FACIAL FEATURE EXTRACTION FROM VIDEO IMAGE

G. Rajitha
UG Student, SVEC, Tirupathi.

ABSTRACT

The main idea of our method is to use the face detection algorithm with a kinect camera to locate human head and estimate head pose. A depth AAM Algorithm is developed to locate the detailed facial features. The head position and pose are used to initialise the AAM global shape transformation which guarentees the model fitting to the correct location. The depth AAM algorithm takes four channels-R, G, B, D into our consideration which combines the colors and the depth of input images. To locate facial feature robustly and accurately, the weights of RGB information and D information in global energy function are adjusted automatically. We also use the image pyramid algorithm and inverse compositional algorithm to speed up the iteration. Experimental results shows that our depth AAM algorithm can effectively and accurately locate facial features from video objects in conditions of complex backgrounds and various poses.

Keywords*: Kinect camera; Head pose estimation; Randomized decision trees; Depth AAM algorithm.*

I. INTRODUCTION

Facial feature extraction from video image is aimed to locate the exact positions and shapes of the components of a human face from the input images, including the eyes, nose, mouth and outline. It provides the basic foundation for face recognition, gesture expression analysis, human-computer interaction studies and so on. Recently, people have proposed various facial feature extraction algorithms which can be divided into two categories depending on the data dimension, either based on 2D images [1, 2, 3] or based on 3D images [4, 5, 6]. Because of the limitations of the existing face detection technology, locating face feature based on 2D images would be greatly impacted by the conditions of complex background and various pose [7]. Locating face feature based on 3D image needs the capturing systems which are usually very expensive to acquire or operate. Significant improvements have been made when the consumer price Kinect camera [8] was launched in 2010, which has the potential to revolutionize many fields of research, including computer vision, computer graphics and human computer interaction. The Kinect camera has great advantages in human posture recognition, although it is unable to locate facial features directly. However, the depth information provided by the camera plays an important role in the head location and head pose estimation.

The AAM algorithm [3, 9] has a profound mathematical background, excellent characterization capabilities and it takes full advantage of a priori information about the object model. A large amount of facial feature location algorithms have been constantly proposed. Rajitha et al. [10] proposed a new AAM algorithm based on Fourier transform which fitting appearance model in the frequency domain that could be better adapted to LK image alignment algorithm. Mircea et al. [11] proposed a texture vector normalization method and a new AAM algorithm using a novel color space. Xiao et al. [12] proposed 2d+3d active appearance models which uses 3D morphable models to capture the shape and texture variations of faces. To differ the normal image and curved image, we are calculating the mean square error(MSE). The MSE is the cumulative squared error between the normal and the curved image.

II. HEAD POSE ESTIMATION

The AAM algorithm's fitting efficiency is closely concerned with the initial position and its model examples, and directly prevents its potential applications. A widely used pose estimation algorithm is the Adaboost method proposed by Viola and Jones [13] and its improved versions. The face detection is implemented with the combination of a bunch of weak detectors, and then it uses a threshold-based image processing method to process and analysis the face image obtained. First, the Adaboost algorithm for non-frontal face detection accuracy is relatively low, and sensitive to the background and the body gesture. The match is still working when the image does not contain a face, which is a waste of time. Secondly, the face image processing with a threshold value is highly subjective. Light conditions or make-up would have a serious impact, which can easily make an estimation error.

The Head Pose Estimation based on the depth of the image [7, 14] is more robust to a complex background and various body poses. The depth imaging technology has advanced dramatically over the last few years, especially with the launch of Kinect camera. Pixels in a depth image indicate the calibrated depth in the scene, rather than a measure of intensity or color. Kinect camera offers several advantages over traditional intensity sensors, working in low light levels, giving a calibrated scale estimate, being color and texture invariant, and resolving silhouette ambiguities in pose. It greatly simplifies the task of background subtraction. The pairs of depth and body part images are used as fully labeled data for learning the classifier.

Feature calculation should be as simple and effective as possible. Classifier can use the GPU for each pixel in parallel computing that could speed up the efficiency. Given a larger computational budget, one could employ potentially more powerful features based on, for example, depth integrals over regions, curvature, or local descriptors e.g. The simple depth comparison features is more effective, which is computed as follows [15]:

$$f_\theta(I,x) = d_I\left(x + \frac{u}{d_I(x)}\right) - d_I\left(x + \frac{v}{d_I(x)}\right) \quad (1)$$

where $d_I(x)$ is the depth at pixel x in image I, and parameters $\theta = (u,v)$ describe offsets u and v, where 'maximum probe offset' means the maximum absolute value for both x and y coordinates which covers almost all the body. The normalization of the offsets by $\frac{1}{d_I(x)}$ ensures the features are depth invariant. If an offset pixel lies on the background or outside the bounds of the image, the depth probe $d_I(x)$ is given a large positive constant value. The feature give a large positive response for pixels x near the top of the body, but a value close to zero for pixels x lower down the body, may help find thin vertical structures such as the arm. The design of these features was strongly motivated by their computational efficiency: no pre processing is needed; each feature need only read at most three image pixels and perform at most five arithmetic operations; and the features can be straightforwardly implemented on the GPU.

Randomized decision trees and forests are effective multiclass classifiers for human pose recognition [16, 17, 18]. A forest is an ensemble of T decision trees, each consisting of split and leaf nodes. Each split node consists of a feature f_θ and a threshold τ At the leaf node reached in tree t, a learned distribution $P_t(c|I, x)$ over body part labels c is stored.

Each tree is trained on a different set of randomly synthesized images. A random subset of 2000 example pixels from each image is chosen to ensure a roughly even distribution across body parts. Each tree is trained using the following algorithm [15]:

Randomly propose a set of splitting candidates $\emptyset = (\theta\ ,\tau)$. Partition the set of examples Q = {(I, x)} into left and rightsubsets by each $\emptyset$:

$$Q_I(\emptyset) = \{(I, x) | f_\theta(I, x) < \tau \qquad (2)$$

$$Q_r(\emptyset) = Q \backslash Q_I(\emptyset) \qquad (3)$$

Compute the $\emptyset$ giving the largest gain in information:

$$\emptyset^* = \text{argmax}\ G(\emptyset) \qquad (4)$$

$$G(\emptyset) = H(\emptyset) - \sum_{s\in(I,r)} \frac{|Q_s(\emptyset)|}{Q} H(Q_s(\emptyset)) \qquad (5)$$

Where Shannon entropy H($\emptyset$) is computed on the normalized histogram of body part labels $I_1(x)$ for all (I, x) ∈Q.

If the largest gain G($\emptyset^*$) is sufficient, and the depth in the tree is below a maximum, then recurse for left and right subsets $G_l(\emptyset^*)$ and $G_I(\emptyset^*)$.

The distributions are averaged together for all trees in the forest to give the final classification [19]:

$$P(c|I, x) = \frac{1}{T}\sum_{t=1}^{4} P_t\ (c|I, x) \qquad (6)$$

Our problem formulation is related to Human Pose Recognition, but we approached it to initialize the parameters of AAM algorithm. Because the facial discrimination of depth information is not sufficient, we can't locate facial feature effectively. That's why we need to apply the AAM algorithm to locate facial feature which can use texture and depth information comprehensively. In our experiments, we only care about the upper body, so all the lower body parts could be merged. The four body parts covering the head are merged to localize the head.

$$F(I,x) = \sum_{t=1}^{4} P_t\,(c|I,x) \qquad (7)$$

Where F(I, x) is coordinates of the head which covered by four body parts. What's more, we can also estimate the head pose and its confidence. A final confidence estimate is given as a sum of the pixel weights reaching each mode. The head pose describe (x_{head}, y_{head}, z_{head}) for XYZ coordinates, and its confidence describe $conf_{head}$. We use the head direction to initialize the parameters of AAM when its confidence greater than 0.8.

III. DEPTH AAM ALGORITHM

In general the depth information is very accurate, though a closer look at the face region shows that it is still much noisier than laser scanned results. Qin et al. [20] use iterative closest point algorithm (ICP) to fit the 3D deformable face with the depth image, can track with face images in real-time.

Traditional AAM algorithm only use the three color channels- RGB data as the data input, while facial feature location is inaccurate due to lock of the depth data. In the paper, we propose the depth AAM algorithm to fit both the texture images and the depth images, while both of them come from our Kinect camera. The camera is capable of recording both texture and depth images with 640X480 pixels resolution at 30 frames per second.

Traditional AAM algorithm has many annotated datasets of face images, such as IMM, BIOID and XM2VTS and so on. But there is not an annotated dataset of face images which combined with the depth images, we have to collect a pair of texture images and depth images by ourselves. To facilitate the calculation of training, we combine the RGBD data in a same image (Depth information compress as the α channel). The shape of an AAM algorithm is defined in the same way as ASM [2], which is composed of the coordinates of the ϑ vertices that make up the mesh: from the image.

$$s = (x_1, y_1, x_2, y_2, \dots, x_n, y_n)^T \qquad (8)$$

This means that the shape s can be expressed as a base shape s_0 plus a linear combination of n shape vectors s_i:

$$s = s_0 + \sum_{i=1}^{n} p_i s_i \qquad (9)$$

In this expression the coefficients p_i are the shape parameters, and the vectors s_i are just the orthonormal eigenvectors obtained from the training shapes. We apply Principal Component Analysis (PCA) to the training meshes with the hand labeled training images. The base shape s_0 is the mean shape and the vectors s_i are the n eigenvectors corresponding to the n largest eigen values.

The appearance of an AAM is defined within the basic mesh $s_{o.}$ The basic shape s_o also denotes the set of pixels $X = (x,y)^T$

That lie inside the basic mesh $s_{o.}$ The appearance of an AAM is then an image A(X) defined over the pixels $x \in s_0$. This means the appearance A(X) can be expressed as a base appearance $A_o(x)$ plus a linear combination of m appearance images $A_i(x)$:

$$A(x) = A_0(x) + \sum_{i=1}^{m} c_i A_i(x) \qquad (10)$$

In this expression the coefficients c_i are the appearance parameters. In our experiments, the pixel number within the aligned mean face is 45831 and each pixel takes 4 channel data (RGBD). So there are 31461*4 = 183324 elements in asingle texture vector total.

The N(x;q) is the global shape normalising transform for the AAM training data which is the set of 2D similarity transforms:

$$N(x;q) = \begin{bmatrix} 1+q_1 & -q_2 \\ q_2 & 1+q_1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} q_3 \\ q_4 \end{bmatrix} \qquad (11)$$

where the for parameters $q = (q_1, q_2, q_3, q_4)^T$ have the following interpretations, the first pair $(q_{1,}q_2)$ are related to the scale k and rotation θ: a = k $\cos\theta$-1 and b = k $\sin\theta$. The second pair (q_3,q_4) are the x and y translations.

A global shape transformation N(w(x; p); q) is used to fit the AAM by the inverse compositional algorithm, which warp every vertex in $W(s_0; p)$ with the 2D similarity transform N(x; q). Fitting the AAM to an image I(X) then consists of minimizing:

$$\sum_{x\in s_0}\left[A_0(x)+\sum_{i=1}^{m}c_iA_i(x)-I(N(W(x;p);q))\right]^2 \quad (12)$$

simultaneously with respect to the appearance parameters c, the linear shape parameters p , and the global shape warp
parameters q.

The inverse compositional AAM fitting algorithm [21], [22], [23] includes appearance variation and global shape transform. We compute the modified steepest descent images:

$$\begin{cases} D_j(x) = \nabla A_0\frac{\partial N}{\partial q_j} - \\ \sum_{i=1}^{m}\left[\sum_{x\in s_0}A_i(x).\nabla A_0\frac{\partial N}{\partial q_j}\right]A_i(x) \quad j=1,\dots,4 \\ D_{j+4}(x) = \nabla A_0\frac{\partial N}{\partial p_j} - \\ \sum_{i=1}^{m}\left[\sum_{x\in s_0}A_i(x).\nabla A_0\frac{\partial N}{\partial p_j}\right]A_i(x) \quad j=1,\dots,4 \end{cases} \quad (13)$$

The concatenated steepest descent images form a single vector with four images for q followed by n images for p. The (j, k)th element of the (n + 4) X (n + 4) Hessian matrix:

$$H=\sum_{x\in s_0}[D_k(x)]^T\,[D_k(x)] \qquad (14)$$

Compute (Δq, Δp) by multiplying the resulting vector by the inverse Hessian:

$$\begin{cases} \Delta q = H^{-1}\sum_{i=0}^{n}D_j\left[I\left(N(W(x;p);q)\right)-A_0(x)\right] \\ \quad j=1,\dots,4 \\ \Delta p = H^{-1}\sum_{i=0}^{n}D_{j+4}\left[I\left(N(W(x;p);q)\right)-A_0(x)\right] \\ \quad j=1,\dots,4 \end{cases} \quad (15)$$

Finally, we compute the appearance parameters:

$$c_i=\sum_{x\in s_0}A_i(x).\left[I\left(N(W(x;p);q)\right)-A_0(x)\right] \quad (16)$$

We have described how the inverse compositional image alignment algorithm with appearance variation and global shape transform to be applied to the depth AAM algorithm.

IV. ALGORITHM

In this section we introduce the depth AAM algorithm. The core of our approach is to estimate head pose with the depth information and apply the depth AAM algorithm to locate facial feature which can comprehensive use texture and depth information. We base our approach on the following assumption:

The depth AAM algorithm is applied using the following
algorithm:

1. Get pairs of texture and depth images with Kinect camera, and annotate the texture images with 68 points by hand, then train the appearance models with the depth information using Equations (8) (9) (10)
2. **while** get texture and depth images with Kinect
camera **do**
3. Recognize human pose using Equations (1) ~ (6)
4. **if** human pose recognized, **then** segment facial images
F(I, x) using Equations (7) and estimate head pose
(x_{head}, y_{head}, z_{head}) and its $conf_{head}$

5. **if** head pose $conf_{head}$ is sufficient, **then** initialize the Equations (11), fit the input data (texture and depth images) with the Equations (12) ~ (16)

6. **end if**
7. **end if**
8. **end while**

V. EXPERIMENTAL RESULTS

Because there is not an annotated dataset of face images which combined with the depth images, we can only collect texture and depth images using Kinect camera for our experiment. Our experiments were performed on a Pentium(R) Dual-core CPU E5400 @2.70GHZ and XBOX360 Kinect camera. Our system was programmed with MS Visual

2008 based on opencv2.3.1. We experiment with the image sequences 640*480 from Kinect camera at 30 fps. Figure1 shows the fitted normal image and figure2 shows the fitted curved image.

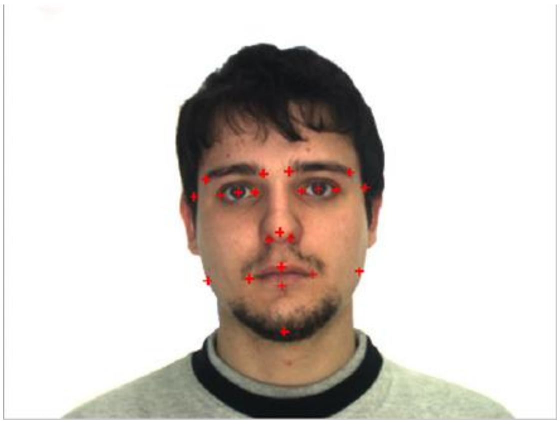

Figure1: Fitting an AAM algorithm to true image

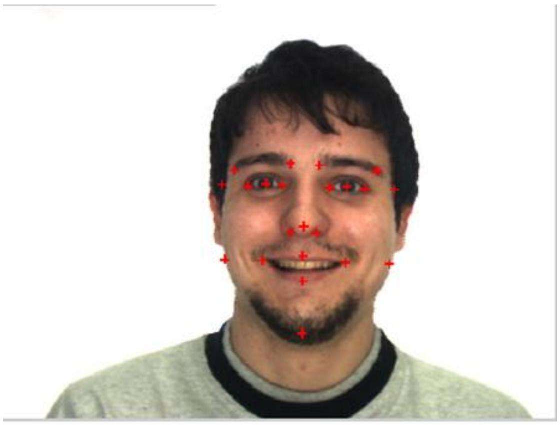

Figure2: Fitting an AAM algorithm to curved image

VI. CONCLUSION AND FUTURE WORK

In this paper, we present a face detection method based on the Kinect camera. It is able to segment facial images and estimate the head pose accurately. Then we introduce the depth AAM algorithm which can be used to locate facial features with both the texture and depth images. The algorithm can use texture and depth information comprehensively and its accuracy and performance is higher than the traditional AAMs. We also show the effectiveness of our approach for real video images. For the future work, we will further improve our method and develop applications in computer vision and human computer interaction.

REFERENCES

[1] Y. Wu and X. Ai, "Face Detection in Color Images Using AdaBoost Algorithm Based on Skin Color Information," in First International Workshop on Knowledge Discovery and Data Mining, 2008, pp. 339- 342.

[2] S. Milborrow and F. Nicolls, "Locating Facial Features with an Extended Active Shape Model," in Computer Vision – ECCV 2008, vol. 5305, pp. 504-513.

[3] I. Matthews and S. Baker, "Active Appearance Models Revisited," International Journal of Computer Vision, vol. 60, no. 2, pp. 135-164, Nov. 2004.

[4] A. Caunce, D. Cristinacce, C. Taylor, and T. Cootes, "Locating Facial Features and Pose Estimation Using a 3D Shape Model," in Advances in Visual Computing, 2009, vol. 5875, pp. 750-761.

[5] W. Zhang, Q. Wang, and X. Tang, "Real Time Feature Based 3-D Deformable Face Tracking," in Proceedings of the 10th European Conference on Computer Vision: Part II, Berlin, Heidelberg, 2008, pp. 720–732.

[6] X. Lu and A. Jain, "Deformation Modeling for Robust 3D Face Matching," in 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, vol. 2, pp. 1377- 1383. 1796

[7] Y. Zhu and K. Fujimura, "3D head pose estimation with optical flow and depth constraints," in Fourth International Conference on 3-D Digital Imaging and Modeling (3DIM), 2003, pp. 211- 216.

[8] V. Frati and D. Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics," in 2011 IEEE World Haptics Conference (WHC), 2011, pp. 317-321.

[9] T. F. Cootes and C. J. Taylor, "Constrained active appearance models," in Eighth IEEE International Conference on Computer Vision (ICCV), 2001, vol. 1, pp. 748-754.

[10] R. Navarathna, S. Sridharan, and S. Lucey, "Fourier Active Appearance Models," in 2011 IEEE International Conference on Computer Vision (ICCV), 2011, pp. 1919-1926.

[11] M. C. Ionita, P. Corcoran, and V. Buzuloiu, “On Color Texture Normalization for Active Appearance Models,” IEEE Transactions on Image Processing, 2009, vol. 18, no. 6, pp. 1372-1378.

[12] J. Xiao, S. Baker, I. Matthews, and T. Kanade, “Real-time combined 2D+3D active appearance models,” in Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, vol. 2, pp. 535-542.

[13] P. Viola and M. Jones, “Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade,” Advances in Neural Information Processing System, vol. 14, p. 1311--1318, 2001.

[14] E. Chutorian and M. Trivedi, “Head Pose Estimation in Computer Vision: A Survey,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 31, no. 4, pp. 607-626, Apr. 2009.

[15] V. Lepetit, P. Lagger, and P. Fua, “Randomized trees for real-time keypoint recognition,” in IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, vol. 2, pp. 775- 781.

[16] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, “Real time motion capture using a single time-of-flight camera,” in 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2010, pp. 755-762.

[17] C. Plagemann, V. Ganapathi, D. Koller, and S. Thrun, “Real-time identification and localization of body parts from depth images,” in 2010 IEEE International Conference on Robotics and Automation (ICRA), 2010, pp. 3108-3113.

[18] L. Bourdev and J. Malik, “Poselets: Body part detectors trained using 3D human pose annotations,” in 2009 IEEE 12th International Conference on Computer Vision, 2009, pp. 1365-1372

[19] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake, “Real-time human pose recognition in parts from single depth images,” in 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011, pp. 1297-1304.

[20] Q. Cai, A. Sankaranarayanan, Q. Zhang, Z. Zhang and Z. Liu, “Real time head pose tracking from multiple cameras with a generic model,” in 2010 IEEE Computer Society

Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2010, pp. 25-32.

[21] G. Papandreou and P. Maragos, "Adaptive and constrained algorithms for inverse compositional Active Appearance Model fitting," in IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1-8.

[22] J. Saragih and R. Goecke, "A Nonlinear Discriminative Approach to AAM Fitting," in IEEE 11th International Conference on Computer
Vision, 2007, pp. 1-8.

[23] S. Koterba, S. Baker, I. Matthews, Changbo Hu, Jing Xiao, J. Cohn and T. Kanade, "Multi-view AAM fitting and camera calibration," in Tenth IEEE International Conference on Computer Vision, 2005, vol. 1, pp. 511- 518.