Hybrid Face Recognition Based on Real-Time Multi-camera Stereo-Matching

J. Hensler, K. Denker, M. Franz, and G. Umlauf

University of Applied Sciences Constance, Germany

Abstract. Multi-camera systems and GPU-based stereo-matching methods allow for a real-time 3d reconstruction of faces. We use the data generated by such a 3d reconstruction for a hybrid face recognition system based on color, accuracy, and depth information. This system is structured in two subsequent phases: geometry-based data preparation and face recognition using wavelets and the AdaBoost algorithm. It requires only one reference image per person. On a data base of 500 recordings, our system achieved detection rates ranging from 95% to 97% with a false detection rate of 2% to 3%. The computation of the whole process takes around 1.1 seconds.

1 Introduction

In the last years, 3d face recognition has become an important tool in many biometric applications. These systems are able to achieve high detection rates. However, there is one major drawback: the overall recognition process, including 3d reconstruction and face recognition, takes several seconds to several minutes. This time is unacceptable for biometric systems, e.g. security systems, credit card verification, access control or criminal detection.

In order to speed up this process, a multi-camera stereo-matching system has been developed that can generate a high-resolution depth image in real-time [1]. Here, we use such a system (shown in Figure 1) for face recognition. A typical recording of this system is shown in Figure 2. Since most computations are done on the GPU, the system needs an average computation time of 263 milliseconds for one high resolution depth image (see [1]). In this paper, we show that the quality of these depth images is sufficiently high for 3d face recognition in the context of an access control system. An access control system requires a high detection rate at a low computation time. Hence, the recognition algorithm combines three different types of information obtained from the multi-camera stereo-matching system: a depth image (Figure 2(b)), a color image (Figure 2(a)), and a 3d reconstruction quality image (Figure 2(c)).

Our 3d face recognition algorithm is structured in two subsequent phases (Figure 3): the data preparation phase (Section 3) and the face recognition phase (Section 4). In the data preparation phase the face data is segmented from the background in the color and depth images. Then, the 3d face data is transformed into frontal position by an optimized iterative closest point (ICP) algorithm.

G. Bebis et al. (Eds.): ISVC 2011, Part II, LNCS 6939, pp. 158–167, 2011.

Fig. 1. The multi-camera stereo-matching system used in this paper generates one depth image from four camera images

(a) Color image (b) Depth image (c) Quality image

Fig. 2. A typical recording of the multi-camera stereo-matching system. Bright pixels in the quality image depict regions with poor variation in the depth image.

Regions with poor quality are improved by a hole-filling algorithm. The face recognition phase uses an AdaBoost classifier based on histogram features that describe the distribution of the wavelet coefficients of the color and depth images.

2 Related Work

Similar to 2d face recognition, 3d face recognition methods can be divided into global and local approaches. Global methods recognize the whole face at once while local approaches separate features of the face and recognize these features independently.

A global approach is used in [2]. After a data preparation using symmetry- and nose-tip detection, an eigenface based recognition is computed on the normalized depth images. For eigenfaces [3] a principal component analysis (PCA) is applied to the images from a face data base to compute basis-images. These basis-images are linearly combined to generate synthetic face images.

Morphable models are parametric face models yielding a realistic impression used for 3D face synthesis [4]. In [5] these models are used for face recognition. The morphable model is fitted to a photograph and a distance of the model parameters is used for recognition. Fitting the morphable model takes several

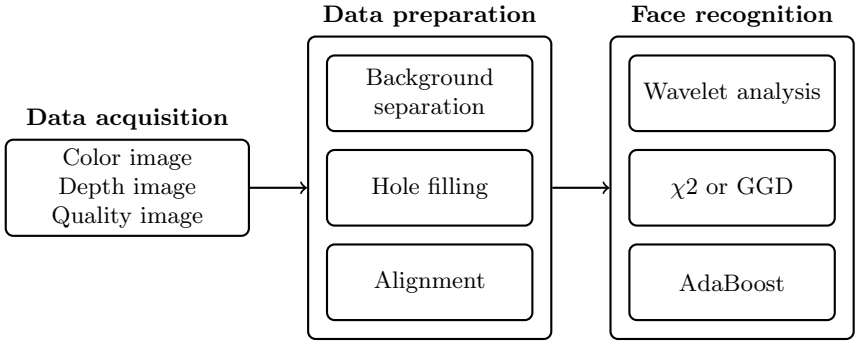


Fig. 3. The structure of the 3d face recognition system

minutes. A fast modification of this method is presented in [6]. Only for the training faces a morphable model is computed. For the recognition a support vector machine (SVM) is used to compare synthetic images of face components from the morphable model with face components extracted from photographs.

SVM based face recognition methods, as [6–8], need a large training data base. The SVM is trained using several hundred positive and negative example data sets. To speed the training of the SVM up, the data is reduced to a set of facial features. Because of this reduction, these methods are local.

An ICP algorithm similar to our data preparation phase is used in [9]. After a pre-matching using facial features, ICP is used to get a precise fit of the test data to a reference face. Differences of surface points on both data sets are used for recognition. Here, a PCA is used to reduce the dimension of the search space, where a Gaussian mixture model is used for the final recognition.

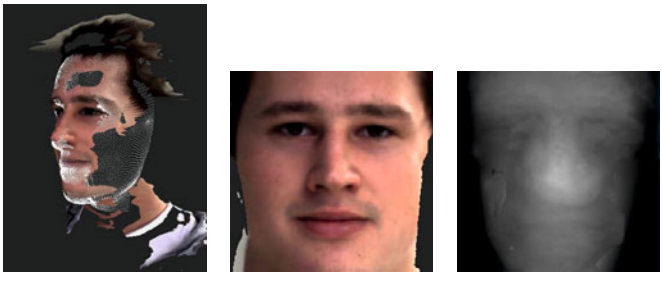
3 Data Preparation

The data preparation phase gets as input the color, depth, and quality images as computed by a system like the one presented in [1].

For face recognition it is necessary to separate the regions in the images that contain information of the face from irrelevant, background regions. In an access control system, we assume that the face is the object closest to the camera. Thus, the points of the face are identified in the depth image to separate the face from the background in the color and quality images.

The quality image contains information about the faithfulness of the 3d reconstruction. Low quality values characterize regions with a large instability in the depth image. Thus, these regions are removed from the 3d face model, leaving holes. These holes are filled with a moving least squares approach fitting a polynomial surface of up to degree four to the points around the hole [10].

Although, after the hole filling the depth image contains a complete 3d model of the face, its affine position relative to the camera is unknown. To align the



(a) (b) (c)

Fig. 4. (a) ICP fit of a 3d mannequin head model (white points) to an incomplete 3d model, (b) aligned color image, and (c) depth image after the hole-filling

3d face model we fit it to a mannequin head model in frontal position using an iterative closest point (ICP) algorithm [11]. For each point on both models the nearest point on the other model is computed. Then, a global affine transformation minimizing the distance of these point-pairs is computed. This affine transformation is applied to the 3d face model and the procedure is repeated until the changes become small enough. For the 3d models in our application with more than 200,000 data points the ICP algorithm is speed up as in [12]:

- Point-pairs are computed only for a random subset of points.
- To compute the point-pairs a kd-tree is used.
- Outliers are rejected by a point-to-point distance threshold.
- For the first few iterations point-to-point distances are used. Later the algorithm uses point-to-plane distances.

A resulting 3d model after ICP alignment is shown in Figure 4(a) for an 3d model without hole filling. The white points show the mannequin model.

After the alignment also the color and the depth image are aligned with the computed affine transformation, see Figures 4(b) and 4(c). Further results of the complete data preparation phase for three depth images of the same person are shown in Figure 5. These images show that the data preparation is robust against different positions of the person to the camera, different rotations of the head, and different facial expressions.

4 Face Recognition

The face recognition phase is based on the aligned and completed depth and color images. First, a 2d wavelet transform is applied to both the depth and the color image. This transform generates a series of smaller images, called subbands, using a bank of low- and high-pass filters. Depending on the choice of the filters, one obtains different types of wavelets. We tested eight wavelets: Quadratic mirror filter (QMF) wavelets of size 5, 9 and 13, Daubechies wavelets of size 2, 3 and 4, and bi-orthogonal CDF wavelets of size 5/3 and 9/7. The

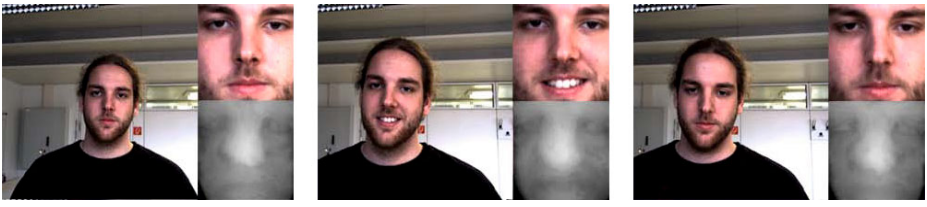


Fig. 5. Result of the data preparation phase: Three different depth images of the same person aligned to a frontal position (aligned color/depth image in resp. right column)

structure of the wavelet-transformed images is shown in Figure 6 where L and H refer to low-pass or high-pass filtering in either horizontal or vertical direction. The number refers to the level (octave) of the filtering. At each level, the low pass sub-band (LL) is recursively filtered using the same scheme.

The low frequency sub-band LL contains most of the energy of the original image and represents a down-sampled low resolution version. The higher frequency sub-bands contain detail information of the image in horizontal (LH), vertical (HL) and diagonal (HH) directions. The distribution of the wavelet coefficient magnitudes in each sub-band are characterized by a histogram. Thus, the entire recording is represented by a feature vector that consists of the histograms of all sub-bands of the depth and the color image. Note that the wavelet coefficients of each sub-band are uncorrelated. Hence, it makes sense to train individual classifiers for each sub-band (referred to as *weak classifiers*) which are subsequently combined into a *strong classifier* by the AdaBoost algorithm. Our weak classifiers are simple thresholds on a similarity metric between sub-band histograms. We tested two types of similarity metrics: (1) the $\chi 2$-metric for histograms, and (2) the Kullback-Leibler (KL) divergence of of a generalized Gaussian density (GGD) functions fitted to the histogram.

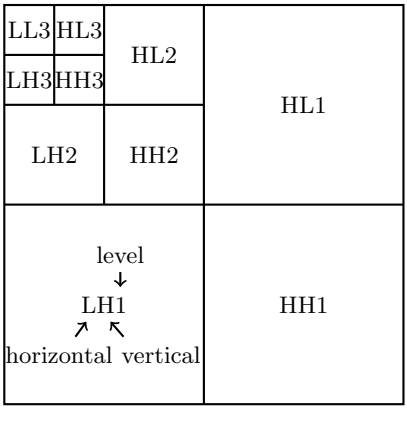


Fig. 6. The sub-band labeling scheme for a three level 2D wavelet transformation

4.1 χ^2-Metric

The distribution of the wavelet coefficients of each sub-band is represented in a histogram. In order to find the optimal bin size for the histograms we used the method of [13] according to which the optimal bin size h is given by

$$h = 2(Q_{0.75} - Q_{0.25})/\sqrt[3]{n} \tag{1}$$

where $Q_{0.25}$ and $Q_{0.75}$ are the 1/4- and 3/4-quantiles and n is the number of recordings in the training data base. The χ^2-metric computes the distance d between two sub-band histograms H_1 and H_2 with N bins as

$$d(H_1, H_2) = \sum_{i=1}^{N} \frac{(H_1(i) - H_2(i))^2}{H_1(i) + H_2(i)}. \tag{2}$$

4.2 KL Divergence between Generalized Gaussian Density Functions

As an alternative to the χ^2-metric, we tested a generalized Gaussian density (GGD) based method [14]. This method defines an individual GGD function that is fitted to the coefficient distribution of a sub-band of the wavelet transform. The optimal fit is obtained from maximizing the likelihood using the Newton-Raphson method [14–16]. The distance between two GGD functions is estimated by the Kullback-Leibler divergence [17].

4.3 The AdaBoost Algorithm

The concept of boosting algorithms is to combine multiple weak classifiers to yield a strong classifier that solves the decision problem. The idea is that it is often easier to find several simple rules for a decision instead of one complex rule. The AdaBoost algorithm uses a training data set to build a strong classifier out of weak classifiers that solve binary decisions. For this purpose, the algorithm needs weak classifiers with a success rate of at least 50% on the training data with independent errors. Then, the AdaBoost algorithm can be shown to improve the error rate by computing an optimal weight for each weak classifier.

Let $y_i = h_i(x)$ denote the output of the i-th of the M weak classifiers to the input x, and α_i the weight of $h_i(x)$ generated by the AdaBoost algorithm. Then, the strong classifier is given by [18]

$$H(x) = \text{sign}\left(\sum_{i=1}^{M} \alpha_i h_i(x)\right). \tag{3}$$

5 Results

For training and testing we collected a data base of approximately 500 depth images from 40 different persons. For some persons the images were taken at different times, with different lighting, different positions with respect to the camera

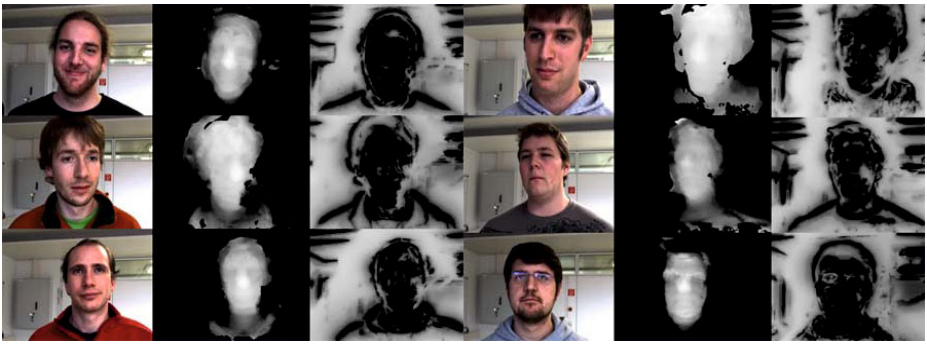


Fig. 7. Example images from our data base used for training and testing of the AdaBoost algorithm

system, different facial expressions (open/closed mouth, smiling/not smiling, open/closed eyes) and different facial details (glasses/no glasses). Some example images are shown in Figure 7.

The results of our recognition system are shown in the receiver operating characteristic (ROC) diagrams in Figure 9 and Table 1. The system was tested with different wavelet transform levels and different wavelet filters. Note that, if the weak classifier are too strong or too complex, boosting might fail to improve the recognition, cf. [19]. An indicator for this behavior is a quick decrease of the error rate in the training phase. The error rate in the training phase compared to the number of weak classifiers is illustrated in Figure 8. Here, in the first wavelet level the error rate starts very low and strong classifiers improve relatively slow. At wavelet level three the error rate starts higher and the boosting finds more weak classifiers to improve the error rate more effectively. Hence, a more robust and more reliable result is achieved in the third level of the wavelet decomposition.

Table 1. Results with our approach after 3-fold cross validation with different wavelet transformation levels and wavelet filters

filter	**level=1**	**level=2**	**level=3**
qmf5	0,9831	0,9898	0,9898
qmf9	0,9848	0,9897	0,9884
qmf13	0,9817	0,9890	0,9895
daub2	0,9798	0,9877	0,9892
daub3	0,9843	0,9859	0,9898
daub4	0,9877	0,9873	0,9891
cdf53	0,9847	0,9893	0,9914
cdf97	0,9836	0,9900	0,9912
Mean	0,9837	0,9886	0,9898
Std	0,0023	0,0015	0,0010

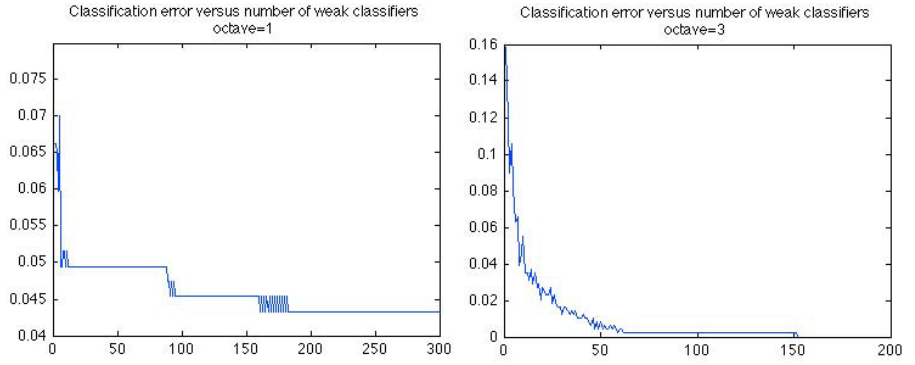


Fig. 8. Classification error versus number of weak classifiers at level one and three of the wavelet decomposition

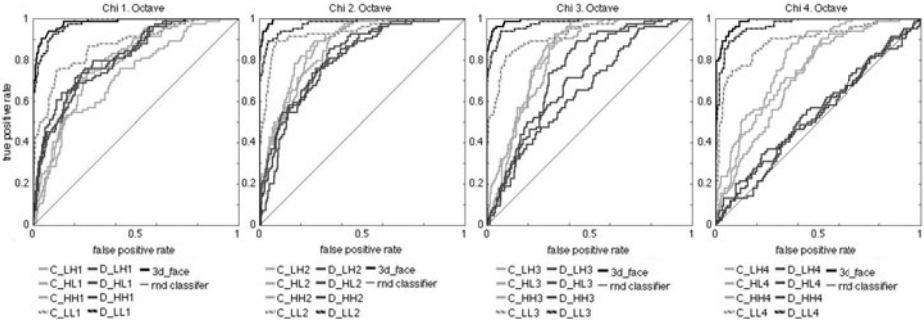


Fig. 9. ROC curves for different wavelet transformation levels. At each level the four sub-bands `LH`, `HL`, `HH`, and `LL` for the depth (`D_`) and color (`C_`) images and their combination with AdaBoost (`3d_face`) are shown.

Table 1 shows that the choice of the used wavelet filter does influence the result clearly. The best result is achieved with the cdf53/cdf97 filter and wavelet transformation at level three. χ^2-histogram-comparison and GGD fitting yield similar results. Since the former is computationally more efficient we use this metric in the current version of our system for faster response times.

The recognition results are shown in Figure 9. The detection rates between 95% and 97% for the low false positive rate of 2% to 3% are obtained at the point of the minimal overall error of the ROC curve. The AdaBoost combination (`3d_face`) of all sub-bands yields the best decision at levels two and three. At wavelet level four, the sub-bands are getting too small and the final AdaBoost classificator is not effective.

For the presented results, we use the FireWire camera system from [1]. Color images and depth maps from this system have a resolution of 1392×1032 pixels. Currently the overall recognition time is 1.086 seconds with the χ^2-metric. This includes the 3d reconstruction (263 ms [1]), the data preparation (731 ms), and

the face recognition (χ^2 method - level 3 - 92ms). The most time is consumed by the data preparation which takes approximately 65% of the overall time. We are working here on further improvements on the ICP algorithm, e.g. finding a better initial guess.

6 Conclusion and Future Work

Our analysis shows that the proposed system has a satisfying face recognition performance which is competitive to other systems, cf. [20]. A special advantage of our system is that it requires only one single reference depth image per person. Other systems often need more than one reference image without obtaining better ROC curves than ours, e.g. [7, 8]. Since the quality of the 3d model, colors, and shadows in the 2D images critically depend on the lighting of the faces, we expect that the performance of the current system can be significantly improved by controlling the lighting conditions.

All computations take about one second which is acceptable for a biometric system. This computation time also allows for taking several subsequent images to improve the detection rate. However, we are still working on various optimizations, especially for the data preparation phase that will further reduce processing time.

Acknowledgements. This work was supported by AiF ZIM Project KF 2372101SS9. We thank the students and employees of the HTWG Konstanz for providing the data in our face data base.

References

1. Denker, K., Umlauf, G.: Accurate real-time multi-camera stereo-matching on the gpu for 3d reconstruction. Journal of WSCG 19, 9–16 (2011)
2. Pan, G., Han, S., Wu, Z., Wang, Y.: 3D face recognition using mapped depth images. In: IEEE Conf. on Computer Vision and Pattern Recognition, pp. 175–181 (2005)
3. Turk, M., Pentland, A.: Eigenfaces for recognition. Cognitive Neuroscience 3, 71–86 (1991)
4. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3d faces. In: SIGGRAPH 1999, pp. 187–194 (1999)
5. Blanz, V., Romdhani, S.: Face identification across different poses and illuminations with a 3d morphable model. In: Int'l. Conf. on Automatic Face and Gesture Recognition, pp. 202–2007 (2002)
6. Weyrauch, B., Huang, J., Heisele, B., Blanz, V.: Component-based face recognition with 3d morphable models. In: Workshop on Face Processing in Video, pp. 1–5 (2003)
7. Lee, Y., Song, H., Yang, U., Shin, H., Sohn, K.: Local feature based 3D face recognition. In: Kanade, T., Jain, A., Ratha, N.K. (eds.) AVBPA 2005. LNCS, vol. 3546, pp. 909–918. Springer, Heidelberg (2005)
8. Lee, J., Kuo, C., Hus, C.: 3d face recognition system based on feature analysis and support vector machine. In: IEEE TENCON 2004, pp. 144–147 (2004)

9. Cook, J., Ch, V., Sridharan, S., Fookes, C.: Face recognition from 3d data using iterative closest point algorithm and Gaussian mixture models. In: 2nd Int'l. Symp. 3D Data Processing, Visualization, and Transmission, pp. 502–509 (2004)
10. Wang, J., Oliveira, M.: A hole-filling strategy for reconstruction of smooth surfaces in range images. In: SIBGRAPI 2003, pp. 11–18 (2003)
11. Besl, P., McKay, N.: A method for registration of 3-D shapes. IEEE Trans. on Pattern Analysis and Machine Intelligence, 239–256 (1992)
12. Rusinkiewicz, S., Levoy, M.: Efficient variants of the ICP algorithm. In: 3dim, p. 145. IEEE Computer Society, Los Alamitos (2001)
13. Freedman, D., Diaconis, P.: On the histogram as a density estimator: L2 theory. Probability Theory and Related Fields 57, 453–476 (1981)
14. Lamard, M., Cazuguel, G., Quellec, G., Bekri, L., Roux, C., Cochener, B.: Content based image retrieval based on wavelet transform coefficients distribution. In: 29th IEEE Conf. of the Engineering in Medicine and Biology Society, pp. 4532–4535 (2007)
15. Varanasi, M., Aazhang, B.: Parametric generalized Gaussian density estimation. J. of the Acoustical Society of America 86, 1404 (1989)
16. Do, M., Vetterli, M.: Wavelet-based texture retrieval using generalized Gaussian density and Kullback-Leibler distance. IEEE Trans. on Image Processing 11, 146–158 (2002)
17. Kullback, S., Leibler, R.: On information and sufficiency. Ann. Math. Stat. 22, 79–86 (1951)
18. Hensler, J., Blaich, M., Bittel, O.: Improved door detection fusing camera and laser rangefinder data with AdaBoosting. In: 3rd Int.'l Conf. on Agents and Artificial Intelligence, pp. 39–48 (2011)
19. Schapire, R.: A brief introduction to boosting. In: International Joint Conference on Artificial Intelligence, vol. 16, pp. 1401–1406 (1999)
20. Bowyer, K., Chang, K., Flynn, P.: A survey of approaches and challenges in 3d and multi-modal 3d+2d face recognition. Computer Vision and Image Understanding 101, 1–15 (2006)