FAST FACIAL FEATURE EXTRACTION USING A DEFORMABLE SHAPE MODEL WITH HAAR-WAVELET BASED LOCAL TEXTURE ATTRIBUTES

Fei Zuo

Eindhoven Univ. of Technol., Depart. EE
5600MB Eindhoven, The Netherlands

Peter H. N. de With

LogicaCMG / Eindhoven Univ. of Technol.
P.O.Box 7089, 5605JB Eindhoven, NL

ABSTRACT

We propose a fast and improved facial feature extraction technique for embedded face-recognition applications. This technique applies to both face alignment and recognition and significantly improves three aspects. First, we introduce local texture attributes to a statistical face model. A texture attribute characterizes the 2-D local feature structures and is used to guide the model deformation. This provides more robustness and faster convergence than with conventional ASM (Active Shape Model). Second, the local texture attributes are modelled by Haar-wavelets, yielding faster processing and more robustness with respect to low-quality images. Third, we use a gradient-based method for model initialization, which improves the convergence. We have obtained good results dealing with test faces that are quite dissimilar with the faces used for statistical training. The convergence area of our proposed method almost quadruples compared to ASM. The Haar-wavelet transform successfully compensates for the additional cost of using 2-D texture features. The algorithm has also been tested in practice with a webcam, giving (near) real-time performance and good extraction results.

1. INTRODUCTION

Accurate facial feature extraction is important for face alignment, which is an indispensable processing step between face detection and recognition. Our aim is to build a feature-extraction system that can be used for face recognition in embedded and/or consumer applications. This imposes specific requirements to the algorithm in addition to extraction accuracy, such as real-time performance under varying imaging conditions and robustness with low-cost imaging hardware.

In an earlier research, *Yuille et al.* [1] use parameterized deformable templates to extract eyes and mouth. However, it is computationally expensive and the convergence is not guaranteed. We presented earlier [2] a faster facial feature localization technique, but it only provides partial feature descriptions (e.g. iris center). Recently, the Active Shape Model (ASM) [3] and Active Appearance Model (AAM) [3] were proposed as two effective techniques for feature extraction. The ASM fits a shape model to the real image by using a local deformation process constrained by a global variance model. This approach only uses 1-D profile information during the model deformation, which has a relatively small convergence area. The AAM incorporates global facial texture modelling giving more matching robustness, but it is slower than ASM and more sensitive to texture variations under different illumination conditions. In addition, both ASM and AAM suffer from wrong convergence when the model is initialized far away from the real face position.

This paper attempts to solve the above problems. The key to our solution is based on three improvements. Firstly, we use a gradient map for fast determination of the approximate facial feature positions. Secondly, we employ 2-D texture attributes for improving the convergence and robustness of a deformable face model. Thirdly, we adopt Haar-wavelets for modelling local texture attributes, which offers high processing speed and robustness for low-quality images.

2. FEATURE MODEL WITH SHAPE DESCRIPTION AND 2-D LOCAL TEXTURE ATTRIBUTES

2.1. Initialization: feature position estimation

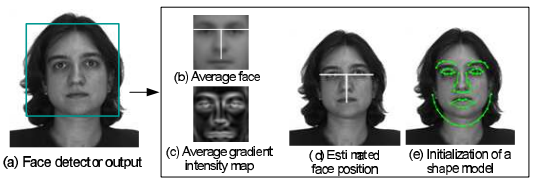

Fig. 1. Estimation of facial feature positions.

Existing techniques like ASM do not have a good initialization stage. In our case, we use a fast estimation of the face position in order to improve the overall algorithm's correctness and convergence. Similar to [4], we employ a gra-

0-7803-8554-3/04/$20.00 ©2004 IEEE.

dient map of the image to determine the best match between a candidate region and an average face template. Once the face position and scale are estimated, our face model (see next section) is initialized accordingly (Fig. 1).

2.2. Face model

We define a face model FM as an ordered set of N_{FP} feature points, $\{FP_i|1 \leq i \leq N_{FP}\}$. Each FP_i is characterized by a pair of vectors, thus $FP_i = \langle \overrightarrow{P_i}, \overrightarrow{T_i} \rangle$. In this formula, $\overrightarrow{P_i}$ gives the position (x_i, y_i) of FP_i, and $\overrightarrow{T_i}$ represents the texture attribute vector associated with FP_i. The shape description of the face is formed by the set of positions $P = \{\overrightarrow{P_i}|1 \leq i \leq N_{FP}\}$. In contrast with the shape model used by ASM, we assign to each feature point an additional texture attribute $\overrightarrow{T_i}$ to describe local feature structures. Fig. 2 illustrates our face model. The shape description P portrays the global topology of the facial features, while texture parameters $T = \{T_i|1 \leq i \leq N_{FP}\}$ describe the local patterns characterizing each feature point. Later we will see that the local texture attributes guide the deformation of the face model to a specific shape instance, while the global feature topology constrains the deformation to maintain a face-like shape.

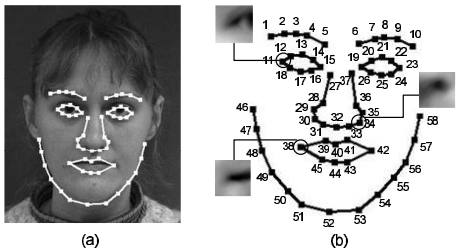

Fig. 2. Face model. (a) Face shape overlayed on a real image. (b) Topological shape with local texture attributes.

2.3. Modelling the local texture structure

We employ 2-D local textures because they contain richer and more reliable local pattern information than 1-D profiles normal to the contour. The latter is widely used in gradient-based deformable models, such as ASM. In our case, we model the 2-D local texture attribute T_i by extracting an $N \times N$ block around each feature point from the image. Subsequently, the local facial feature structures are transformed using Haar-wavelets for robustness and high processing speed. A closer examination of the local feature patterns in face images shows that they usually contain relatively simple patterns having strong contrast. The 2-D basis images of Haar-wavelets match very well with these patterns, so that it is attractive to exploit them for efficient signal representation. Furthermore, the simplicity of Haar wavelets supports the requirement of real-time implementation.

2.3.1. *Illumination normalization*

The local appearance of the feature is usually uniformly affected by illumination. For each feature block B with pixels $P(x, y)$, we reduce the illumination interference by normalization based on its mean μ_B and variance σ_B, hence, $P_N(x, y) = (P(x, y) - \mu_B)/\sigma_B$. The correction has been proven quite effective in our experiments. In the sequel, we will efficiently combine the above normalization with the implementation of the Haar-wavelet transform.

2.3.2. *Fast computation of Haar-wavelet features*

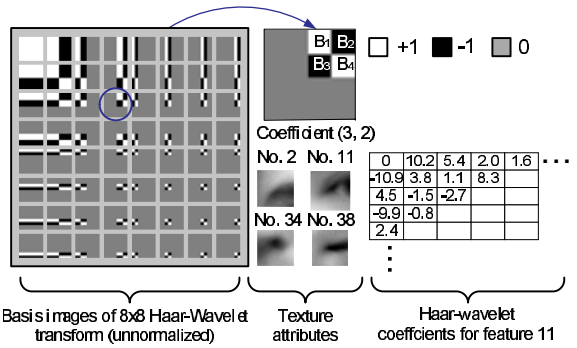

Fig. 3. Haar feature modelling.

Haar-wavelet decomposition mainly involves summations of pixel sub-blocks (see the Haar basis images shown in Fig. 3), which can be efficiently computed by using two auxiliary 'integral images' [5].

The integral image I of image with pixels P is defined as

$$I_P(u, v) = \sum_{x=1}^{u} \sum_{y=1}^{v} P(x, y). \tag{1}$$

For a block B with its top-left corner (x_1, y_1) and bottom-right corner (x_2, y_2), the summation of the pixel intensity in this block can be computed as: $S(B) = I(x_1, y_1) + I(x_2, y_2) - I(x_2, y_1) - I(x_1, y_2)$. A fast algorithm [5] can be applied to obtain the integral image of a given image in only one pass over the image. Similarly, a 'squared' integral image I_q can be obtained by replacing $P(x, y)$ in Equation (1) by $P^2(x, y)$, which facilitates fast computation of the block variance σ_B.

For a given pixel feature block B, the corresponding Haar-wavelet coefficient $H(u, v)$ can be computed by

$$H(u, v) = \frac{\sum_{i=1}^{N_B} \{Sgn(B_i) \sum_{x=1}^{M_{B_i}} \sum_{y=1}^{M_{B_i}} [B_i(x,y) - \mu_B]\}}{N(u,v) \cdot \sigma_B}$$

$$= \frac{1}{N(u,v) \cdot \sigma_B} \sum_{i=1}^{N_B} [Sgn(B_i) \cdot S(B_i)]. \tag{2}$$

Note that Equation (2) already incorporates the illumination correction from Section 2.3.1. In the above, B_i refers

to sub-blocks corresponding to non-zero coefficient areas in the basis images (see Fig. 3). The number of these sub-blocks is denoted as N_B. $Sgn(B_i)$ refers to the sign of the coefficient part corresponding to sub-block B_i, while the size of sub-block B_i is $M_{B_i} \times M_{B_i}$. Since coefficient $H(0,0)$ only contains the average intensity value of the block, it is zero for all illumination-corrected block images and can be ignored during the matching. For all remaining basis images, the total area of +1 signed sub-blocks is equal to the area of -1 signed sub-blocks. $N(u,v)$ is the normalization factor for coefficient $H(u,v)$. The summation of sub-block data can be efficiently computed using the integral image, so that the term $S(B_i)$ occurs in the second expression of Equation (2). Since $N_B \leq 4$ for $2^n \times 2^n$ blocks, the computation of one Haar coefficient involves at most 9 table lookups of the integral images (see Equation (1)).

2.3.3. Local texture attribute modelled by Haar features

From the above, each local feature attribute vector $\overrightarrow{T_i}$ can be represented by a transformed data vector

$$\overrightarrow{T_i} = (H_2(F_i), H_3(F_i), ..., H_C(F_i))^T, \quad (3)$$

where $H_k(F_i)$ gives the k-th (in zigzag order) Haar coefficient of the $N \times N$ block F_i around FP_i, and $C = N \times N$. Fig. 4 portrays that the Haar-wavelet coefficients represent a highly compact description of the local feature structures. For most feature points, less than 4% of the total coefficients are required to retain up to 95% of the feature signal energy. Thus, in Equation (3), the number of selected coefficients $C \ll N \times N$ and C can be variable to adapt to different feature structures. This significantly increases the search speed of the algorithm and reduces the interference of the image noise.

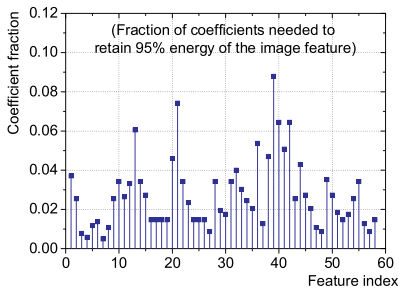

Fig. 4. Haar representation power.

3. FEATURE EXTRACTION BY STATISTICAL DEFORMABLE MODEL

We have built a statistical face model from M manually annotated face images. The statistical face model SFM consists of the following elements:

$$SFM = (P_S, T_S, SP) \quad (4)$$

In Equation (4), P_S is the average face shape over all training samples, $P_S = \{(\overline{x_i}, \overline{y_i}) | 1 \leq i \leq N_{FP}\}$. Note that prior to averaging, all shapes are aligned to a common coordinate system as in ASM [6]. Similarly, T_S is the average local texture attributes, $T_S = \{\overline{\overrightarrow{T_i}}\}$. Similar to ASM, a PCA (Principal Component Analysis) transformation is performed on all the shapes, resulting in a transformation space represented by SP, which embodies the major shape-variation directions.

From the initial estimation of the feature position (Section 2.1), we can overlay a model shape P_S to the real image. The deformation of the model to the real face is then carried out by the following two iterative processes.

1. **Local attribute matching.** For each feature point, a local optimal position is searched along a trace composed of eight radial lines originating from the current model point. The search is based on evaluating the Euclidean distance between the current local feature attribute $\overrightarrow{T_i}$ and $\overline{\overrightarrow{T_i}}$.

2. **Global shape regulation.** The shape P is updated, normalized (invariant to affine transformation) and transformed back to the shape space defined by SP, and each coefficient is limited to maintain a plausible face shape.

4. EXPERIMENTAL RESULTS

We designed a statistical face model from 37 annotated faces, each consisting of 58 feature points [6]. The test set consisted of face images from the BioID face database [7], Aleix database [8] and our own HN2R database. These databases contained various faces taken under different conditions from the training images. The test faces were scaled to roughly the same size (300×300 pixels). Besides this, we tested the complete algorithm within the framework of a live face recognition system employing a normal web camera. The model deformation in all test uses only single-resolution search.

4.1. Experiments with the deformable face model

4.1.1. Convergence ability

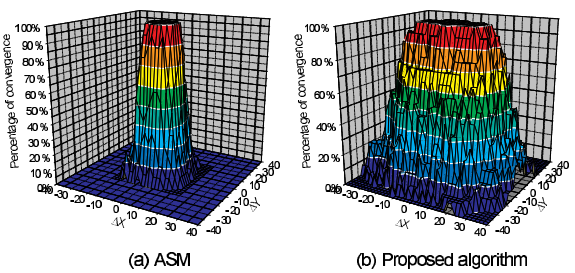

Fig. 5. Comparison of convergence areas.

For each test face, we position the geometric gravity center of the initial model to $(x_r + \Delta x, y_r + \Delta y)$, where (x_r, y_r) is the gravity center of the manually labelled shape. In each case, the model gradually converges to a best fitted shape. If the gravity-center displacement between the fitted shape and the manually labelled shape is D_c, the algorithm is considered converged for all situations with $D_c < 8$. Fig. 5 portrays a comparison of convergence areas. It can be seen that the convergence area of our proposal is almost four times as large as the area covered by ASM.

4.1.2. Feature extraction accuracy

To measure the extraction accuracy, we initialize the model with randomly chosen displacement $\Delta x, \Delta y \in [-12, +12]$ to the reference position. We compute the average point-to-point distance between the fitted model and the manually labelled shape. To reduce the error that may be introduced by manual labelling, we use a *normalized* point-to-point distance metric, in which the feature points along a certain contour are equally spaced. The evaluation results are shown in Fig. 7(a). Up to 90% of the test cases achieve an accuracy of less than 9 pixels, while in ASM, only 60% of the test cases achieve the same accuracy.

Our deformable model fitting takes 30–70 ms to process one face (Pentium-IV, 1.7 GHz), which is comparable to ASM. Fig. 6 shows two examples. It can be seen that our proposed feature extraction is able to achieve correct convergence, even when the model is poorly initialized.

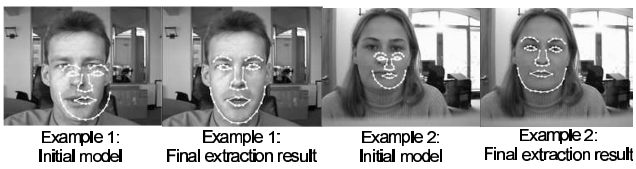

Fig. 6. Feature extraction examples (without performing the initial position estimation step).

4.2. Experiments with the complete algorithm

We have also applied the complete algorithm into a live face recognition system using a web camera (Fig. 7(b)). The system uses a face detector to give the rough position of the facial area from the input sequence (surrounding boxes). For the facial feature extraction, we apply both feature position estimation ('T'-signs) and deformable face model fitting (white dotted shape). The combined steps can process five frames per second and the extraction results are quite good. Note that the algorithm is also robust for the presence of glasses.

5. CONCLUSIONS

In this paper, we obtained an improved and fast facial feature extraction by employing three aspects. First, we introduced a face model incorporating both global shape description and 2-D texture attributes of local features. The 2-D texture attributes can be used to guide the local deformation process. We have found that this clearly outperforms ASM with larger convergence areas. Second, we use Haar-wavelets to efficiently represent local texture attributes, thereby facilitating fast processing and increasing robustness for low-quality images. Third, we used a fast gradient-based matching algorithm to estimate the feature locations, which alleviates erroneous convergence when the model is initialized inappropriately. Our proposed feature extraction technique converges accurately, especially with faces quite different from the training faces.

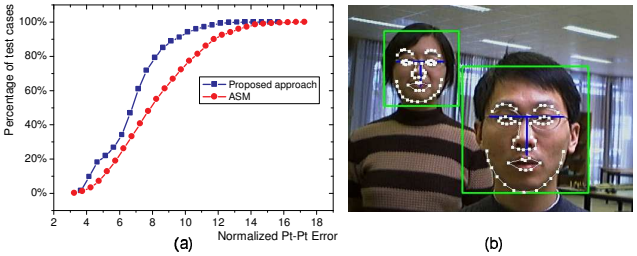

Fig. 7. (a) Accuracy test. (b) Extraction with a live webcam

The proposed technique has given good results when applied in a prototype real-time face recognition system for customized consumer applications.

6. REFERENCES

[1] A. L. Yuille, D. S. Cohen, and P. W. Hallinan, "Feature extraction from faces using deformable templates," in *Proc.CVPR*, 1989, pp. 104–109.

[2] F. Zuo and P. H. N. de With, "Towards fast feature adaptation and localization for real-time face recognition systems," in *Proc. SPIE*, 2003, vol. 5150, pp. 1857–1865.

[3] T. Cootes, "Statistical models of appearance for computer vision," Tech. Rep., Univ. Manchester, 2001.

[4] B. Froba and C. Kulbeck, "Real-time face detection using edge-orientation matching," in *Proc.AVBPA 2001*, 2001, pp. 78–83.

[5] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. CVPR*, 2001, pp. 511–518.

[6] M. B. Stegmaan, "Analysis and segmentation of face images using point annotation and linear subspace techniques," Tech. Rep., DTU, 2002.

[7] HumanScan, "BioID face database," 2003.

[8] A. M. Martinez and R. Benavente, "The AR face database," Tech. Rep., CVC #24, 1998.