

AAM Based Facial Feature Tracking with Kinect

Qingxiang Wang¹, Yanhong Yu²

¹*School of Information, Qilu University of Technology, China*

²*Basis Medical College, Shandong University of Traditional Chinese Medicine, China*

Email: wangqx@qlu.edu.cn

Abstract: Facial features tracking is widely used in face recognition, gesture, expression analysis, etc. AAM (Active Appearance Model) is one of the powerful methods for objects feature localization. Nevertheless, AAM still suffers from a few drawbacks, such as the view angle change problem. We present a method to solve it by using the depth data acquired from Kinect. We use the depth data to get the head pose information and RGB data to match the AAM result. We establish an approximate facial 3D gird model and then initialize the subsequent frames with this model and head pose information. To avoid the local extremum, we divide the model into several parts by the poses and match the facial features with the closest model. The experimental results show improvement of AAM performance when rotating the head.

Keywords: Facial feature tracking, active appearance model, view based model, Kinect.

1. Introduction

Tracking of facial features, which is widely used in face recognition, gesture, expression analysis, expression transfer and human-computer interaction, is a very active research direction in computer vision. However, because of the complexity of the facial expression, locating and tracking of the facial features is still a hot spot in this field. One of the mainly used methods of tracking facial features is AAM (Active Appearance Model), proposed by Coots, Edwards and Taylor [1]. It is an algorithm for matching the statistical model of the object shape and appearance to a new image that is widely used in face feature localization, identification [2], video tracking [3], attitude estimation [4].

After its design, many improvements have been proposed. Most of the recent improvements of AAM are in three aspects: efficiency, discrimination and robustness [5]. But it still suffers from the view angle change problem. Most of the work is based on the assumption that the head is close to the near frontal parallel views. However, when the head rotates, the linear relationship of the facial features could not be kept in cases of large rotations. This problem can be settled by introducing the pose information into the model

Some researches are proposed referring to this issue. The pose information which they use includes 2D and 2D+3D. C o t s et al. [6] proposed a 2D ellipse approximate method. They create five models in specific views and assume that the model parameters are related to viewing the angle θ . After that θ and the relationship between the models parameters is derived and used for forecast and model selection of perspective. W a n g et al. [7] puts forward a similar method with more feature points to improve the robustness. They need to predicate the pose from 2D texture by assumption of the approximation of the ellipse too. X i a o et al. [8] have proposed a 2D+3D method which increases 3D constraints on the basis of 2D textures, L i e b e l t, X i a o and Y a n g [9] extends this to multi-cameras which need continuous data or a stereo camera to construct the 3D model on the pose based model. G o n z a l e z - M o r a et al. [10] proposed bilinear AAM which decoupled the appearance into the pose subspace and the identity subspace. Although great progress has been made, the facial features tracking with AAM under conditions of complex backgrounds, illumination changes and pose changes is still difficult if only relying on a RGB image.

In recent years, more and more studies tend to use the 3D model and depth information. Kinect can provide depth information and RGB image information at the same time, which provides a new tool for facial feature extraction in the video. With the depth data the head pose could be easily acquired, like F a n e l l i et al. [11], that makes it possible to match the facial feather positions more accurately.

With the depth camera, C a i et al. [12] proposed a deformable model fitting method which is based on ICP (Iterative Closest Point). They found the corresponding relations between the depth and the deformable model and then used them to initialize and solve the deforming coefficient to locate the facial features. T h i b a u t et al. [13] constructed the expressions base and personalized the specific person's expression. They got the expression coefficient from the input depth and then transferred the coefficient to a new face. B a l t r u s a i t i s, R o b i n s o n and M o r e n c y [14] used the RGB and depth partial template to solve the problem of different illumination. Y a n g et al. [15] and B e l l m o r e [16] improved the ASM (Active Shape Model) algorithm with RGB and depth image.

The precision and robustness is the key to the study of facial feature matching. The matching of facial features is susceptible to illumination and pose that causes a lot of matching errors. However, due to the use of infrared, 3D depth data extracted from Kinect, it is less affected by illumination and the introduced depth information could enhance the accuracy of the pose decision. Due to the additional depth data and since 2D images are from dual cameras, angle difference exists between the cameras, the pixels are not in one-to-one correspondence that is needed to be solved

in order to obtain the depth of each pixel value. At the same time, the quality of the initialization has a large effect on the active appearance model. Thus, how to initialize the matching model under the view change conditions after head pose extracting is one of our research questions.

In this paper we have two tasks: 1) with the additional depth image, the method should connect the pixels between the depth and RGB images and establish a facial 3D grid model and view based AAM models; 2) when the head is rotated, the AAM should be initialized properly.

The overview of the paper is shown in Fig. 1. The work includes two parts, preparation and run-time, the former is off-line and consists of view based models training and head pose regression forests training, the latter is the match process which is online.

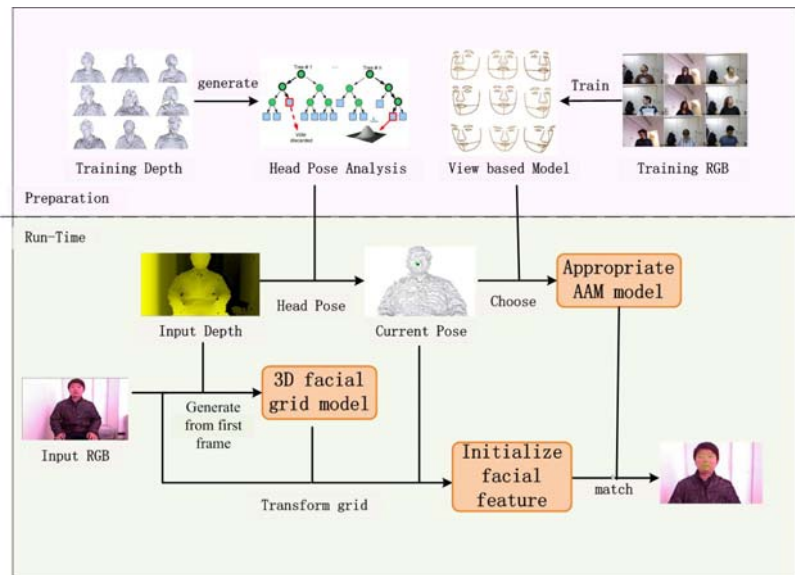


Fig. 1. Overview of the work done

The paper is organized as follows. In Section 2 we introduce the AAM. In Section 3 the preparation of the experiment is given, including the view based model and head pose analysis. In Section 4 the run-time process is presented, including the initialization of the 3D facial grid and fitting process. The experimental results obtained in this paper are presented in Section 5. Finally, the work of the paper is summarized in the last section.

2. Active appearance model

The AAM [1] uses statistical models and input images appearance to estimate the shape of the face. It divides the model into two parts, shape and appearance, and assumes that both of them can be obtained by combination of the linear space.

The shape of an AAM is defined by a 2D triangulated mesh. The shape S is defined as the x and y -coordinates of the n vertices that make up the mesh: $S = (x_1, y_1, \dots, x_n, y_n)^T$. A compact linear shape model is given by

$$(1) \quad S = S_0 + \sum_{i=1}^m p_i S_i,$$

where p_i are the shape parameters and S_0 is the mean shape and the vectors S_i are the m largest eigenvalues.

The appearance is

$$(2) \quad A = A_0(x) + \sum_{i=1}^l q_i A_i(x),$$

where q_i are the appearance parameters and A_0 is the mean appearance and the vectors A_i are the l largest eigenvalues.

After applying PCA (Principal Components Analysis) to the shapes and appearance of the training images, S_i and A_i can be acquired. AAM matches a shape to the face minimizing:

$$(3) \quad \arg \min \left(\sum_{x \in S_0} \left[A_0(x) + \sum_{i=1}^m q_i A_i(x) - I(N(W(x; p); u)) \right]^2 \right).$$

For convenience of the description, we define the combination parameter $C = \{p, q, u\}$, including the parameters in (3), the shape parameters p , the appearance parameters q and the translation, rotation and scaling parameters u . AAM algorithm calculates the residual error of the actual appearance and the predicted appearance, and then optimizes the combination coefficient C by iterative refinement to minimize the error. At the end the coefficient is used to get the shape of the facial features. More details can be seen in [1].

Every shape in the train images is labelled and composed of a fixed number of points. In this paper we use the definition of XM2VTS14 [17] frontal data, 68 points.

3. Preparation-view based models and head pose estimate model

3.1. View based models

Labeled face images are needed for the training process of AAM.

We use the database of Biwi Kinect Head Pose Database (http://www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html#db). The data-base contains over 15K images with RGB and depth data captured from Kinect in 24 image sequences. Every sequence includes several hundred frames and the exterior parameters of RGB and depth.

Most of them are similar to the adjacent ones, so we only manually label about 200 images. We use the approximate normal face plane to express the view. In order to simplify the expression, we make the approximate face normal correspondence to a point of the gauss sphere, thus every normal can be expressed by the longitude and

latitude of the point of gauss sphere, as {longitude, latitude}. Then we divide the images into 9 views and train 9 view based models. The 9 views are $\{0, 0\}$, $\{0, 30\}$, $\{0, -30\}$, $\{30, 0\}$, $\{-30, 0\}$, $\{30, 30\}$, $\{-30, 30\}$, $\{30, -30\}$, $\{-30, -30\}$, as shown in Fig. 2. The images which are close to two parts, like $\{15, 15\}$, will be trained in both models.

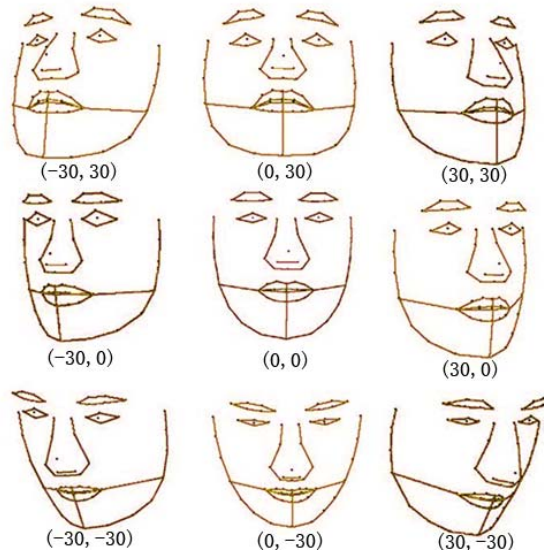


Fig. 2. View based models

Each part is trained as a separate AAM model for an angle range of the head pose. These models are defined as view based models.

3.2. Head pose estimate model

In this paper we use the head pose estimate method, proposed by Fanelli et al. [11], to estimate the angle and head center of the head pose from Kinect depth data based on discriminative random regression forests. They establish random trees trained by splitting each node to simultaneously reduce the Shannon entropy of the class labels distribution and the variance of the head position and orientation.

4. Run time – initialization and match based on facial 3D grid model

In this section we will introduce how to mark the facial features of the input image sequence. The process includes two parts, initialization and match. First we establish a facial 3D grid of the face of the input frame and then transform it with the head pose and initialize the next frame with projection of the grid.

4.1. Facial 3D grid model

Kinect has two cameras, one is RGB, and the other is depth. We define the intrinsic parameters of the RGB as A_{RGB} and the depth as A_d . Each camera has its unique camera coordinate system and image coordinate system.

The establishment of the facial 3D grid model is to extract the 3D coordinates of the facial features on the RGB camera coordinate system. At first, we extract the facial features on the RGB image by AAM, then calculate the 3D coordinates of these features by projecting the depth to the RGB and averaging the depth values in a small neighborhood region.

In the first frame which contains a face, we use Viola-Jones face detection [18] to get the face region and initialize the AAM to get the facial features by an original AAM match process. The face must be with no expression and the front view is better than the other views. If the first frame is not the front view, we should use the head pose estimation result to choose the approximate view based model to extract the facial features. The extracted facial features by AAM are on RGB image coordinate system, we should calculate the coordinates in a RGB camera coordinate system.

Assuming that a feature point's coordinate is defined as P_{RGB} in RGB camera coordinate system, P_d in depth camera coordinate system, P_{RGB}' in RGB image coordinate system and P_d' in depth image coordinate system, as shown in Fig. 3. O_d is the origin of the depth camera coordinate system and O_{RGB} is the origin of the RGB camera coordinate system.

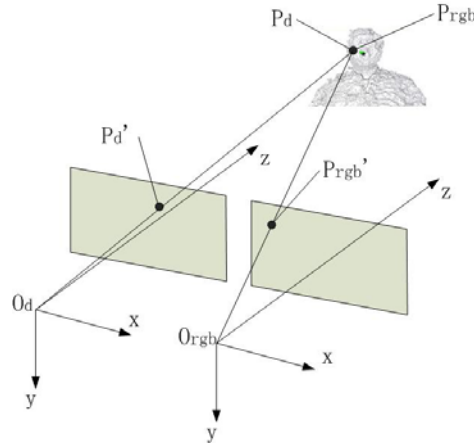


Fig. 3. RGB and depth camera coordinate system

The problem is how to get the P_{RGB} with known P_{RGB}' with the help of an intermediate coordinate P_d . The method is as follows:

$$(4) \quad P_{RGB} = R \cdot P_d + T,$$

where R and T are the rotation and translation of the exterior parameter;

$$(5) \quad P_{RGB}(Z) = r_{31} \cdot P_d(X) + r_{32} \cdot P_d(Y) + r_{33} \cdot P_d(Z) + t_3,$$

where $\{r_{31}, r_{32}, r_{33}\}$ is the third row of R and $\{t_3\}$ is the third row of T . In this equation the values of r_{31} and r_{32} are below 0.01, so we approximately abandon the first two terms in the calculation. We need to know the $P_d(Z)$ corresponding to $P_{RGB}(Z)$. But the pixels of depth and RGB are not in one to one correspondence. Some pixels in depth may correspond to one pixel or none in the RGB coordinate

system. We could not find the in depth camera coordinate through the RGB image coordinate directly. We present a method to solve this problem. At first we projected every pixel of the facial region in a depth image to RGB image and then weighted the average of these points around the facial feature P_{RGB} in a small neighbourhood region.

Having the $P_{\text{RGB}}(Z)$, we could get the $P_{\text{RGB}}(X)$ and $P_{\text{RGB}}(Y)$ by

$$(6) \quad P_{\text{RGB}}' = A_{\text{RGB}} \cdot P_{\text{RGB}},$$

where A_{RGB} are the intrinsic parameters.

Finally we could get the RGB camera coordinate of every facial feature and create a facial grid, then move the grid center to the head center's projection position in a RGB sensor camera. After the above steps, we could get a facial 3D grid model.

4.2. Initialization

The initialization has two steps:

1. Choose the view based model

When the next frame is coming, firstly we use the head pose estimate method to get the rotated matrix R and the head center T . Then we use the angle to choose the approximated view based AAM model.

2. Initialize the iterative initial value of AAM

After the first step, we use the rotation matrix R and the head center T to transform the facial 3D grid. Then we project the grid to a RGB image plane with the intrinsic parameters. At last we use the projected result as the initialization of AAM. The transform and projection are the following:

$$(7) \quad \begin{pmatrix} u \\ v \\ 1 \end{pmatrix} = \begin{pmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix},$$

where $(X, Y, Z)^T$ is one vertex of the facial 3D grid, $(u, v)^T$ is the projection on the image plane of $(X, Y, Z)^T$. The first matrix on the right side presents the intrinsic parameters of RGB camera, f_x and f_y are the focal lengths expressed in pixel units, (c_x, c_y) is a principal point that is usually at the image center, the second matrix is the rotation-translation matrix $R|T$.

The initialization is shown in Fig. 4, in which the red dots are the initialization facial feature of AAM.

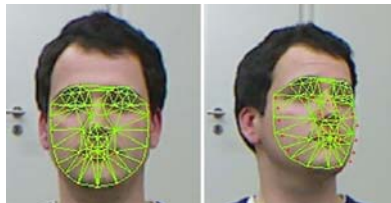


Fig. 4. The red dots are the points on the 3D model that will be used to initialize the AAM

4.3. Match

After initialization, we use the selected view based AAM model to match the result. The maximal iterations in this algorithm are set to 10. The algorithm is shown below.

Algorithm 1. Initialization and match

- Step 1.** for each frame f in the video sequence
- Step 2.** if the first frame and contains face
- Step 3.** establish the facial 3D grid model
- Step 4.** else
- Step 5.** get the head pose (R|T) with regression forests
- Step 6.** rotate and translate the facial 3D grid and project to RGB camera plane $(X, Y, Z)^T \rightarrow (u, v)^T$
- Step 7.** choose the view based AAM model
- Step 8.** initialization with the projection
- Step 9.** use AAM to calculate the shape coefficient C (see Section 2)
- Step 10.** output the shape coefficient C

5. Experimental results

5.1. Experimental data

The data used in this paper is a database named Biwi Kinect Head Pose Database which is published by G. Fanelli on his website, acquired from Kinect and it contains over 15K images of 20 people (6 females and 14 males; 4 people were recorded twice). For each frame in the database, a depth image and the corresponding RGB image (both 640×480 pixels) is provided.

For feature point training, we use Biwi Kinect Head Pose Database and manually label about 300 pictures by the `am_tools_win_v3` downloaded from Coots's web site of Manchester. Each picture is labelled by 68 points on the face. All the experiments are implemented in C++ and OpenCV2.3 on a PC with Intel (R) CPU, 2.33GHz, 2G Memory. We tested our method on four sequences from the database.

We divide the pictures by the head pose angles into 9 view based parts for training AAM models (the PCA precision is set to 95% to get the m and l largest eigenvalues in (1) and (2)) and use the method of Fanelli et al. [11] for head pose estimation with the depth data to choose the current appropriate AAM model.

Our method is compared with the original AAM [1]. For an original AAM, all the pictures with different views are trained as one model.

5.2. Result

We test our method on Biwi Kinect database, the partial result is shown in Fig. 5. In the figure, the faces in a row are selected from one video sequence of the database and named with sequences 1, 2, 3 from top to bottom. The running time of each sequence is shown in Table 1.



Fig. 5. Partial results of the test on Biwi Kinect database

Table 1. Running time

Sequence	Establish facial 3D grid	Head pose	Initialization	Match	Total
	ms	ms per frame	ms per frame	ms per frame	ms per frame
1	540.107	82.157	0.022	16.830	99.009
2	577.923	70.529	0.022	17.133	87.684
3	597.563	66.386	0.023	15.564	81.973

In the table the second column shows the time for establishing the facial 3D grid which is done once on the first frame. This is about 0.5 s. The third up to the fifth columns give the time for getting the head pose information (Rotation and Translation), initializing AAM and choosing the view based model and matching the result, those totally consume about 81-99 ms.

The partial continuous frames are shown in Fig. 6 (our method) and Fig. 7 (the original literature method), arranged from left to right and from top to bottom, from No 364 frame to No 424 frame for a four frames interval.



Fig. 6. Our method. sequential frames (from 364 to 424 steps 4)



Fig. 7. Original literature method. Sequential frames (from 364 to 424 steps 4)

From Fig.7 we can see that the facial features in the frames of Nos 364, 384, 420 are mismatched with the original literature method. This may be caused by the bad initialization which causes the iterations not to converge. In the frames of Nos 380, 388, 412, a small number of facial features are mismatched. This may be caused by overfitting of the all-view model. In these frames, our method can match correctly.

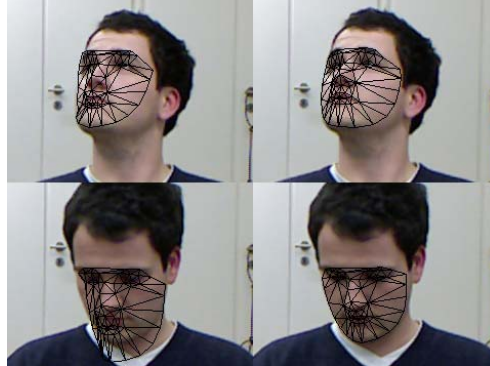


Fig. 8. Without and with a view based model

We can see more details in Fig. 8. Each image pair of the figure has two images, the left is the original literature method and the right is the proposed herein. It is obvious that our method is better than the old one.

For error measurement, we use a method similar to *Cristinacce* and *Cootes* [19]. But since there are view changes and eyes closed in the video sequence, we use the distance of the two inner eye corners as normalization instead of the distance of a pupil:

$$(8) \quad e = \frac{1}{N \cdot D} \sum_{i=1}^N e_i,$$

where N is the number of the facial features, e_i is the RMS (Root Mean Square) error between the extracted facial features and the ground truth, D is the distance of the inner eye corners.

The RMS error is shown in Fig. 9 and the rate of convergence (the iterative formula in Equation (3) is convergent), as shown in Fig. 10. In Fig. 9 the sample proportion is the distribution of RMS error (the accuracy is 0.01, about 0.2-0.25 pixels, relative to the distance of the two inner eye corners). From the figure we can see that the convergence rate and RMS error of our method is better than the original literature method.

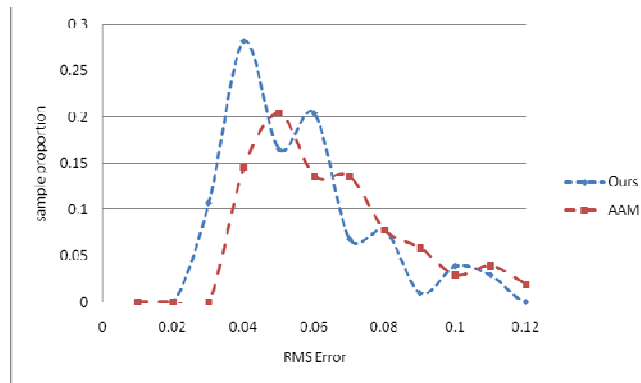


Fig. 9. The distribution of RMS error of the method proposed and AAM

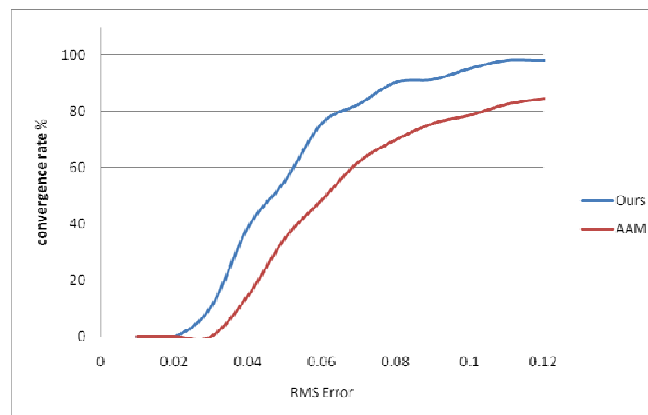


Fig. 10. Convergence rate (the RMS error is relative to the distance of the two inner eye corners)

6. Conclusions

The main contributions of this paper are as follows. (1) A view based model is created, choosing a method based on the view based AAM models which are trained by sampling different perspective images. When matching we could choose the appropriate AAM model with the current view information getting it from the depth. (2) A facial 3D grid model is suggested, establishing a method based on the facial feature points. When matching, we transform the grid with the head pose

information, project it to the imaging plane and initialize the AAM. This can improve the accuracy and robustness of the facial feature tracking.

We present a method for view based facial feature tracking under the depth and RGB data captured from Kinect. To solve the initialization and head motion problem of the active appearance models, we propose a convenient method for initialization and model choosing of a view based AAM under depth data from a video sequence, which enhances the accuracy of the algorithm.

We propose a method, which uses the depth and RGB image, to extend the active appearance models. The method is based on the view based AAM models which are trained by sampling different perspective images. The view is divided into 9 parts. We propose a facial 3D grid model establishing method based on facial feature points and depth data at the first frame, then initialize the AAM and choose an appropriate model at different views through this grid that improved the accuracy and robustness of AAM.

The experimental results show the improvement of AAM under head rotation. But because of the lack of data, our experiment does not match the pose out of $\{-45, 45\}$. Another shortcoming is that the depth information (normal, edge) would be more useful in the matching process than we have achieved. This could be a future work on the method.

Acknowledgment: This work was supported by National Natural Science Foundation of China (Grant No 81102738, No 81573829), Jinan Star Science and Technology Plan (No 201406004, No 20120104) and Institute Support Innovation Project of Jinan Administration of Science & Technology (No 201202012).

We should thank G. Fanelli to supply their database and head pose estimate code on their website.

References

1. Coates, T. F., G. J. Edwards, C. J. Taylor. Active Appearance Models. – IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **23**, 2001, No 6, pp. 484-498.
2. Liang, Lin, Rong Xiao, Fang Wen, Jian Sun. Face Alignment Via Component Based Discriminative Search. – In: Proc. of 10th European Conference on Computer Vision: Part II (Berlin, Heidelberg), 2008, pp. 72-85.
3. Matthews, I., J. Xiao, S. Baker. 2d Vs. 3d Deformable Face Models: Representational Power, Construction, and Real-Time Fitting. – International Journal of Computer Vision, Vol. **75**, 2007, No 1, pp. 93-113.
4. Murphy-Chutorian, E., M. M. Trivedi. Head Pose Estimation in Computer Vision: A Survey. –IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. **31**, 2009, No 4, pp. 607-626.
5. Gao, X., Y. Su, X. Li, D. Tao. A Review of Active Appearance Models. – IEEE Transactions on Systems, Man, and Cybernetics. Part C. Applications and Reviews, Vol. **40**, 2010, No 2, pp. 145-158.
6. Coates, T. F., G. V. Wheeler, K. N. Walker, C. J. Taylor. View-Based Active Appearance Models. – Image and Vision Computing, Vol. **20**, 2002, No 9-10, pp. 657-664.
7. Wang, X., X. Feng, M. Zhou, Y. Wang. Real-Time View-Based Face Alignment Using an Extended Aam. – In: Proc. of International Conference on Intelligent Computing and Intelligent Systems, 2009, pp. 557-561.

8. Xiao, J., S. Baker, I. Matthews, T. Kanade. Real-Time Combined 2d+3d Active Appearance Models. – In: Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004, pp. 535-542.
9. Liebelt, J., J. Xiao, J. Yang. Robust Aam Fitting by Fusion of Images and Disparity Data. – In: Proc. of IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2006, pp. 2483-2490.
10. Gonzalez-Mora, J., F. De la Torre, R. Murthi, N. Guil, E. L. Zapata. Bilinear Active Appearance Models.– In: Proc. of 11th International Conference on Computer Vision, 2007, pp. 1-8.
11. Fanelli, G., T. Weise, J. Gall, L. J. Van Gool. Real Time Head Pose Estimation from Consumer Depth Cameras. – Lecture Notes in Computer Science, Vol. **6835**, 2011, pp. 101-110.
12. Cai, Q., D. Gallup, C. Zhang et al. 3D Deformable Face Tracking with A Commodity Depth Camera. – In: Proc. of 11th European Conference on Computer Vision, 2010, pp. 229-242.
13. Thibaut, W., B. Sofien, Li Hao et al. Realtime Performance-Based Facial Animation. – In: Proc. of ACM SIGGRAPH, 2011.
14. Baltrusaitis, T., P. Robinson, L. Morency. 3D Constrained Local Model for Rigid and Non-Rigid Facial Tracking. –In: Proc. of Computer Vision and Pattern Recognition, 2012, pp. 2610-2617.
15. Yang, F., J. Huang, X. Yu et al. Robust Face Tracking with a Consumer Depth Camera [C]. – In: Proc. of 19th IEEE International Conference on Image Processing, 2012, pp. 561-564.
16. Bellmore, C. Facial Feature Point Fitting with Combined Colour and Depth Information for Interactive Displays. Kate Gleason College of Engineering, 2012.
17. Messer, K., J. Matas, J. Kittler, K. Jonsson. Xm2vtsdb: The Extended M2vts Database. – In: Proc. of 2nd International Conference on Audio and Video-Based Biometric Person Authentication, 1999, pp. 72-77.
18. Viola, P., M. J. Jones. Robust Real-Time Face Detection. – International Journal of Computer Vision, Vol. **57**, 2004, No 2, pp. 137-154.
19. Cristinacce, D., T. F. Cootes. Facial Feature Detection and Tracking with Automatic Template Selection. – In: Proc. of 7th International Conference on Automatic Face and Gesture Recognition, 2006, pp. 429-434.