1(a)). For each video in the database C, it must be checked, if it contains the requested person: $M(V_b) \stackrel{?}{=} id_{wanted}$ However, a threshold is necessary to generate the binary answer for each database video.

A different approach to look at the task of forensic analysis is the way of information retrieval. Given a large database of information, in this case video data C, a specific information should be found by some query information (Fig. 2.1(b)). Here, the wanted information R are all videos showing the specified identity from the query: $R = \{V_b \in C | M(V_b) = id_{wanted}\}$. By defining the problem like this, a binary decision can be avoided, and the usage of a threshold is obsolete. Instead, it is sufficient to rate the likelihood for each video in the database that it shows the specified identity. This results in a sorted list of the database videos with the most similar ones to the query pattern at the top of the list.

It should be noted, that the perception as information retrieval task is different from the recognition scenario. In recognition, a gallery G contains a well defined and previously built set of data where the identity for each entry is clear. This means that the mapping $M : G \to I$ is known and used to categorize the videos

(a) (b)

Figure 2.1: Green boxes represent database videos and the blue box represents a reference video. (a) Identification performs an identification task for each database sample video to the reference video. (b) Retrieval poses one search request to the database with the reference video as search pattern.

by identity. For the database C in the presented scenario this is not true. There might be several videos V_i of one identity id_m in the database. But the information that the V_i belong together does not exist.

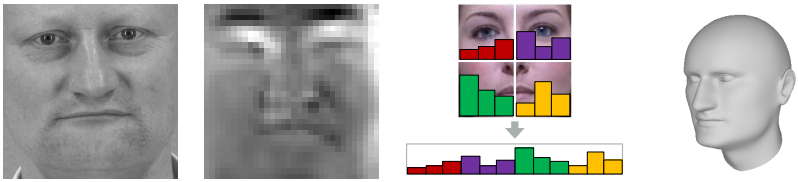
Considering the forensic analysis as information retrieval task, the respective performance measures can be used. As measure to rate the ranked result the average precision is used:

$$a = \sum_{k=1}^{B} p(k) \cdot \Delta r(k),$$

with the precision $p(k)$ at rank k and the difference for the recall $\Delta r(k)$ from rank $k \quad 1$ to k: $\Delta r(k) = r(k) \quad r(k \quad 1)$. Recall r and precision p result from the amount of true positives tp, false positives fp and false negatives fn up to rank k:

$$p(k) = \frac{tp(k)}{tp(k) + fp(k)},$$
$$r(k) = \frac{tp(k)}{tp(k) + fn(k)}.$$

It is $0 \leq a \leq 1$ for the average precision a. For $a = 1$ all relevant videos in the database, which show the wanted identity, are ranked at the topmost positions. The lower the relevant matches are ranked, the lower the average precision becomes. An important feature of the average precision is that it does not just represent the best match, but the whole ranking. Therefore, a relevant match at rank two in the list yields a better score than one at rank three. But both contribute to the score. This procedure fits our scenario of forensic analysis. Usually, the results will be inspected by humans at the end. In this case two aspects are relevant. First, it is not sufficient to sort only one correct match to the top of the list, but as many



(a) (b) (c) (d)

Figure 3.1: Different ways of representing a face: (a) intensity image, (b) in a subspace, (c) by local features, (d) 3D-model.

as possible. Secondly, it is not a severe problem if a few wrong videos appear between the correct ones.

Building the mean out of N queries to the database, results in the mean average precision map:

$$map = \frac{1}{N} \sum_{i=1}^{N} a_i.$$

3 Face Model

Face recognition for videos can be split into two steps: modeling of the face and modeling of the temporal sequence. First, in this section the face modeling will be examined (step 3 in Fig. 1.2). This means to model the single frames f_j in a video V. In the next section the modeling of the sequence V as a collection of frames is discussed (step 4 in Fig. 1.2).

While there exists a large variety of possibilities to describe objects in images, a clear amount has established itself in the field of face recognition. The initial step is a brief discussion of the established approaches. The main concepts to describe a face in an image [LJ11] are presented in the following list and in Fig. 3.1:

- *Intensity image* – The intensity image of the face taken by the camera is used as face descriptor. This was already denoted as f before (Fig. 3.1(a)).

- *Subspace methods* – The intensity face image is projected into a pre-trained face subspace. The well-known Eigenfaces [TP91] and Fisherfaces [BHK97] approaches work this way. They use a PCA or an LDA respectively for the projection (Fig. 3.1(b)).

- *Local features* – The face is divided into several local patches. For each patch, features like Local Binary Patterns or Gabor features are extracted [ZJN07]. The combination of the patch features yields the face model (Fig. 3.1(c)).

- *Model based* – The face is represented by a 3D-model. An individual face model can be generated out of a 2D-image [BV03] (Fig. 3.1(d)).

The resulting model for a frame f will be denoted as $\tilde{f}$. Approximately, the complexity of the approaches increases from the top to the bottom of the list. With increasing complexity the necessary processing time increases as well. The processing time ranges from practically none for the intensity image, because $\tilde{f} = f$, to several seconds for the generation of an individual 3D-model for a specific face.

4 Sequence Model

Modeling a sequence of face images allows the step from still image face recognition to face recognition in video. Obviously, a sequence $\tilde{V}$ of face models $\widetilde{\tilde{f}_j}$ contains more information than a single model, provided that the same image acquisition system is used. However, usually video data is of much worse quality than still image data. The loss of quality for video data mostly comes from lower resolution and a less constrained environment. Common techniques to create a sequence model $\overline{V}$ are:

- *Best shot* – The quality of each frame $\widetilde{f}_j$ in the sequence is rated with respect to the face recognition task. The frame which seems suitable best for the recognition is selected: $\overline{V} = \widetilde{f}_{best}$. This way, the task is reduced to still image face recognition.

- *Set of frames* – The frames of one video are interpreted as a set of vectors: $\overline{V} = \{\widetilde{f}_j | j = 1..F\}$. Thus, comparing two videos means to compare two sets of vectors. An analysis for the most basic similarity measures was performed in [CMH+11], showing that the Nearest Neighbor Distance seems to be the best.

- *Linear subspace* – All frames of a sequence together build a subspace in the image space. This subspace could be modeled, for example, by the affine or convex hull [CT10]. The Mutual Subspace Method (MSM) [YFM98, FY05] is the most basic one of the approaches. The similarity between subspaces in this case is measured by the principle angle between them.

- *Manifold* – Instead of assuming a linear subspace, the sequence is modeled as a nonlinear manifold. A big variety of manifold models and comparison approaches have been tested: e.g. LLE [HP09], Isomap [Yan02] or kernel based methods [CT10, SM11]. However, their high flexibility brings the risk of overfitting the data.

- *Probabilistic* – Two approaches fall in this category: distribution based and test based. In the first, a distribution of the frames in some space is determined and the similarity between videos is rated by standard distribution distances [ZC06]. The second possibility consists of drawing sample frames from the videos to test the identity hypothesis [DLZ+13].

A short complexity analysis. Two steps need computation: model generation and model comparison. Model generation is the less important part as this needs to be done only once for a video database C. However, there is typically more than one search request to the database C. Thus, comparisons should have higher priority with respect to computation time. A simple way to estimate the cost for one comparison is the dimension D of the sequence model $\overline{V}$. Let $\widetilde{d}$ denote the dimension of one frame model $\widetilde{f}_j$. Then, the dimension D for the sequence model is usually the lowest for the best shot approach $D = \widetilde{d}$. The dimension D is the highest for the set of frames and the manifold approaches $D \geq F \cdot \widetilde{d}$, where at least all frames are part of the model. The dimension D of the other approaches is typically somewhere in between.

5 Possible improvements

Typically, set of frames based sequence modeling yields the best recognition results. But it is quite slow. Two possibilities are presented to reduce the dimension of the sequence model for set of frames based approaches. The first one is to perform a vector quantization. Practically this is done by understanding the sequence $\widetilde{V}$ as a set and clustering it. For each cluster, one representative vector is kept. However, this method looses information by omitting data from further processing. For this reason, the second approach is a content based reduction of the sequence model dimension D [Her13]. Similar frames are found based on the head pose and a fused representation of them is kept in the sequence model.

Another improvement to reduce the computation time is inspired by the most well-known application for a cascade, the Viola-Jones object detector [VJ01]. The approaches are comined in a cascade. Starting with the fastest method of sufficient

performance in the first stage of the cascade and ending with the slowest and best-performing method n the last stage. Each stage in the cascade can either eliminate complete videos or some frames in each video. The remaining data is processed by the next stage. Formally speaking, let C^0 denote the initial database of videos V_b^0. A stage s with input

$$C^{s-1} = \{V_1^{s-1}, \dots, V_{B^{s-1}}^{s-1}\}$$

and

$$V_b^{s-1} = (f_1^{s-1}, \dots, f_{F_b^{s-1}}^{s-1})$$

has two processing options. The first is to reduce the number of videos, leading to the output

$$C^s = \{V_i^{s-1} \mid i \in N_v, N_v \subset \{1, \dots, B^{s-1}\}\}.$$

The second possibility is to identify and remove irrelevant frames from a sequence V_b^{s-1}. Thus, the output is

$$V_b^s = (f_i^{s-1} \mid i \in N_f, N_f \subset \{1, \dots, F_b^{s-1}\}).$$

Of course, a stage s can combine both processing options. Keeping track of the removed videos in each stage allows to create a full ranked list of the videos in the database with respect to the query. The difficulty in building a good cascade is to choose the right number of stages with their corresponding parameters. One possibility is to manually define performance requirements for each stage and then search for the approach that best fulfills them. This is mentioned in the original Viola-Jones detector design. There are attempts to automatize the design of a cascade for the binary classification case [SRB04]. However, they can not be transfered in a simple way to the information retrieval case and it is unclear if this is possible at all. This leaves the manual design as the only design option at the moment.

6 Evaluation

For evaluation, the combined Honda/UCSD dataset [LHYK03, LHYK05] is used. Face images are downscaled to 32 $\times$ 32 pixels. The dataset contains 92 videos of 35 persons. The evaluation was done using the leave-one-out strategy. This means to use one video as query and the remaining 91 as database. The mean average precision map is based on all 92 possible queries. The measured query time t consists of the actual time necessary for the database search t_s and the necessary

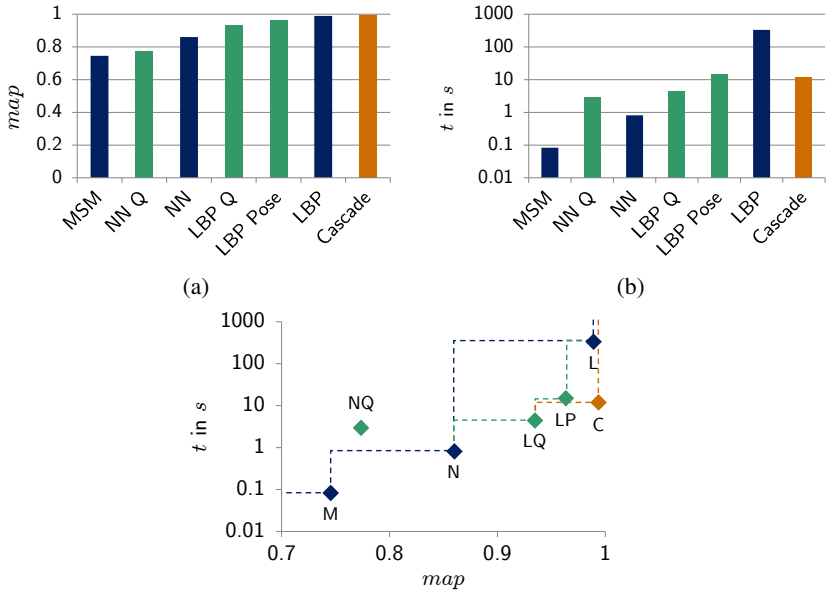


Figure 6.1: Comparison of different approaches. Basic approaches are blue, the ones using dimension reduction are green and the cascade approach is brown. The 'Q' denotes simple vector quantization and 'Pose' the content based dimension reduction. (a) mean average precision map, (b) average query time t and (c) comparison of map and t. Pay attention to the logarithmic scale of the time axis.

time to prepare the query video t_p: $t = t_p + t_s$. It contains the whole time which is needed for one search in the database. In real world scenarios, the query video is usually not in the database and therefore not preprocessed. Thus, the time t_p to build the sequence model for the query video needs to be included.

Fig. 6.1 shows the measured results. As basic approaches, MSM with intensity images (MSM), nearest neighbor with intensity images (NN) and nearest neighbor with local binary patterns (LBP) were chosen. The three methods show the expected behavior: MSM being the fastest, but worst, LBP being the slowest, but best and NN in the middle. As can be seen in Fig. 6.1(c) all three have the right to exist because higher computation time correlates with higher recognition performance. Which one should be used depends on the processing time limits. Better approaches compared to the basic ones, would be below the dashed line, worse ones above. The better an approach is, reaching a high map in a small time t, the

more to the lower right corner of the diagram it would be located.

Improving the set of frames based nearest neighbor method by quantization makes LBP faster (LBP Q), but not NN (NN Q). This is because the quantization time t_p of the query video is higher than the whole query time t for the pure NN. So NN Q is a useless approach. However, LBP Q is located between NN and LBP, both in terms of map and t. The head pose based dimension reduction of LBP (LBP Pose) yields better search results than LBP Q, but needs a little more processing time. At the end, it is located between LBP Q and pure LBP.

Finally, a cascade of the three basic approaches is considered. It uses MSM in the first, NN in the second and LBP in the last stage. The optimization of the cascade results in the following process: MSM sorts out about 30 percent of the videos, NN sorts out about 90 percent of the frames in each of the remaining videos and LBP is performed on the rest. This means that the LBP stage only has to process about 7 percent of the original data. The results show that the cascade approach renders the LBP Pose and the pure LBP approach useless as it is faster and yields a better map than both.

It should be noted, that all presented methods allow querying faster than real time. Each video in the dataset lasts about 10 seconds, making a total playtime of about 900 seconds. Even the slowest approach needs less than 400 seconds for one query.

7 Conclusion

A thorough analysis of basic face recognition techniques was given with respect to the scenario of forensic analysis. The mutual subspace method proved to be the fastest basic solution showing an acceptable performance. The best basic solution with respect to recognition performance uses the Local Binary Patterns. Several improvements to reduce the processing time were presented and evaluated. The most promising solution seems to be a cascade of basic face recognition techniques. The manual design of the cascade might be a drawback but also allows for situation specific adaptation. Altogether, the cascade achieved the highest recognition performance on the evaluated dataset while needing less computation time than most of the other approaches.

Bibliography

[BHK97] P.N. Belhumeur, J.P. Hespanha, and D.J. Kriegman. Eigenfaces vs. fisherfaces: Recognition using class specific linear projection.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 19(7):711–720, 1997.

[BV03] V. Blanz and T. Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1063–1074, 2003.

[CMH+11] Shaokang Chen, Sandra Mau, Mehrtash T. Harandi, Conrad Sanderson, Abbas Bigdeli, and Brian C. Lovell. Face recognition from still images to video sequences: A local-feature-based framework. *EURASIP Journal on Image and Video Processing*, 2011, 2011.

[CT10] H. Cevikalp and B. Triggs. Face recognition based on image sets. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 2567–2573. IEEE, 2010.

[DLZ+13] Sihao Ding, Ying Li, Junda Zhu, Yuan F Zheng, and Dong Xuan. Robust video-based face recognition by sequential sample consensus. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 336–341. IEEE, 2013.

[FY05] K. Fukui and O. Yamaguchi. Face recognition using multi-viewpoint patterns for robot vision. *Robotics Research*, pages 192–201, 2005.

[Her13] Christian Herrmann. Extending a local matching face recognition approach to low-resolution video. In *Advanced Video and Signal Based Surveillance (AVSS), 2013 10th IEEE International Conference on*, pages 460–465, 2013.

[HP09] A. Hadid and M. Pietikäinen. Manifold learning for video-to-video face recognition. *Biometric ID Management and Multimodal Communication*, pages 9–16, 2009.

[LHYK03] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman. Video-based face recognition using probabilistic appearance manifolds. *IEEE Conf. On Computer Vision and Pattern Recognition*, 1:313–320, 2003.

[LHYK05] K.C. Lee, J. Ho, M.H. Yang, and D. Kriegman. Visual tracking and recognition using probabilistic appearance manifolds. *Computer Vision and Image Understanding*, 99(3):303–331, 2005.

[LJ11] Stan Z. Li and Anil K. Jain. Introduction. *Handbook of Face Recognition*, pages 1–18, 2011.

[PGM11] P. Jonathon Phillips, Patrick Grother, and Ross Micheals. Evaluation methods in face recognition. *Handbook of Face Recognition*, pages 551–574, 2011.

[SM11] G. Shakhnarovich and B. Moghaddam. Face recognition in subspaces. *Handbook of Face Recognition*, pages 19–49, 2011.

[SRB04] Jie Sun, James M Rehg, and Aaron Bobick. Automatic cascade training with perturbation bias. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–276. IEEE, 2004.

[TP91] M. Turk and A. Pentland. Eigenfaces for recognition. *Journal of cognitive neuroscience*, 3(1):71–86, 1991.

[VJ01] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2001. CVPR 2001*, volume 1, pages I–511. IEEE, 2001.

[Yan02] M.H. Yang. Face recognition using extended isomap. In *International Conference on Image Processing. 2002*, volume 2, pages II–117. IEEE, 2002.

[YFM98] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *Third IEEE International Conference on Automatic Face and Gesture Recognition, 1998*, pages 318–323. IEEE, 1998.

[ZC06] Shaohua Kevin Zhou and Rama Chellappa. From sample similarity to ensemble similarity: Probabilistic distance measures in reproducing kernel hilbert space. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):917–929, 2006.

[ZJN07] Jie Zou, Qiang Ji, and George Nagy. A comparative study of local matching approach for face recognition. *IEEE Transactions on Image Processing*, 16(10):2617–2628, 2007.