H-047

A Study of
Extracting 3D Facial Feature from Kinect's Image by Integrating ASM and Depth Map

LI YAN[†] LUO DAN[†] JUN OHYA[†]

1. INTRODUCTION

Facial feature extraction is one of the most important issues in computer vision. It can be used in many applications such as face recognition, facial expression recognition, or facial animation, etc. Typical facial feature extraction methods extract 2D facial contour information from common input facial images. However, 2D feature cannot represent the facial posture or orientation precisely. In order to improve the accuracy of feature extraction, to reconstruct 3D face model or to make lifelike 3D facial animations, many of the researchers have begun the study of extracting 3D facial feature.

In this paper, we present a proposal of integrating 2D facial feature with depth map from IR sensor to extract 3D facial feature points.

In order to extract 3D facial feature, first we use Active Shape Model (ASM) [1] to obtain 2D facial feature points. ASM is a flexible template based facial shape extraction method. It uses trained template as constraint to find the most similar contour and represents facial features by continuous 2D vector. It can acquire satisfying results without losing the efficiency. Thus, in this paper, standard ASM is applied in color image to find 2D facial features, which is described in Section 2.

There are many approaches to obtain depth information, such as reconstruction from stereo cameras or "structure from motion" from video sequence. In this paper, we use Microsoft Kinect to obtain the depth map so that the distance between the object and Kinect sensor can be obtained. As Kinect is household equipment and it has its open-source code, it is easy to use and improve. It uses an ordinary digital camera and an IR sensor, which can measure the depth of the chosen point on the screen. Therefore, in this paper, we obtain and modify the feature points' depth information from calibrated Kinect's depth map, in order to integrating 2D facial feature points with depth information. We describe this process in the Section 3.

The experimental results are shown in the Section 4. And we conclude this paper in the Section 5.

2. ACTIVE SHAPE MODEL

We use ASM to extracting 2D feature points from Kinect's color image. ASM is a flexible template matching face recognition methods. By varying the model parameters, its goal is to minimize the difference between the model and the input image.

2.1 Shape Model

In ASM, any of the facial features can be represented as n points. Usually we use manually annotation landmarks on training images. The contour of every sample X_i can be represented as the coordinate of landmarks $\{(x_n, y_n)\}$.

Then we use Procrustes Analysis to align the training shape model in the same co-ordinate frame so as to remove variations caused by position and scale of each training set. The main idea is to minimize the sum of squared distances $\sum_{i=1}^{N}|X_i-\bar{X}|^2$ by global shape normalizing transformation (translation, rotation and scaling). Then we extract the principal components of the training data with linear transformation by Principal Component Analysis (PCA). The training set S can be represented by the following equation:

$$S \approx \bar{X} + pS \qquad (1)$$

where $pS = \sum_{i=1}^{t} p_i S_i$, p_i is the i_{th} largest eigenvalue and S_i is the i_{th} t dimensional eigenvector.

From equation (1) the shape parameter p are obtained, which can control many model points only by t parameters in order to fit the chosen training set to the target image.

2.2 Fitting

We use statistical model of the landmark's gray-level profile in the training set to find the shape feature most similar to the shape model, in the input image. We obtain k pixels' gray-level on the both sides of the chosen landmark's normal in each of the training set. Thus, we get covariance matrix of chosen landmark's normalized derivative profile in the training set.

Then, in order to locate the landmark on the target shape, we calculate the Mahalanobis distance of the sample landmark from the model mean $\bar{g}$ by:

$$d = (g_s-\bar{g})C_{gs}^{T}(g_s-\bar{g}) \qquad (2)$$

where g_s is the s_{th} landmark on the target shape and C_{gs}^{T} is its covariance matrix from $\bar{g}$, then the minimal distance is chosen as the final converged point.

When fitting, the shape model is used as constraint template to describe the variables of the initial shape. The goal of the fitting is to let the above profile converge to the input face image. Similarly, we can use Procrustes Analysis to align the initial shape model to the target shape. First we extract the translation, rotation and scaling variables as the pose parameters from the initial shape model. As described in Eq. (1), the shape vector can be deformed through adjusting eigenvalue p. Therefore, we substitute p to the attitude parameters to establish a new shape model which is closer to the input facial feature contours.

3. OBTAIN DEPTH MAP

Feature points' related depth information can be obtained by Kinect's IR sensor in order to extract and rebuild facial feature points in 3D space.

3.1 Calibrating the Depth Map with Color Image

Since Kinect's IR sensor and color camera are not exactly in the same position, IR-image need to be aligned with RGB image at first in order to calibrate depth map and color map in the same view port. Pinhole Camera Model [2] is applied to calibrate the depth camera to color camera. It is formed by projecting real-world coordinate M into the screen coordinate M' using a perspective transformation as follows:

[†] Graduate School of Global Information and Telecommunication Studies, Waseda University

$$M' = A \cdot [R|T] \cdot M \tag{3}$$

where A is camera's intrinsic parameters such as base line, focal length and some other distortions. $[R|T]$ is transformation matrix.

The base line and focal length of Kinect's depth camera are intrinsic [3], so we can calculate a matrix of the depth map's screen coordinate without intrinsic. Then we can transform the matrix by translation and rotation coefficients which are also intrinsic. Next, we substitute color camera's base line and focal length into the matrix. Thus we can get a calibrated matrix of depth map.

3.2 Unit Transformation

Kinect uses laser grid to obtain objects' depth information, so it presents the real-world depth coordinate in depth map, yet facial feature points are extracted from color image with the screen coordinate system. Therefore, we need to transform the real-world coordinate system to the screen.

Firstly, we calibrate depth map into vertical distance to reduce the error. Kinect's depth map stores the distance between object and focal point. However, not all of the feature points are located in the center of the screen. So we need to calibrate the initial depth information to the vertical distance between the focal point and the plane where the point lies on. The j_{th} calibrated distance h_j can be calculated by following proportional relationship:

$$h_j = d_j * (f/D_j) \tag{4}$$

where f as an intrinsic which is the focal length of Kinect, d_j is the depth information obtained by Kinect and D_j is the distance between the feature points on the screen and the focal point.

Then we transform the calibrated depth map to the screen coordinate system. The proportion of the transformed depth information to real world distance is approximately equals to the focal length f to mean calibrated depth $\bar{h}$. Therefore, over the screen coordinate system, the transformed depth information H_j can be referred as:

$$H_j = (f/\bar{h}) * h_j \tag{5}$$

where H_j is used as depth information of our shape model.

4. EXPERIMENTAL RESULTS

The feature extraction is performed by the training set consisting of the 240 images of 40 persons' faces as our training set. Each of the training set has 58 manually annotated landmarks. Then Adaboost Frontal Face Detector is used as template to detect face in color image and we use ASM algorithm to extract 2D feature points.

The experiments are programmed by C++ with Visual C++ 2008 in Windows system. Both of the color image and depth map are captured by Microsoft Kinect.

In order to judge the accuracy of ASM in deferent distances, we use Fig.1 to show whether the feature points are placed on the facial contours. Then we compared 3D facial feature extraction results of ASM combined with depth map which is proposed by this paper in deferent distances. The results are shown in Fig.2. Respectively, we draw the 2D feature points on the color image, depth map and color-depth mixed image and perform the generated 3D facial mask from four different views in order to inspection.

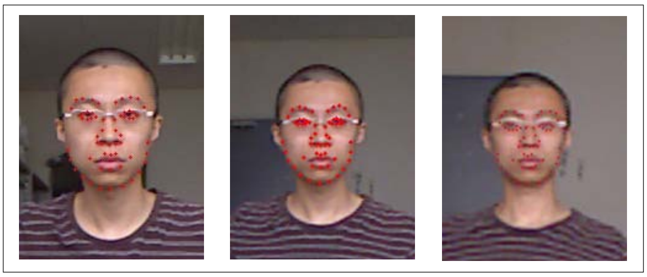

Fig.1. The 2D feature extraction results from color image in 1000, 1400, 2000 distance (unit: mm).

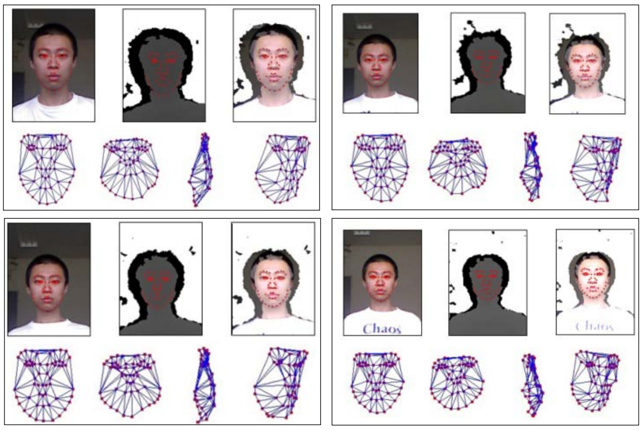

Fig.2. 2D feature points drawn on 3D feature extraction results in 1065, 1188, 1218, 1390 distance (unit: mm).

We successfully extracted 2D facial feature points by ASM from color image and then use the calibrated depth information as the parameter to establish 3D facial feature points. However, we noticed some points are located in the wrong depth position and some points are not located on the contour correctly, because of the low resolution of Kinect's color camera. ASM does not work well for very long distances, but on the other hand Kinect's IR sensor performs with better accuracy for longer than 1200 mm. Therefore, we need to adjust with respect to the distance in advance.

5. CONCLUSION

This paper has proposed a framework for extracting 3D facial features by combining 2D facial feature points with depth map acquired by Kinect. We obtained quite promising experimental results, but there are some extraction errors.

The future work of the study includes adding more landmarks into the training set in order to perform more details such as pupils, nose tip or lips. Then some other application such as face reconstruction by obtaining the facial texture and warp it onto the 3D face mesh can be implemented. And also we will consider more about efficiency of the extraction method in order to implement it in real-time by Kinect.

References

[1] T. Cootes, C. Taylor, D. Cooper, J. Graham, "Active Shape Models - Their Training and Application", Computer Vision and Image Understanding, Vol. 61, No. 1 (1995).

[2] David A. Forsyth, Jean Ponce. *Computer Vision, A Modern Approach*. Prentice Hall. p4-6 (2011).

[3] Smisek. J, Jancosek. M, Pajdla. T, "3D with Kinect", IEEE Computer Vision Workshops (ICCV Workshops), p1154-1160 (2011).