Face pose estimation of SDM based on multi-template matching

Hu Chen¹, Chaoqian Gao² and Tianping Li*  
¹School of Physics and Electronic Science, Shandong Normal University, Jinan, Shandong, 250358, China  
²School of Physics and Electronic Science, Shandong Normal University, Jinan, Shandong, 250358, China  
³School of Physics and Electronic Science, Shandong Normal University, Jinan, Shandong, 250358, China  
*Corresponding author’s e-mail: sdsdltp@163.com

Abstract. This paper presents the face pose research of SDM algorithm and POSIT algorithm with multiple template matching. Firstly we detect the input face image, locate the feature points in the detected facial area, then compare the traditional SDM algorithm with the improved multi-template SDM algorithm, and then select the multi-template SDM algorithm. Feature point location is finally applied to the POSIT algorithm. Experimental results prove the effectiveness of the method and make it competitive.

1. Introduction

1.1. Background
With the development of artificial intelligence, there are more and more applications of artificial intelligence technology. Face recognition technology is the key technology in the field of artificial intelligence. These technologies are widely used in security monitoring and access control systems. Research value, but although face recognition has been developed for many years, it has not achieved the expected results.

Head pose estimation is an important part of face recognition, and its research also involves many disciplines, including computer vision, pattern recognition, image processing and so on. It has the characteristics of interdisciplinary research, difficult and challenging research, and has a wide range of application prospects. It has been widely used in many fields.

1.2. Main Work of the Study
Fanelli et al. [6] described a head pose estimation [1] method based on random regression forest. To train the classifiers, they used a 3D deformable model to generate a dataset of 50K images [1] to randomly generate poses, and the percentage of test instances that correctly identified within the threshold was 90.4%. Methods like [3], [8], [10], [4] still rely on face detection, such as eyes, nose, mouth and other features. When facial features are only partially displayed in the scene due to extreme head poses (for example, excessive deflection angles), the accuracy of the system decreases. In order to deal with large pose human faces, a different initialization correspondence estimation method is needed.
1.3. Organizational Structure
The structure of this paper is as follows: Section 2 mainly introduces the preliminary preparation of head pose estimation, and Section 3 introduces the main methods including SDM algorithm with multiple template and POSIT algorithm. Section 4 presents the experimental results. Section 5 presents the conclusions of this article.

2. Preliminary Preparation
Head pose estimation usually requires the face area to be located in advance in order to further analyze the head pose of the face image. The preliminary work of head posture usually includes face detection and facial feature point location. According to the located face area, the position of the key points of the face is further determined, and then the head pose is estimated according to the established face model.

2.1. Face Detection
This paper focuses on the face detection algorithm of Haar-like features combined with AdaBoost classifier. Some weak classifiers are generated by learning the Haar features of the training samples, and then a strong classifier is formed by cascading. This strong classifier is used to detect the face area. Haar features can currently be divided into 3 categories: edge features, linear features, and center-enclosed features. Because the length, width, and placement of Haar feature templates vary widely, the amount of calculation is too expensive when performing feature calculations. For any point in the figure, the expression of the point integral image value is as follows:

$$i(x, y) = \sum_{x \leq x', y \leq y'} i(x', y')$$ (1)

Where ii (x, y) represents the value of point (x, y) in the integrated image, and i (x', y') represents the pixel value of point (x', y') in the original image.

The idea of AdaBoost algorithm is to increase the weights of the wrongly classified samples from the previous round of weak classifiers. This mainly uses the weighted majority voting method to increase the weak classification with a small classification error rate. The weight of the classifier makes it play a larger role in the voting, and reduces the weight of the weak classifier with a large classification error rate, which makes it play a smaller role in the voting.

2.2. Facial Feature Point Location
The purpose of facial feature point positioning is to obtain a set of feature points, which represent the coordinates of key positions such as ears, eyes, nose tip, and mouth corners on the face image. Learn some static probability models or some information from these training data. Align the estimated shape with the real shape to determine the precise position of the face on the picture, and lay the foundation for subsequent head pose tasks.

There are many facial feature point localization algorithms. SDM uses the method of optimizing feature point coordinates directly for feature point detection. SDM algorithm is a gradient descent algorithm with good feature point positioning effect. As shown in the figure1, it is a flowchart of the facial feature point location algorithm of the SDM algorithm.

Given a picture containing m pixels $$d \in R^{m \times d}$$, $$d \in R^{m \times d}$$ represents p feature points on the picture, and h represents a non-linear feature extraction function. Our goal is to return $$x_0$$ to the correct shape $$x^*$$ of the face on the average face shape of the true shape of all known samples. That is to find $$\Delta x$$ that minimizes the following $$f(x_0 + \Delta x)$$:

$$f(x_0 + \Delta x) = \|h(d(x_0 + \Delta x)) - \phi_c\|$$ (2)

Where $$\phi_c = h(d(x_c))$$ represents the SITF features extracted from the true feature points of the face.

It is usually impossible for the SDM algorithm to achieve the optimal value of the function in one iteration. Correspondingly, a series of gradient descent directions $$\{R_k\}$$ and offset $$\{b_k\}$$ produced by SDM are also obtained by using equation (3):

$$x_k = x_{k-1} + R_{k-1} \phi_{k-1} + b_{k-1}$$ (3)
For each image, from the many possible initial points $x_0^i$, the error between the predicted point and the target point is calculated and minimized. $R_0$, $b_0$ can also be solved using equation (4).

$$\arg \min_{R_0, b_0} \sum \sum \|\Delta x_i^i - R_0 \phi_0^i - b_0^i\|^2$$

After $R_0$ and $b_0$ are obtained, iteration can be performed. After 4 to 5 steps of iteration, the direction of gradient descent $\{R_k\}$ and offset $\{b_k\}$ can be obtained generally.

Figure 1 shows the implementation of the SDM algorithm. The first is the input image, the middle is the average face obtained by initialization, and the last is the final positioning effect image.

3. Method

Figure 2 shows a block diagram of the head pose study. The process mainly includes the following steps: 1) accurate face detection on the 2D image and extraction of the main feature points of the face; 2) determining the correspondence between the facial features in the 2D image and the 3D face model[5]; 3) Use common pose estimation methods, such as POSIT (Proportional Orthogonal Projection Iterative Transformation Algorithm), to estimate the head pose.

3.1. Multi-template SDM

In the previous chapter 2, we have introduced the SDM algorithm for facial feature point location as a preliminary preparation for facial feature location. The effect of facial feature points is good or bad, which directly affects the head pose estimation. The traditional SDM algorithm, when the pose of the face does not change greatly, can achieve better convergence effect after several iterations. However, when the deflection angle of the head is too large, it is difficult to truly fit the actual human face. Therefore, this paper proposes the positioning of facial feature points on various templates, and divides the pose of the face into five types: left side, left side, front, right side, and right side from the horizontal direction.

3.2. POSIT Algorithm

The POSIT algorithm can estimate the three-dimensional pose information of the target object from two-dimensional images by iteratively iterating through processes such as orthogonal projection and size transformation. The POSIT algorithm takes a three-dimensional head model, two-dimensional main feature points of the face, and camera parameter information as input data, and iteratively obtains the target's three-dimensional pose data (attitude matrix $R$ and translation matrix $T$). In the face image, the key point position $U$ of the face corresponds to a two-dimensional coordinate system, and the three-dimensional face model $S$ corresponds to the target coordinate system. The calculation process of the
POSIT algorithm is as follows [9].

The unknown head pose with solution is represented by the 3*3 selection matrix R and the translation vector T. Perspective photography transform is

\[
x_0 = \frac{f}{Z}X, \quad y_0 = \frac{f}{Z}Y \quad (5)
\]

The internal parameter matrix of the camera is \([f_x, f_y, u_0, v_0]\). After a series of transformations, we can get

\[
\begin{align*}
wx_0 &= sRx_1 + sRy_1 + sRz_1 + sT_x \\
wy_0 &= sRx_2 + sRy_2 + sRz_2 + sT_y
\end{align*} \quad (6)
\]

Therefore, both the rotation matrix R and the translation vector T can be solved. It can be concluded from the foregoing that the POSIT algorithm is based on a given three-dimensional face model S, and iteratively calculates the final rotation matrix R and translation parameters T [2], so that the model is finally projected onto the face image through rotation and translation. And fitting the corresponding key point U on the face, and the formula can be expressed as:

\[
\arg\min_{R,T} \prod_{i=1}^{n} \left( \mathbf{S} \mathbf{R} \mathbf{T} - \mathbf{U}_i \right) \quad (7)
\]

According to the following three formulas, three angles can be obtained

\[
\begin{align*}
yaw &= a \tan(2(R_{12}, R_{31})) \\
pitch &= a \tan(2(-R_{31}, \sqrt{R_{12}^2 + R_{13}^2})) \\
roll &= a \tan(2(R_{21}, R_{11}))
\end{align*} \quad (8)
\]

4. Experimental Results and Discussion

In the preparatory work, this paper selects self-built image database to verify the face detection algorithm. The following figure 3 shows several self-built and FERET face detection renderings. We selected the front face and the pictures with a certain deflection angle, and we can see that no matter whether the front face or a certain deflection angle, a better detection effect can be achieved.

Figure 4. Face detection

In the previous preparations, the facial feature point location was also included. We introduced the implementation of the SDM algorithm earlier, but when the head deflection is too large, the implementation effect is not good. We propose an SDM algorithm based on 5 types of templates. First, train facial pose samples, classify them, and then select the corresponding pose for SDM localization. Figure 6 shows the comparison between the traditional SDM algorithm and the improved multi-template SDM algorithm. The leftmost image in Figure 6 is the positioning map of the traditional SDM algorithm. We found that the positioning effect on the nose is not ideal. Figure 6 on the right is the improved algorithm effect diagram of this paper, which has a good positioning effect.

Figure 5. Comparison of SDM and multi-template SDM

Through the previous preparations, face detection detects the face area and locates feature points on the face area. With the three-dimensional head model, the coordinate information of the facial feature
points detected in the input image, and the camera parameter information as input parameters, the rotation matrix \( R \) and translation vector \( T \) [7] can be calculated by the POSIT algorithm, and finally the corresponding three-dimensional space angle can be obtained by the previous formula.

In order to verify the effectiveness of the head pose estimation algorithm, 100 volunteers were selected from the CAS-PEAL-R1 data set, and 5 poses were selected from the images of each volunteer. A total of 500 images were used to estimate the head pose, and the error between the experimental results and the true value was determined. Images that are all greater than 6 degrees are inaccurate pose estimation images. In the experimental results of pose estimation, there are 22 inaccurately estimated images and 478 accurate pose estimation images, with an accuracy rate of 95.6%. The experimental results are shown in Table 1. As can be seen from Table 1, the accuracy of the POSIT algorithm in estimating the head posture of a grayscale image can reach 95.6%, and the experimental results are ideal. Figure 7 is a result of head posture estimation performed on data created on the Internet.

| Table 1. Accuracy tested on the CAS-PEAL-R1 dataset |
|-----------------------------------------------------|
| Total test pictures (frames) | 500 |
| Correctly identify pictures (frames) | 478 |
| Correct recognition rate (%) | 95.6% |

Figure 6. Head pose estimation results

5. Conclusion

This paper presents a head pose estimation study based on the POSIT algorithm. We propose a face feature point location based on SDM based on multi-template matching, which improves the accuracy of the head pose estimation before being applied to the POSIT algorithm.

Acknowledgments

We are grateful to the editors and referees for their invaluable suggestions for improving the paper.

References

[1] Tulyakov S, Vieriu R, Semeniuta S, et al 2014 Robust Real-Time Extreme Head Pose Estimation IEEE 2014 22nd ICPR pp 2263-68.
[2] Breidt M, Bulthoff H H, Curio C, et al 2016 Accurate 3D head pose estimation under real-world driving conditions: A pilot study international conference on intelligent transportation systems pp 1261-68.
[3] Sun Y and Yin L 2008 Automatic pose estimation of 3d facial models ICPR pp 1–4.
[4] Weise T, Bouaziz S, Li H, et al 2011 Realtime performance-based facial animation ACM TOG 30(4) pp 77–85.
[5] Huang Y, Zhang X, Fan Y, et al 2012 Reshaping 3D facial scans for facial appearance modeling and 3D facial expression analysis Image and Vision Computing 30(10) pp 750-761.
[6] Fanelli G, Dantone M, Gall J, et al 2012 Random Forests for Real Time 3D Face Analysis IJCV 101(3) pp 437–458.
[7] Ching-Fa Chen 1994 The mark-based approaches for motion estimation from 3-D point sets Journal of the Chinese Institute of Engineers 17.1 pp 43-53.
[8] Lu X and Jain A K 2006 Automatic feature extraction for multiview 3D face recognition international conference on automatic face and gesture recognition pp 585-590.
[9] Yang A, Li B, Yan Z, et al 2019 A Bi-Directional Carrier Sense Collision Avoidance Neighbor
Discovery Algorithm in Directional Wireless Ad Hoc Sensor Networks Sensors 19(9).

[10] Chang K I, Bowyer W, Flynn P J, et al 2006 Multiple Nose Region Matching for 3D Face Recognition under Varying Facial Expression IEEE Transactions on PAMI 28(10): pp1695-1700.