Facial model fitting algorithm based on active appearance model

LI Lu-ning, Hernsoo Hahn, Youngjoon Han

(*Dept. of Engineering and Electronics, Soongsil University, Seoul 156-743, Korea*)

Abstract: Active appearance model(AAM) is an efficient method for the localization of facial feature points, which is also useful for the subsequent work such as face detection and facial expression recognition. In this paper, we mainly discuss the AAMs based on principal component analysis (PCA). We also propose an efficient facial fitting algorithm, which is named inverse compositional image alignment (ICIA), to eliminate a considerable amount of computation resulting from traditional gradient descent fitting algorithm. Finally, 3D facial curvature is used to initialize the location of facial feature, which helps select the parameters of initial state for the improved AAM.

Key words: active appearance model(AAM); inverse compositional image alignment (ICIA); facial fitting; facial curvature

CLD number: TP391.41 **Document code**: A

Article ID: 1674-8042(2012)04-0323-05 **doi**: 10.3969/j.issn.1674-8042.2012.04.005

There is currently a great deal of interest in model-based approaches to the interpretation of human faces. The attractions are two-fold: robust interpretation is achieved by constraining solutions to be valid instances of the model example; and the ability to "explain" the face in terms of a set of model parameters provides a basis for face interpretation. In order to realize these benefits, the model of object appearance should be as complete as possible. Although model-based methods have proven quite successful, there are few existing methods that use a full, photo-realistic model and attempt to match it directly by minimizing the difference between the model-synthesized face and the image under interpretation. They typically involve a very large number of parameters (50 − 100) in order to represent the human face. Subsequently, active appearance model (AAM) was proposed by Cootes et al.[1], which is an efficient iterative matching process for the location of facial feature points.

Actually, fitting an AAM to a face is a non-linear optimization problem. A usual approach is to iteratively update the parameters (the shape and appearance coefficients)[1-2]. Given the current estimates of the shape parameters, it is possible to warp the input image backwards onto the model coordinate frame and then compute an error image between the current model instance and the image that the AAM is being fit to. In most previous algorithms, it is simply assumed that there is a constant linear relationship between this error image and the additive incremental updates to the parameters. Unfortunately, the assumption that there is such a simple relationship between the error image and the appropriate update to the model parameters is generally incorrect. The result is that existing AAM fitting algorithms perform poorly, both in terms of the number of iterations required to converge, and in terms of the accuracy of the final fit.

In this paper we propose an efficient AAM fitting algorithm named inverse compositional image alignment (ICIA), which is based on Lucas-Kanade image alignment[3]. The ICIA method does not consider updates of the shape parameters. Instead, it updates the entire warp by composing the current warp with the computed incremental warp. With the ICIA algorithm, we can project out the appearance variation thereby eliminating a great deal of computation.

Besides, the standard AAM combined with principal component analysis (PCA), which is the basis of ICIA fitting algorithm, is also presented. Finally, the initial pose parameter is given out by key points of the facial feature using curvature of 3D facial surface. Because the initial pose parameter is an important factor which affects the accuracy of AAMs. The paper uses the curvature of 3D facial surface to complete preliminary location of the facial features,

※ **Received data**: 2012-08-02

Foundation item: The MKE(The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA (National IT Industry Promotion Agency) (NIPA-2012-H0301- 12-2006); The Brain Korea 21 Project in 2012.

Corresponding author: LI Lu-ning (runyonzeelee@163.com)

and takes the location result as the initial pose parameter of the improved AAMs matching.

1 AAMs

AAM is a statistics based template matching method, where the variability of shape and texture is captured from a representative training set. PCA on shape and texture data allows building a parameterized face model that fully describes with the photo-realistic trained faces as well as the unseen. Further details can be seen in Ref. [2].

1.1 Shape model

The shape is defined as the quality of a configuration of points which is invariant under Euclidian similarity transformations[4]. The representation used for a single n-point shape is a $2n$ vector given by

$$s = (x_1, y_1, x_2, y_2, \cdots, x_{n-1}, y_{n-1}, x_n, y_n)^{\mathrm{T}}$$

with n shape annotations, follows a statistical analysis where the shapes are previously aligned to a common mean shape using a generalized procrustes analysis (GPA) removing location, scale and rotation effects. After PCA, we can model the statistical variation with

$$\boldsymbol{s} = \boldsymbol{s}_0 + \boldsymbol{\Phi}_s \boldsymbol{b}_s, \tag{1}$$

where $\boldsymbol{s}_0$ is the mean shape, $\boldsymbol{\Phi}_s$ is a weighted linear combination of eigenvectors of the covariance matrix, $\boldsymbol{b}_s$ is a vector of shape parameters which represents the weights. We can change the form of Eq. (1) as

$$\boldsymbol{s} = \boldsymbol{s}_0 + \sum_{i=1}^{n} \boldsymbol{p}_i \boldsymbol{s}_i, \tag{2}$$

where the coefficients p_i are the shape parameters. Since we can easily perform a linear reparameterization, wherever necessary, we assume that the vectors $\boldsymbol{s}_i$ are orthonormal.

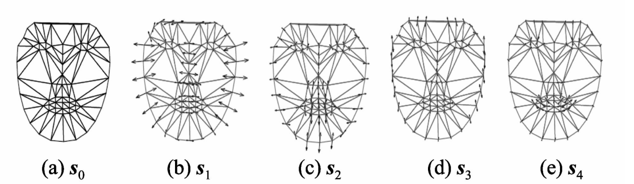


Fig. 1 Linear shape model of an independent AAM

An example of independent AAM shape model is shown in Fig. 1. On the left of the figure, we plot the triangulated base mesh $\boldsymbol{s}_0$. In the remainder of the figure, the base mesh $\boldsymbol{s}_0$ is overlaid with arrows corresponding to each of the first four shape vectors $\boldsymbol{s}_1$, $\boldsymbol{s}_2$, $\boldsymbol{s}_3$ and $\boldsymbol{s}_4$.

1.2 Texture model

For pixels sampled, the texture is represented by the vector. Building a statistical texture model requires warping each training image so that the control points match those of the mean shape. As a result, we partition the convex hull of the mean shape by a set of triangles using the Delaunay triangulation. Each pixel inside a triangle is mapped into the correspondent triangle in the mean shape, as shown in Fig. 2.

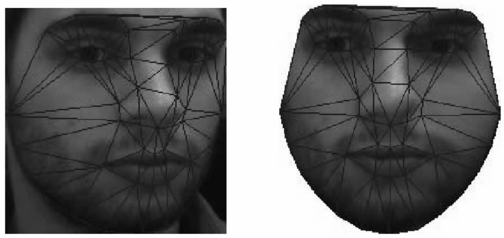


Fig. 2 Texture mapping example

A texture model can be obtained by means of a PCA on the normalized textures

$$\boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\Phi}_g \boldsymbol{b}_g, \tag{3}$$

where $\boldsymbol{g}$ is the synthesized texture, $\boldsymbol{g}_0$ is the man texture, $\boldsymbol{\Phi}_g$ contains highest covariance texture eigenvectors and $\boldsymbol{b}_g$ is a vector of texture parameters. We can change the form of Eq. (3) as

$$\boldsymbol{A}(x) = \boldsymbol{A}_0(x) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(x), \tag{4}$$

where the coefficients λ_i are the appearance parameters. Since we can easily perform a linear reparameterization, wherever necessary, we assume that the images $\boldsymbol{A}_i$ are orthonormal.

1.3 Combined model

The shape and texture from any training example is described by the parameters $\boldsymbol{b}_s$ and $\boldsymbol{b}_g$. To remove correlations between shape and texture model parameters, a third PCA is performed to the following data

$$\boldsymbol{b} = \begin{pmatrix} w_s \boldsymbol{b}_s \\ \boldsymbol{b}_g \end{pmatrix} = \begin{pmatrix} \boldsymbol{W}_s \boldsymbol{\Phi}_s^{\mathrm{T}}(\boldsymbol{s} - \boldsymbol{s}_0) \\ \boldsymbol{\Phi}_g^{\mathrm{T}}(\boldsymbol{g} - \boldsymbol{g}_0) \end{pmatrix}, \tag{5}$$

where $\boldsymbol{W}_s$ is a diagonal matrix of weights for shape and texture parameters with the ratio $r = \sum_i \lambda_{gi} / \sum_i \lambda_{si}$, where $\boldsymbol{\lambda}_s$ and $\boldsymbol{\lambda}_g$ are shape and texture eigenvalues, respectively. Using PCA again, $\boldsymbol{\Phi}_c$

holds the t_c highest eigenvectors, and we obtain the combined model, $\boldsymbol{b} = \boldsymbol{\Phi}_c \boldsymbol{c}$. Thus, we can express the shape and texture using the combined model by

$$\boldsymbol{s} = \boldsymbol{s}_0 + \boldsymbol{\Phi}_s \boldsymbol{W}_s^{-1} \boldsymbol{\Phi}_{c,s} \boldsymbol{c}, \tag{6}$$

$$\boldsymbol{g} = \boldsymbol{g}_0 + \boldsymbol{\Phi}_g \boldsymbol{\Phi}_{cg} \boldsymbol{c}, \tag{7}$$

where $\boldsymbol{\Phi}_c = \begin{pmatrix} \boldsymbol{\Phi}_{cs} \\ \boldsymbol{\Phi}_{cg} \end{pmatrix}$ and $\boldsymbol{c}$ is a vector of appearance controlling both shape and texture.

1.4 Model instantiation

Eqs. (2) and (4) describe the AAM shape and appearance variation. Given the AAM shape parameters

$$\boldsymbol{p} = (p_1, p_2, p_3, \cdots, p_n)^{\mathrm{T}},$$

we can use Eq. (2) to generate the shape of the AAMs. Similarly, given the AAM appearance parameters

$$\boldsymbol{\lambda} = (\lambda_1, \lambda_2, \cdots, \lambda_m)^{\mathrm{T}},$$

we can generate the AAM appearance $\boldsymbol{A}(x)$ defined in the interior of the base mesh $\boldsymbol{s}_0$. The AAM model instance with shape parameters $\boldsymbol{p}$ and appearance parameters $\boldsymbol{\lambda}$ is then created by warping the appearance $\boldsymbol{A}$ from the base mesh $\boldsymbol{s}_0$ to the model shape $\boldsymbol{s}$. This process is illustrated in Fig. 3 for concrete values of $\boldsymbol{p}$ and $\boldsymbol{\lambda}$.

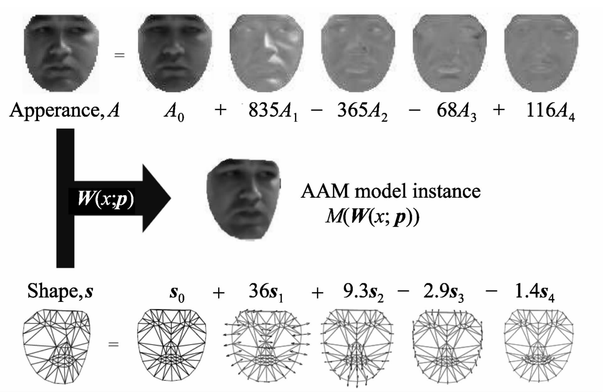


Fig. 3 An example of AAM instantiation

In particular, the pair of meshes $\boldsymbol{s}_0$ and $\boldsymbol{s}$ define a piecewise affine warp from $\boldsymbol{s}_0$ to $\boldsymbol{s}$. We denote this piecewise affine warp $\boldsymbol{W}(x;\boldsymbol{p})$. The final AAM model instance is then computed by forwards warping the appearance $\boldsymbol{A}$ from $\boldsymbol{s}_0$ to $\boldsymbol{s}$ with $\boldsymbol{W}(x;\boldsymbol{p})$. This process is defined by

$$M(\boldsymbol{W}(x;\boldsymbol{p})) = \boldsymbol{A}(x). \tag{8}$$

2 Alignment based on AAMs

2.1 Goal of fitting

Suppose we are given an input image $\boldsymbol{I}(x)$ that we wish to fit an AAM to. And the model instance is $M(\boldsymbol{W}(x;\boldsymbol{p}))$. To define the fitting process, we must formally define the criterion to be optimized in the fitting process. Naturally, we want to minimize the error between $\boldsymbol{I}(x)$ and $M(\boldsymbol{W}(x;\boldsymbol{p})) = \boldsymbol{A}(x)$. Suppose x is a pixel in $\boldsymbol{s}_0$. The corresponding pixel in the input image $\boldsymbol{I}$ is $\boldsymbol{W}(x;\boldsymbol{p})$. At the pixel x the AAM has the appearance

$$\boldsymbol{A}(x) = \boldsymbol{A}_0(x) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(x).$$

At the pixel $\boldsymbol{W}(x;\boldsymbol{p})$, the input image has the intensity $\boldsymbol{I}(\boldsymbol{W}(x;\boldsymbol{p}))$. We minimize the sum of squares of the difference between these two quantities

$$\sum_{x \in \boldsymbol{s}_0} [\boldsymbol{A}_0(x) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(x) - I(\boldsymbol{W}(x;\boldsymbol{p}))]^2. \tag{9}$$

The goal of AAM fitting is then to minimize the expression in Eq. (9) simultaneously with respect to the shape parameters $\boldsymbol{p}$ and the appearance parameters λ. We denote the error image as

$$\boldsymbol{E}(x) = \boldsymbol{A}_0(x) + \sum_{i=1}^{m} \lambda_i \boldsymbol{A}_i(x) - I(\boldsymbol{W}(x;\boldsymbol{p})). \tag{10}$$

Various researchers have put emphasis on minimizing the expression in Eq. (9) by means of descent optimization algorithm[4-6]. However, the disadvantage of these gradient descent! algorithms is that they are very slow. The partial derivatives, Hessian, and gradient direction all need to be recomputed in each iteration[7].

Therefore, an efficient way to update the parameters is proposed. Instead of the previous algorithm, which solves for and the updates the parameters $\boldsymbol{p} \leftarrow \boldsymbol{p} + \nabla \boldsymbol{p}$, we consider the entire warp by composing the current warp with the computed incremental warp with parameters $\nabla \boldsymbol{p}$. In particular, it is possible to update

$$\boldsymbol{W}(x;\boldsymbol{p}) \leftarrow \boldsymbol{W}(x;\boldsymbol{p}) \circ \boldsymbol{W}(x;\nabla \boldsymbol{p}). \tag{11}$$

This compositional approach is different, yet provably equivalent, to the usual additive approach[8]. Ref. [8] presents the Lucas-Kanade image fitting algorithm and forward compositional image alignment algorithm specifically, they are the basis of ICIA.

2.2 ICIA

The ICIA algorithm is a modification of the forwards compositional algorithm where the roles of the template and example image are reversed[8]. Rather than computing the incremental warp with respect to $\boldsymbol{I}(\boldsymbol{W}(x;\boldsymbol{p}))$, it is computed with respect to the template $\boldsymbol{A}_0(x)$.

The compositional framework computes an incremental warp $\boldsymbol{W}(x;\nabla \boldsymbol{p})$ to be composed with the current warp $\boldsymbol{W}(x;\boldsymbol{p})$. The minimization is over

$$\sum_x [I(\boldsymbol{W}(x;\boldsymbol{p})) - \boldsymbol{A}_0(\boldsymbol{W}(x;\nabla \boldsymbol{p}))]^2 \quad (12)$$

with respect to $\nabla \boldsymbol{p}$ and then updating the warp using

$$\boldsymbol{W}(x;\boldsymbol{p}) \leftarrow \boldsymbol{W}(x;\boldsymbol{p}) \circ \boldsymbol{W}(x;\nabla \boldsymbol{p})^{-1}. \quad (13)$$

Taking the Taylor series expansion of Eq. (12) gives

$$\sum_x [I(\boldsymbol{W}(x;\boldsymbol{p}) - \boldsymbol{A}_0(\boldsymbol{W}(x;0)) - \nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}} \nabla \boldsymbol{p})]^2. \quad (14)$$

Assuming again that $\boldsymbol{W}(x;0)$ is the identity warp, the solution to this least squares problem is

$$\nabla \boldsymbol{p} = \boldsymbol{H}^{-1} \sum_x [\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^{\mathrm{T}} [\boldsymbol{I}(\boldsymbol{W}(x;\boldsymbol{p})) - \boldsymbol{A}_0(x)], \quad (15)$$

where $\boldsymbol{H}$ is Hessian matrix with $\boldsymbol{I}$ replaced by

$$\boldsymbol{H} = \sum_x [\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^{\mathrm{T}} [\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]. \quad (16)$$

Since the template $\boldsymbol{A}_0$ is constant and the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ is always evaluated at $\boldsymbol{p}=0$, most of the computation in Eqs. (15) and (16) can be moved to a precomputation step. The result is a very efficient image alignment algorithm. The steps for ICIA can be shown as

Pre-compute:

Step 3 Evaluating the gradient $\nabla \boldsymbol{A}_0$ of the template $\boldsymbol{A}_0(x)$;

Step 4 Evaluating the Jacobian $\frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ at $(x;0)$;

Step 5 Computing the steepest descent images $\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$;

Step 6 Compute the Hessian matrix using Eq. (15).

Iterate until converged:

Step 1 Warpping $\boldsymbol{I}$ with $\boldsymbol{W}(x;\boldsymbol{p})$ to compute $\boldsymbol{I}(\boldsymbol{W}(x;\boldsymbol{p}))$;

Step 2 Computing the error image $\boldsymbol{I}(\boldsymbol{W}(x;p)) - \boldsymbol{A}_0(x)$;

Step 7 Computing $\sum_x [\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}]^T [\boldsymbol{I}(\boldsymbol{W}(x;\boldsymbol{p})) - \boldsymbol{A}_0(x)]$;

Step 8 Computing $\nabla \boldsymbol{p}$ using Eq. (15);

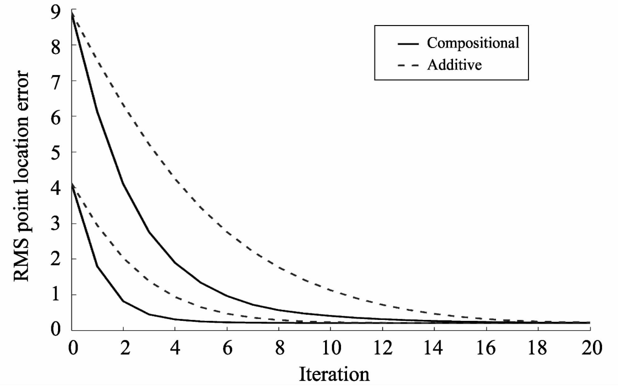
Step 9 Updating the warp

$$\boldsymbol{W}(x;\boldsymbol{p}) \leftarrow \boldsymbol{W}(x;\boldsymbol{p}) \circ \boldsymbol{W}(x;\nabla \boldsymbol{p})^{-1}.$$

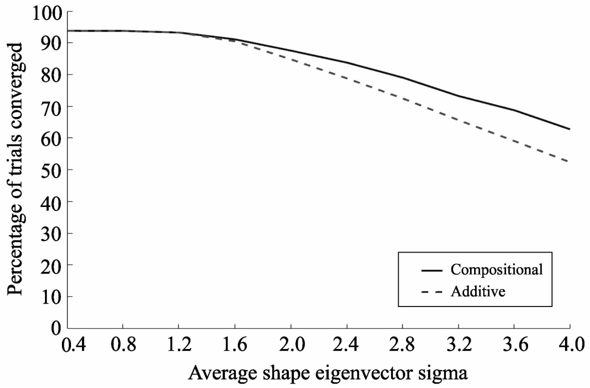
2.3 Comparison between additive and compositional updates

The main advantage of the inverse compositional algorithm is that the Hessian and $\nabla \boldsymbol{A}_0 \frac{\partial \boldsymbol{W}}{\partial \boldsymbol{p}}$ are both constant and so can be precomputed. Since this is most of the computation, the resulting algorithm is very efficient.

Fig. 4 shows the results of comparing the additive and compositional updates to the warp in Step 9.



(a) Rate of convergence perturbing shape



(b) Frequency of convergence perturbing shape

Fig. 4 Results of comparing the additive and compositional updates

3 Location initialization based on curvature

Curvature on 3D surface is the most basic characteristic, which is very useful for 3D-model-based method. Moreno A B[9] computed the curvature of the surface and the mean curvature and divided the face into several kinds of surfaces.

In this paper, we locate the face by means of searching the local maximum of curvature (Fig. 5). It is easy to locate the face via the 4 points-tip of nose, inner corner of the eyes, the middle point of the eyes.

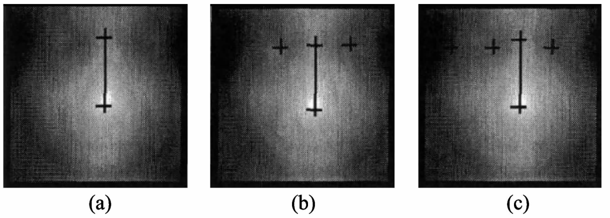


Fig. 5 Locate the face by means of curvature at special points

Depending on this locating method, AAM can give a better performance compared with the original AAM.

4 Conclusion

In this paper, we mainly propose a more efficient fitting algorithm named ICIA compared with the previous fitting approaches based on AAM. Since the AAM model is the basis of the fitting algorithm, the shape model, appearance model and the combined model are also introduced briefly. Finally, we propose a method for locating the face as initialization, which increases the accuracy and speed of fitting process. Fitting process based on AAM is the unique step for the subsequent jobs such as face detection, facial expression recognition and so on. This fitting algorithm affords a robust method for facial feature location, which is faster and accurate.

References

[1] Cootes T F, Edwards G J, Taylor C J. Active appearance models. Proc. of European Conference on Computer Vision 1998(ECCV98), Freiberg, Germany, 1998, 2: 484.

[2] Cootes T F, Taylor C J. Statistical models of appearance for computer vision. University of Manchester, 2001.

[3] Lucas B, Kanade T. An iterative image registration technique with an application to stereo vision. Proc. of the 7th International Joint Conference on Artificial Intelligence (IJCAI1981), 1981, 2: 674-679.

[4] Sclaroff S, Isidoro J. Active blobs. Proc. of the 6th IEEE International Confe rence on Computer Vision, 1998: 1146-1153.

[5] Jones M J, Poggio T. Multidimensional morphable models: A framework for representing and matching object classes. Proc. of the 6th International Conference on Computer Vision(ICCV-98), Mombay, India, 1998: 683-688.

[6] Blanz V, Vetter T. A morphable model for the synthesis of 3D faces. Proc. of the 26th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH'99), 1999: 187-194.

[7] Matthews I, Baker S. Active appearance models revisited. International Journal of Computer Vision, 2004, 60 (2): 135-164.

[8] Baker S, Matthews I. Lucas-Kanade 20 years on: A unifying framework: Part 1: The quantity approximated, the warp update rule, and the gradient descent approximation. International Journal of Computer Vision, 2003, 56 (3): 221-255.

[9] Moreno A B, Sanchez A, Velez J F, et al. Face recognition using 3D surface-extracted descriptors. Proc. of Irish Machine Vision and Image Processing (IMVIP2003), 2003.