Integrated face detection, tracking, and pose estimation

Integrated Face Detection, Tracking, and Pose Estimation

MIYAMA,Masayuki, MATSUDA,Yoshio
Department of Natural Science and Technology, Kanazawa University, Kanazawa, Japan
miyama@t.kanazawa-u.ac.jp

Abstract—**This paper presents a proposal of an integrated method for face detection, tracking, and head pose estimation. We use the de-facto Viola-Jones method for face and face part detection. We adopt affine motion model estimation as a tracking method. The combination enables efficient detection around the search area limited by tracking. Moreover, it reduces false detection because of the consistent processing with earlier results. In addition, the method re-initializes the position and size of the face and face parts in every frame. That initialization immediately corrects tracking jitter. The head pose is estimated using coordinates of both eyes and a mouth relative to the nose as the origin in the coordinate system. The computational cost is low because it uses only those three points. Experimental results show accurate estimation of the head pose. The average error is 6.50 deg in yaw angle, and 7.65 deg in pitch angle.**

Keywords-face and face part detection; tracking; head pose estimation; affine motion model estimation

I. INTRODUCTION

Recently, face detection has become truly practical. Face detection is adopted in digital still cameras for autofocus and auto-exposure. Furthermore, face and face part detection is applied to head pose estimation and face recognition [1][2]. If a computer can infer the intent of a human from the gaze or facial expression, then more natural and easy-to-use human-computer interfaces (HCIs) can be implemented.

The Viola-Jones method is often used for face and face part detection [3]. The method uses Haar-like features and a cascade classifier learned with AdaBoost. This method is efficient, and its detection performance for frontal face images is high. However the profile face detection rate is low. For video processing, we can detect the profile face with the combination of face tracking. Moreover, a method of head pose estimation using face tracking and a 3D head model has been proposed [4]. Although the estimation accuracy of this method is high, its computational cost is also high.

This paper presents a proposal of an integrated method for face detection, tracking, and head pose estimation, aiming to be applicable to HCI in mobile products. We use the de-facto Viola-Jones method for face and face part detection. We adopt affine motion model estimation, which can accommodate shape change, as a tracking method. The combination enables efficient detection around the search area limited by tracking. It also reduces false detection because the processing is consistent with the previous results. In addition, the method re-initializes the face and face part position and size in every frame. That initialization corrects tacking jitter immediately. The head pose is estimated using coordinates of a left eye, a right eye, and a mouth relative to a nose, which is the origin in the coordinate system. Their coordinates are obtained using face part tracking. The computational cost using only these three points is lower than the method using complicated 3D head model. Nevertheless, experimental results show accurate estimation of the head pose, with an average error of 6.50 deg in yaw angle, and 7.65 deg in pitch angle.

This paper is organized as follows. Section 2 presents a description of conventional techniques. Section 3 describes the proposed method. Section 4 presents experimentally obtained results and discussion. Section 5 concludes this paper.

II. CONVENTIONAL METHODS

This section explains face and face part detection. Then affine motion model estimation for face tracking is explained. Conventional methods used for head pose estimation are explained last.

A. Face and Face Part Detection

The object detection method proposed in [3] uses Haar-like wavelet to calculate feature values. The method sets a candidate area in which the object may exist in the image. Then the wavelets of various kinds and sizes are placed at various positions in the candidate area, as presented in Fig.1. A wavelet consists of a white rectangle and black rectangles. A difference between the sum of pixels which lie within a white rectangle and the sum of pixels in black rectangles is calculated. It becomes a feature value for object detection.

The possible wavelets in the candidate area are vastly numerous. It is inefficient to produce a classifier using all possible wavelets as features. Selecting effective features for detection is necessary. The method adopts AdaBoost as a learning algorithm that constructs a strong classifier composed of weak classifiers. They are connected in a cascade. The cascade classifier rejects objects that are apparently false in the early stages, thereby drastically reducing the computation time.

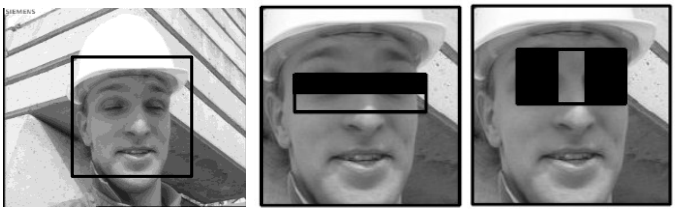

Figure 1. Examples of Haar-like wavelet.

B. Affine Motion Model Estimation

We use an algorithm that adopts the affine motion model with a global illumination change ξ [5]. The model can express a motion of the region such as rotation, zoom in/out, and transformation. All pixel flow in a region can be expressed as a set of linear equations. The model comprises seven parameters as presented below.

$$\Theta_l = [a_1^l\ a_2^l\ a_3^l\ a_4^l\ a_5^l\ a_6^l\ \xi]^{\mathrm{T}} \tag{1}$$

$$\boldsymbol{d}_{\Theta_l}(s) = \begin{bmatrix} u_l(x,y) \\ v_l(x,y) \end{bmatrix} = \begin{bmatrix} a_1^l + a_2^l x + a_3^l y \\ a_4^l + a_5^l x + a_6^l y \end{bmatrix} \tag{2}$$

Therein, s stands for a coordinate of (x, y), and $\boldsymbol{d}_{\Theta_l}(s)$ signifies a motion vector at coordinate s. The algorithm estimates an affine motion model Θ_l corresponding to region R_l. The algorithm estimates Θ_l with accumulation of $\Delta\widehat{\Theta_l}$ obtained using the iterative re-weighted least square method minimizing the sum of a residual $r(s, \Theta_l)$ for each pixel in R_l as the following.

$$\Delta\widehat{\Theta}_l = \underset{\Delta\Theta_l}{\arg\min} \sum_{s\in \mathrm{R}_l} \frac{1}{2} w_s \{r(s, \widehat{\Theta}_l)\}^2 \tag{3}$$

$$r(s, \widehat{\Theta}_l) = \nabla J\left(s + \boldsymbol{d}_{\widehat{\Theta}_l}(s)\right) \boldsymbol{d}_{\Delta\widehat{\Theta}_l}(s) + \Delta\xi_l + J\left(s + \boldsymbol{d}_{\widehat{\Theta}_l}(s)\right) - I(s) + \xi_l \tag{4}$$

In those equations, $I(s)$ represents an illumination value at s in I. The illumination gradient $\nabla I(s)$ is defined as $\nabla I(s) = \begin{bmatrix} I_x(s) \\ I_y(s) \end{bmatrix}^{\mathrm{T}}$, using illumination differentials $I_x(s)$ and $I_y(s)$ in the x and y directions. The residual $r(s, \Theta_l)$ is derived from a linear approximation of the illumination conservation law expressed as $I(s) = J\left(s + \boldsymbol{d}_{\Theta_l}(s)\right) + \xi_l$. Weight w_s is a weight assigned to a pixel at s. Setting up a small weight for each outlier pixel makes the estimation more robust.

The face shape changes according to the viewpoints. The affine motion model estimation supports the shape change. Template matching and feature point matching are well known tracking methods. Template matching does not support the shape change. Feature point matching can estimate the motion of the point accurately, but it cannot estimate the motion of an area independently.

C. Head Pose Estimation

The head pose is expressed as three angles. Yaw is a rotation angle around an axis from a head top through a neck center. Pitch is a rotation angle around an axis through both ears. Roll is a rotation angle around an axis from the nose through the back of head.

Yao et al. proposed a method for head pose estimation using an affine motion model of a face [6]. If we assume a face as a plane, then the face motion can be expressed well by the model. The motion model is obtainable with the method described above using two images of a frontal face and the target face. The head angle α is obtained with the equation $\alpha = F(\Theta, f)$, where Θ represents the affine motion model and f stands for the focal distance of a camera. If we assume that a circle is placed in front of the face and a photograph is taken diagonally to the face, then an ellipse is projected on the image screen. F is a function to obtain a rotation angle so that a circle is projected on the screen. This method can only estimate the rotation angle around the major axis of the ellipse.

Some methods to estimate the head pose using a 2D or 3D head model combined with tracking of feature points have been proposed. A method using the 2D model composed of a triangle of a left eye, a right eye, and a mouth is extremely simple [7]. The pose estimation near the frontal face is accurate, but it can support only the yaw angle. A method using the 3D model estimates the head pose accurately, but its computational cost is high [8].

III. Proposed Method

A. Face Detection and Tracking

A face and face parts are detected according to the following algorithm.

1) Initialization

A search region (SR) for a face is set up on the whole image. A predicted region (PR), a detected region (DR), and an estimated region (ER) for the face are empty. These regions for each face part are empty.

2) Face Detection

A face is searched in the SR. If detected, then go to step 2a. If not detected, then go to step 2b.

a) Detected

The DR for the face is substituted for the ER. If the ER for a face part is empty, then the PR for the part is set up at the standard position with the standard size in the face. The SR for the part is an extension of the PR. Go to step 3.

b) Not Detected

If the PR for the face is not empty and has skin color, then the PR is substituted for the ER. Go to step 3. If the above condition is false, then the face disappears. In that case, advance time and go to step 1.

3) Right Eye Detection

A right eye is searched in the SR. If detected, then go to step 3a. If not detected, then go to step 3b.

a) Detected The DR is the ER. Go to step 3c.

b) Not Detected The PR is the ER. Go to step 3c.

c) Consistency Check

If the ER for the part is without the tolerated range in the face, then the part disappears. In that case, the DR is empty and the ER is the standard area in the face. Go to step 4.

4) Left Eye Detection Same as step 3. Go to step 5.

5) Nose Detection Same as step 3. Go to step 6.

6) Mouth Detection Same as step 3. Go to step 7.

7) Revision of Face Region

If the face is undetected and all face parts are detected, then the ER for the face is revised according to the positions of the face parts. Go to step 8.

8) Face and Face Parts Tracking

Affine motion model estimation is executed using the current image and the next image. The face and face parts are translated using the motion model of the face. Advance time and go to step 2.

The face parts are detected efficiently only within the face region. The face motion model is applied to both the face and face parts. This method results in the robust detection against partial occlusion. The combination of detection and tracking enables efficient detection only around the search area limited by the tracking. It also reduces false detection because of processing continuity between two frames. Because of the combination, a profile face can be estimated accurately. In addition, the method re-initializes the position and size of the face and face parts in every frame. The initialization corrects tracking jitter immediately. Furthermore, undetected face is revised in terms of position and size using detected face parts.

B. Pose Estimation

Coordinates of the face parts are expressed as presented in Fig. 2. The nose position is set up as the origin. Coordinates of the other parts are defined from the origin. The distance between the face parts is normalized so that the coordinates are independent of the image size. For example, they are normalized by defining the distance between two eyes as a unit distance. Using these coordinates, angle α can be approximated by the linear equation below.

$$\alpha = \boldsymbol{a}^T \boldsymbol{x} = a_1 x_{re} + a_2 y_{re} + a_3 x_{le} + a_4 y_{le} + a_5 x_m + a_6 y_m + b \quad (5)$$

Therein, (x_{re}, y_{re}), (x_{le}, y_{le}), and (x_m, y_m) respectively represent the coordinates of the right eye, left eye, and mouth. Using experimental data of coordinate $\boldsymbol{x}$ and angle α, unknown parameters $\boldsymbol{a}$ can be calculated using the least square method minimizing a square norm of error ε expressed as $\varepsilon = \alpha - \boldsymbol{a}^T \boldsymbol{x}$.

IV. Experimental Results

A. Face Detection and Tracking

1) Experimental Condition

We used a face detection program in OpenCV for the experiment. We used the Foreman sequence, in which the face moved left and right frequently and largely. The number of frames was 176. True regions of a face and face parts in a frame were determined manually. We determined that the answer was right when an overlap of the true region and the estimated region was more than 50% of the true region. We compared the proposed method with the method only using Haar-like detection. The combination method of Haar-like detection and tracking with template matching was also compared. The latest region detected as a face or a face part was used as a matching template.

2) Results and Discussion

Tables I and II show the experimentally obtained results. The face detection rate of Haar-like method is low. This method can detect frontal faces. However, profile faces are not detected. Many false detections of face parts were made. False detection of face parts often occurs because the standard position and size are not considered. Fig.3(a) and

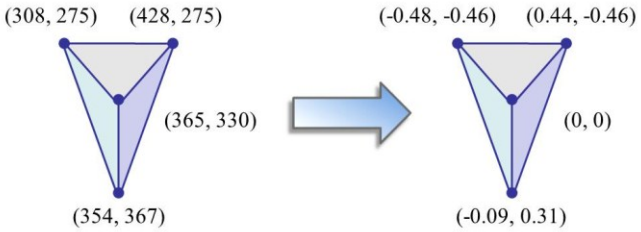

Figure 2. Coordinate expression of face parts.

TABLE I. Rate of Correct Answer

Objects	Haar (%)	Template (%)	Proposed (%)
Face	39.43	96.00	100.0
Left Eye	78.29	95.42	100.0
Right Eye	89.14	85.71	98.29
Nose	66.29	94.29	98.29
Mouth	84.57	88.57	97.14

TABLE II. Number of False Detections

Objects	Haar (#)	Template (#)	Proposed (#)
Face	106	7	0
Left Eye	38	8	0
Right Eye	19	25	3
Nose	59	10	3
Mouth	27	20	5

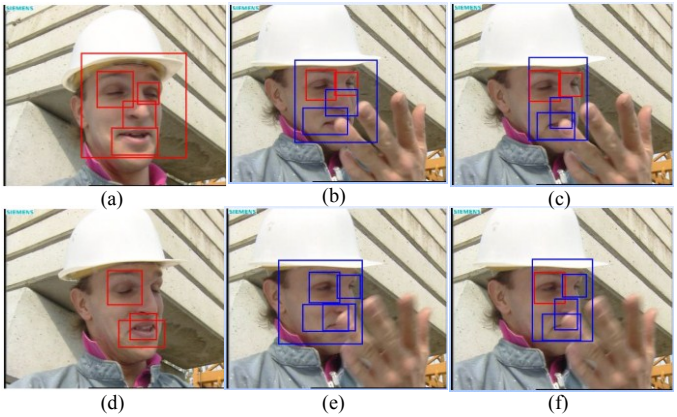

Figure 3. Examples of experimentally obtained results.

3(d) show processing examples. The face, the left eye, and the nose are not detected because the face direction is diagonal in 3(d).

The other methods produce better results. The proposed method achieves an even higher rate of right answers than the template method because the proposed method supports shape changes. The processing examples are shown in Fig. 3(b), 3(c), 3(e), and 3(f). A hand is shown moving before the face in the frame. Template matching fails to track the nose and the mouth because of occlusion, as shown in 3(b) and 3(e). The proposed method produces better results in spite of occlusion, as shown in 3(c) and 3(f). A face motion models is applied to both the face and face parts, resulting in robust detection against partial occlusion. They also show that the proposed method supports shape change of the face.

Figure 4. Example data in Pointing'04 database

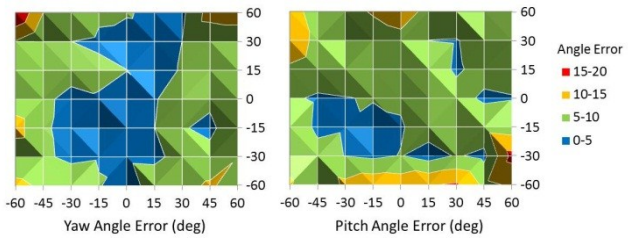

Figure 5. Yaw and pitch angle error.

B. *Head Pose Estimation*

In this experiment, we intended to estimate yaw and pitch angle.

1) *Experimental Condition*

We used the Pointing'04 database in this experiment [9]. This database comprises data of 15 people. The yaw angle and pitch angle vary from -90 to 90 deg in steps of 15 deg. The sample data are portrayed in Fig. 4. We used data of 10 people from the database images with the head angle from -60 to +60 deg. First, we manually adjusted the head position for each image in the center roughly. Then we made a sequence of images in the order of angles, as portrayed in the Fig. 4. Face part coordinates were obtained by application of the proposed algorithm to the sequence.

An experimental procedure is the following: 1) obtain center coordinates of face parts for a person to learn, 2) learn unknown parameters using the coordinates of face parts and true angle, 3) estimate yaw and pitch angle using the parameters and coordinates of face parts for a person to evaluate, and 4) calculate the error between the true angle and the estimated angle.

We conducted experiments of two kinds. In the first, the person used for learning was the same as the person used for evaluation. In the second, we learned unknown parameters using data of nine people and evaluated the other one person. In these experiments, we obtained an average angle error with estimated angles of 10 people.

2) *Results and Discussion*

In the first experiment, Fig. 5 shows the respective yaw and pitch angle errors. The horizontal axis shows the yaw angle. The vertical axis shows the pitch angle. Color represents the error. Yaw and pitch angle errors were within 10 deg in the whole range. The error was small near the front. The average yaw angle error was 6.50 deg. The average pitch angle error was 7.65 degree. In this case, we were able to estimate the head pose accurately, although calibration for each person was necessary. In the second experiment, the average yaw angle error was 10.17 deg. The average pitch angle error was 13.96 deg. These can easily stand comparison with conventional results obtained using the same database [1].

V. CONCLUSION

This paper proposed an integrated method for face detection, tracking, and head-pose estimation. The combination of face detection using the Viola-Jones method and tracking by the affine motion model estimation enables detection of a profile face. We proposed head-pose estimation using coordinates of both eyes and a mouth with the origin of a nose. For learning and evaluating using the same person, the yaw angle error was 6.50 deg, and the pitch angle error was 7.65 deg within a range of +-60 deg. The head pose can be estimated accurately in this case, although calibration for each person is necessary. Greater accuracy within a wider range might be obtained using a nonlinear equation such as SVM regression to approximate the head pose. Detailed comparisons with the other methods remain as a subject for future work.

ACKNOWLEDGMENT

This research was supported by a Ministry of Education, Culture, Sports, Science and Technology, Grant-in-Aid for Scientific Research (C), 22560325, 2010-2012.

REFERENCES

[1] E.M. Chuntorian, M.M. Trivedi, "Head Pose Estimation in Computer Vision: A Survey," IEEE Trans. Pattern Analysis And Machine Intelligence, vol.31 , no.4 , pp.607-626 , April 2009.

[2] W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "Face Recognition: A Literature Survey," ACM Computing Surveys, vol.35, no.4, pp.399-458, December 2003

[3] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," In Proc. of CVPR, 2001.

[4] D. DeCarlo and D. Metaxas, "Optical flow constraints on deformable models with applications to face tracking, " Int'l. J. Computer Vision, vol. 38, no. 2, pp. 231-238, 2000.

[5] J.M. Odobez, P. Bouthemy, "Direct incremental model-based image motion segmentation for video analysis," *Signal Processing*, vol.66, pp.143-155, 1998.

[6] P. Yao, G. Evans, and A. Calway, "Using Affine Correspondence to Estimate 3-D Facial Pose", IEEE International Conference on Image Processing. I. Pitas, pp. 919-922, Oct, 2001.

[7] A. Nikolaidis and I. Pitas, "Facial feature extraction and pose determination, " Pattern Recognition, vol. 33, no. 11, pp. 1783-1791, 2000.

[8] M. Malciu and F. Preteux, "A robust model-based approach for 3D head tracking in video sequence," in Proc. IEEE Int'l. Conf. Automatic Face and Gesture Recognition, 2000, pp. 169-174.

[9] Pointing'04-Visual Observation of Deictic Gestures - "http://www.prima.inrialpes.fr/Pointing04/" .