Fast Simplex Optimization for Active Appearance Model

Yasser Aidarous* and Renaud Séguier

SUPELEC/IETR
Avenue de la Boulaie CS 47601
F-35576 Cesson-Sévigné CEDEX, France
{yasser.aidarous,renaud.seguier}@supelec.fr

Abstract. This paper presents a fast optimization method for active appearance model based on Nelder & Mead simplex in the case of mouth alignment under different expressions. This optimization defines a new constraint space. It uses a Gaussian mixture to initialize and constraint the search of an optimal solution. The Gaussian mixture is applied on the dominant eigenvectors representing the reduced data given by Principal Component Analysis. The new algorithm constraints avoid calculating errors of solutions that don't represent researched forms and textures. The constraint operator added to simplex verifies in each iteration that the solution belongs to the space of research. The tests performed in the context of generalization (learning and testing datasets are different) on two datasets show that our method presents a better convergence rate and less computation time compared to the AAM classically optimized.

Keywords: Expression analysis, Active Apprearance Model, Nelder Mead simplex.

1 Introduction

The development of the multifunction mobile technologies (photo, video) and the transmission capacity of the wireless networks claims a fast and reliable communication between the user and the machine. For gestures or motion recognition and users localization, it is necessary to align (find landmarks) objects to be recognized. In our applications in Human Machine Interaction (HMI): animate an avatar to communicate with users depending on their emotion state, we seek to align mouths under different expressions. This application puts several constraints on the choice of the algorithm for several reasons:

- Objects to align are high deformable
- Generalization: the analyzed objects are not belonging to the training dataset (unknown faces and mouths)
- Fast alignment

* This work is sponsored by "Région de Bretagne".

T. Wada, F. Huang, and S. Lin (Eds.): PSIVT 2009, LNCS 5414, pp. 106–117, 2009.

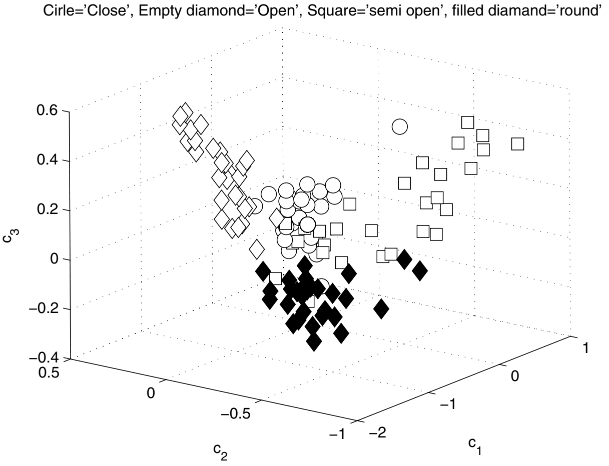

Fig. 1. Representation of different mouth expressions. The space formed by first three eigenvectors of the third PCA.

To reach our goal, we opt to use an Active Appearance Models (AAM). The AAM method allows us to model and synthesize object to align by controlling one appearance vector representing form and texture conjointly. In [9] the appearance vector is constrained using the eigenvalue given by the third PCA, this constraints does not take into account the data repartition in reduced dimension. The fig. 1 represents the distribution of mouths images, under different expressions, in the space defined by the first three PCA eigenvectors. We notice the existence of holes in this space where the variables of appearance do not model one of the expressions looked for (and learned). Then, we are faced with a problem of fragmented space. It urges us to redefine our research space in a way that the proposed solutions represent the searched forms.

We add a new "constraints" operator to a simplex optimization applied on appearance vector. This operator will define a relevant space of constraints for the first dominant variables of appearance. We exploit a Gaussian mixture to initialize and model the space of constraints. The new operator will allow to verify the affiliation of the solution in the search space. It will serve to eliminate the solutions presenting different appearances to those searched in the image without error calculation.

The paper is organized as follow. In section 2 we present briefly the AAM and related works for their improvements. In Section 3, which forms the core of the paper, we present an adaptation of the simplex to the AAM, with specific initialization and constraints using Gaussian mixture. Then, in section 4 we present experimental results in the case of fragmented space generalization. Section 5 concludes the paper with advantages supplied by the use of the new optimization.

2 Background

2.1 Active Appearance Model

AAM [6] uses PCA to encode both shape and texture variation of training dataset. The shape of an object can be represented by vector s and the texture (gray level) by vector g. We apply one PCA on shape and another PCA on texture to create the model, given by:

$$\begin{aligned} \mathrm{s_i} &= \bar{s} + \mathit{\Phi}_s * b_s \\ g_\mathrm{i} &= \bar{g} + \mathit{\Phi}_g * b_g \end{aligned} \tag{1}$$

where s_i and g_i are shape and texture, $\bar{s}$ and $\bar{g}$ are mean shape and mean texture. $\mathit{\Phi}_s$ and $\mathit{\Phi}_g$ are vectors representing variations of orthogonal modes of shape and texture respectively. b_s and b_g are vectors representing parameters of shape and texture. i is the image dataset index. By applying a third PCA on vector $b = \begin{bmatrix} b_s \\ b_g \end{bmatrix}$ we obtain:

$$b = \mathit{\Phi} * c \tag{2}$$

ϕ is a matrix of d_c eigenvectors obtained by PCA and c is the appearance parameters vector.

The modifications of c parameters change both shape and texture of the object. Each object is defined by the appearance vector c and pose vector t:

$$t = \begin{bmatrix} t_x \; t_y \; \theta \; S \end{bmatrix}^T \tag{3}$$

where t_x and t_y are x and y axis translation, θ is the angle of orientation and S is Scale.

AAM learns the linear regression models which gives us the predicted modifications of model parameters δc and δt:

$$\begin{aligned} \delta c &= R_c G \\ \delta t &= R_t G \end{aligned} \tag{4}$$

R_c and R_t are the appearance and pose regression matrix respectively. The model search is driven by the residual G of the search image and model reconstruction.

Later in this paper we note Regression Matrix by ‘RM’.

2.2 Related Works

The segmentation phase proposed by Cootes [6] use an optimization based on multiple linear regression. This optimization have difficulties to align objects which present a fragmented space in reduced dimension (fig. 1). The problem of fragmented space may be treated with a hierarchical method or by executing several AAM (each AAM representing one expressions) but the time consumption must be multiplied by the number of expressions. [1] proposes hierarchical decomposition of the face by several components (eyes, mouths,...), and models the variability of each component. This decomposition is also used in [2] without worrying about the pose parameters to analyze face expressions. In [2] every possible expression of each hierarchical component is modeled by an AAM (for example: two AAM for mouth, one for open mouths and the other for closed mouths). Several AAM was used to overcome the problem of fragmented space. This method was performed in multi view problem in [5] using Direct Appearance Models [3] in which the form is predicted from the texture directly. Several model was created in [4] corresponding to different face expressions and in the search phase it keeps the model which fits the image with minimal error. A Gaussian mixture was used in [8] to make the difference between the different expression classes of the same object modeled by AAM. The mixture is applied on the real learning data images. Each expression class is represented by a Gaussian and defines a model with a specific RM. During the search phase, the number of AAM equals the number of expressions applied. The retained solution generates the minimal error between the generated model and the input image.

[10] suggested, within the framework of fragmented space problem, using an algorithm of optimization based on Simplex to optimize the choice of the vector associating the appearance and the pose. The initialization, of all appearance variables, was made by a Gaussian mixture which allows the simplex to consider all fragments in the search for the optimal solution. The authors use the constraints defined by [9]. However the proposed suffers from an important time execution. In the next section we propose a new initialization and constraints to overcome the fragmented space problem. Unlike in [10] we:

- Apply the Expectation Maximization algorithm on the dominant eigenvectors (defined in the next section) given by PCA to model search space with Gaussian mixture
- Initialize the dominant appearance variables (associated to dominant eigenvectors) using the Gaussian mixture
- Initialize remained variables of appearance vector randomly in the space defined by associated eigenvalues
- Constraint the search of optimal appearance vector using the Gaussian mixture elaborated on the dominant appearance variables. We add new operator to the simplex to verify the affiliation of the proposed solution to the new defined search space
- Constraint remained variables of appearance vector using constraints defined in [9]

This new optimisation reduces the time consuming by avoiding calculating errors of solutions don't belonging to the new search space.

Later in this paper we note our algorithm 'SPGM'.

3 Simplex and GM Optimization for AAM

3.1 Nelder and Mead Simplex

The simplex Nelder & Mead can find the minimum of a function of n variables iteratively with $n+1$ initial solutions. For two variables, simplex is a triangle. The simplex compares the values of the function on each summit point of triangle. Thus the summit point where the function is the highest is replaced by another which will be calculated based on the existing summits. When we have to align an object (detecting characteristics points and texture) with AAM, we must find a vector v that minimizes the sum of quadratic errors e^2 with:

$$v = \begin{bmatrix} c \\ t \end{bmatrix} \quad and: \quad E(v) = \sum_{i=1}^{M} e_i^2 \tag{5}$$

where c is the appearance vector, M is the number of pixels of the model and e_i the error in the pixel i.

3.2 Simplex and GM

After creating the model, we obtain the appearance vector c representing each image from the learning dataset. We consider that the appearance vector is constituted of two sub-appearance vectors:

$$c = \begin{bmatrix} c_\lambda \\ c_{rand} \end{bmatrix} \tag{6}$$

where c_λ is the sub-appearance vector representing the dominant eigenvectors and c_{rand} represents the sub-appearance vector given with the low eigenvectors.

The fig. 2 represents the accumulation percentage of variance of the eigenvectors given by the third PCA (Eq.3). The number of dominant eigenvectors λ is the minimal number of eigenvectors that express 80% of data variance. For example in the case of 116 images from France Telecom dataset, According to the accumulative variance represented by eigenvectors given by the third PCA (fig. 2), the number of eigenvectors takes into account is $\lambda = 6$. However, the images form a fragmented space (fig. 1) under the first dominant eigenvectors and uniform space under the other variables. Then we can't consider the variables to optimise with the same importance. This lead us to initialize and constraint the appearance variables using two different manner: Gaussian mixture for the dominant variables and uniform for the other variables. We calculate a Gaussian mixture using Expectation Maximization (EM) [12] algorithm, during the modeling phase, on the sub-appearance vectors c_λ given by learning dataset images in the reduced space given by the third PCA:

$$g(c_\lambda, \Theta) = \sum_{k=1}^{N_{gauss}} \pi_k \aleph(\mu_k, \Gamma_k) \tag{7}$$

where N_{gauss} is the number of Gaussians in the mixture, π_k is the weights associated to the Gaussian k. μ_k is the mean of the Gaussian k, Γ_k is the covariance matrix of the Gaussian k, $\aleph(\mu_k, \Gamma_k)$ is the normal distribution defined by the mean μ_k and the covariance matrix Γ_k, and $\Theta = (\pi_1, ...\pi_k, \mu_1...\mu_{N_{gauss}}, \Gamma_1, ...\Gamma_{N_{gauss}})$ the vector that characterizes the mixture.

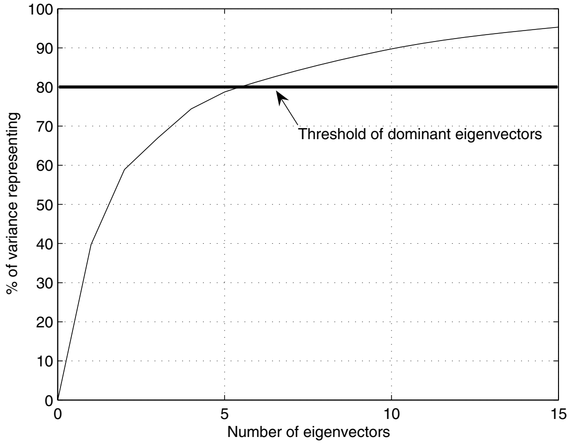

Fig. 2. Accumulative variance represented with the number of eigenvector taking into account given by the third PCA

SPGM initialization
Appearance vector initialization is done in two steps:

- Initialize c_λ: using means and covariances of Gaussians allow us to choose randomly $n+1$ sub-appearance vectors c_λ^j belonging to the space defined by the mixture. This is allow us to initialize c_λ according to the distribution of learning data.
- Initialize c_{rand}: we choose randomly $n+1$ vectors in the space is defined by the eigenvalues of the third PCA in the AAM as follow:

$$-3\sqrt{\lambda_j} < c_{rand}^j < 3\sqrt{\lambda_j} \tag{8}$$

λ_j is the j^{nd} eigenvalue and c_{rand}^j is the j^{th} variable.

Knowing that the AAM are robust to 10% in translation and zoom [7], we initialize the $n+1$ vectors pose t_j (Eq.3) (Zoom, rotation, translation in x and y) in a range of 10% of the initial pose vector (defined by a face detector) characterizing the object to align.

SPGM constraints
After each simplex iteration the proposed solution $v_{new} = [c_\lambda^{new}, c_{rand}^{new}, t_{new}]^T$ must verify constraints. Unlike [10] where the mixture is applied on all the parameters of the appearance vector, we are implementing the Gaussian mixture on the appearance variables corresponding to the dominant eigenvectors. The Gaussian mixture serves a compelling solutions proposed by the simplex remains in a space defined by Gaussians. The remaining appearance variables are constrained by the equation 8. The constraints are defined for each of 3 vectors forming v_{new}:

c_λ^{new}: The mixture, defined by Θ delineate the area of search for the optimal solution. Each Gaussian is equated with a class of expression. After each proposed solution $v_{new} = [c_\lambda^{new}, c_{rand}^{new}, t_{new}]^T$, we need to know if c_λ^{new} can be assigned to a class or not. We use a Bayesian model of affiliation. In our case, we have no a priori knowledge on the class to which the image belongs, and thus the probabilities of occurrence for each class are equal. The assignment is identical according to $P(c_\lambda^{new}|I_k)$ (probability of occurrence of c_λ^{new} knowing I_k) or to $P(I_k|c_\lambda^{new})$ (probability of occurrence of I_k knowing c_λ^{new}). In the multi normal case, the density of probability $f_k(c_\lambda^{new})$ of c_λ^{new} knowing I_k is:

$$\begin{aligned} f_k(c_\lambda^{new}) &= (2\pi_k)^{-p/2}\,|\Gamma_k|^{-1/2}\exp\{-\tfrac{1}{2}r_k(c_\lambda^{new})\} \\ where: r_k(c_\lambda^{new}) &= (c_\lambda^{new}-\mu_k)^T\Gamma_k^{-1}(c_\lambda^{new}-\mu_k) \end{aligned} \tag{9}$$

So the assignment of a new solution to a class (Gaussian) will obey to the following rule:

$$\begin{aligned} &choose\ \hat{k}\ as: \\ &f_{\hat{k}}(c_\lambda^{new}) = \max_{k\leq N_{gauss}}\{f_k(c_\lambda^{new})\} \end{aligned} \tag{10}$$

This is equivalent to finding the minimum on k of the function $sc_k(c)$ which will be called *Discriminant Score*:

$$sc_k(c_\lambda^{new}) = r_k(c_\lambda^{new}) + log(|\Gamma_k|) \tag{11}$$

The discriminant score coincides with the Mahalanobis distance added to a term describing the dispersion of data on each Gaussian. To assign new solution to a Gaussian, we have to search the near Gaussian center following the Mahalonobis distance.

We define the operator *Constraints* which ensure the affiliation of the solution c_λ^{new} to the search space before calculating the error. The new operator will calculate the affiliation scores of this solution for each class and compare it to a threshold Th_{sc_k}. Th_{sc_k} is fixed to the sc_k value score when the solution reaches $3\sigma_k$ (three times the standard deviation), we will ensure that the proposed solutions belong or are fairly close to the forms search space. Two cases are:

- The solution c_λ^{new} presents a score sc_k less than Th_{sc_k}. Then c_λ^{new} belongs to the constraint space. The constraint operator allows the calculation of the error, and c_{new} is used by the simplex.
- The solution c_λ^{new} presents a score sc_k over Th_{sc_k}. The constraint operator replaces c_λ^{new} with another randomly solution belonging to the Gaussian where the score is maximal.

c_{rand}^{new}: like in [10] the variables are constraint to remain in the space formed by the equation 8.

t_{new}: The pose parameters are constrained to respect the constraints in [9]:

$$-0.1 \times T_0 < t_{new} < 0.1 \times T_0 \tag{12}$$

where T_0 is the true localization of the mouth in the image.

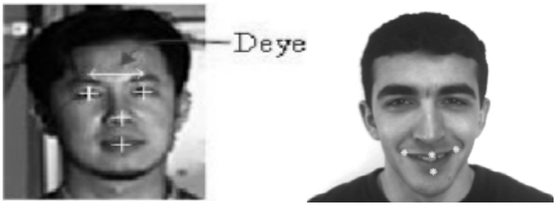

Fig. 3. D_{eye} distance and the 4 points used to calculate the error marking

Tests constraints on c_{λ}^{new} are not time consuming compared to the errors calculations (using warping). The constraints imposed on c_{λ}^{new} allow us to remove solutions knowing in advance that they do not represent forms searched. Then the algorithm converge rapidly to the optimal solution. The algorithm will stop after a certain number of iterations (to ensure maximum time of treatment), or when it will converge in population. The convergence in population is obtained if difference between the error values of the proposed solutions do not pass the threshold S_E. In the case of error normalization, done by Stegmann [9], the mean value E_{mean} of the error is stable on different images when we use PCA to project the object in the correct alignment. We propose to settle $S_E = 0.1 \times E_{mean}$.

4 Experiments

We have elaborated our tests in the context of generalization (the learning datasets and test ones are different) in mouths alignment. In order to show the efficiency of our method, we are going to compare the method by using regression matrix and the method proposed in [10]: simplex optimization initialized using Gaussian mixture (SPIGM). We test our method and that of regression matrix in both following cases:

- Learning on France telecom dataset (116 mouth images under 4 different expressions) and test on 40 images of Supelec dataset
- Learning on Supelec dataset (40 mouth images under 4 different expressions) and test on 116 images of France telecom dataset

The 4 mouths expressions segmented in the two performed tests are shown fig. 4. To eliminate vagueness of ground truth markings we consider results starting

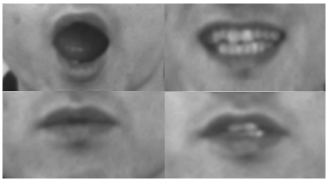

Fig. 4. Example of mouth expressions in the SUPELEC dataset

from 10% of D_{eye}. We consider that the algorithm diverge when the difference between any found point and ground truth point is higher than $0.25 \times D_{eye}$.

4.1 Error Marking

To qualify the convergence of the AAM we will define an error marking. This error $f_i (i = 1, 2, 3, 4)$ is calculated for each point i of the mouth (fig. 3) such as:

$$f_i = \frac{(p_i^{find} - p_i^{real})}{D_{eye}} \qquad and: \qquad err = \max(f_i) \tag{13}$$

where err is the marking error, p_i^{real} are the coordinates of the ground truth of the i^{th} marking point of the mouth and p_i^{find} the coordinate of the i^{th} marking point of the mouth found by AAM.

The algorithm converges when the 4 errors are lower than a given convergence threshold. The threshold was calculated according to the distance between the eyes D_{eye}.

4.2 Results

The figures 5 and 6 represent a comparison between optimization by RM, by SPIGM proposed in [10] (simplex optimization initialized using Gaussian mixture) and by SPGM (Simplex optimization constrained and initialize with Gaussian mixture applied on the dominant eigenvectors). To evaluate time complexity, we use the number of error calculation used to converge. This number is reduced to warping number which is the most time consuming step in the error

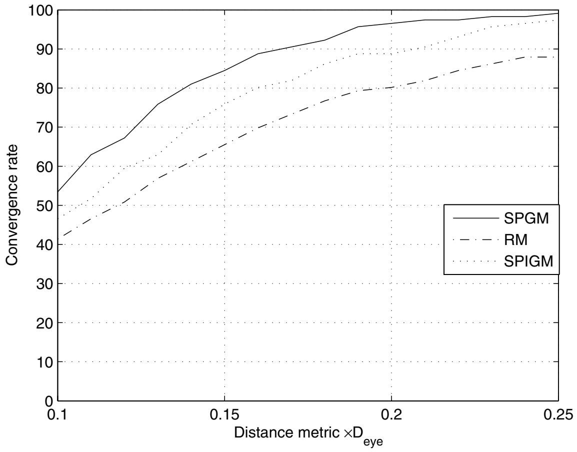

Fig. 5. Results obtained on the France Telecom dataset using 40 images from SUPELEC during learning phase

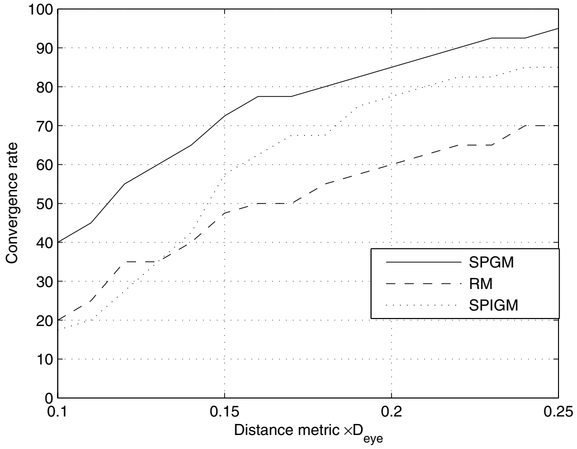

Fig. 6. Results obtained on the SUPELEC dataset using 116 images from France Telecom dataset in learning phase

calculation procedure. In both tests we notice that SPGM is more efficient than the two others optimization method. For a convergence rate of $0,15 \times D_{eye}$, the RM presents a convergence rate of 65% and 48%, whereas the SPIGM realises a convergence rate of 76% ans 58% and the SPGM method reach a convergence rate of 85% and 72%. The results obtained show the efficiency of SPGM in generalisation of mouth presents fragmented space. The RM use introductory knowledge on segmented mouth by learning a relation between the error and the parameters which turns out to be wrong in the case of generalization. The simplex adapts to the mouth to be segmented without following privileged directions provided before. This allows the simplex to find shapes in coherence with the image in entrance and not with learning images.

In order to certify the RM convergence, we have to make a grid on scale and translation in x and y. The number of necessary warpings for RM is 465 warping. The SPIGM loses a lot of time to calculate errors of solutions that are found in areas devoid of data that does not represent forms searched. This is due to constraints applied that don't reflect the nature of forms in search space.

Figure 7 shows the convergence at $0,15 \times D_{eye}$ of our method as well as the results variance, in relation to mean results obtained on 10 attempts, for the two tests. We notice that our algorithm rapidly converges toward a stable convergence rate (higher than RM). The SPGM converges, in both tests, toward a stable convergence rate (minimal variance) in warping number closer to 280 warping for France Telecom dataset and 250 for Supelec dataset. The difference in term of warping number needed to converge in the two tests is due to the number of training instances (116 images in the Telecom dataset and 40 images for Supelec dataset) and test instances (40 images for Supelec). But the earning in warping presented by SPGM is about 40% (in the both tests) comparing to RM.

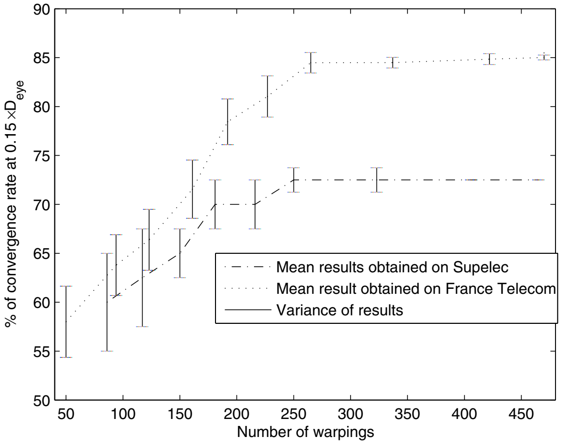

Fig. 7. Percentage of convergence at $0.15 \times D_{eye}$ of each elaborated test versus the number of warping needed by SPGM method to attempt this percentage

5 Conclusion

We have elaborated an AAM optimization method based on a Nelder & Mead simplex initialized and constrained by a Gaussian mixture. This method allow us to overcome the fragmented space problem and converge speedly. Contrary to [10]:

- The Gaussian mixture is applied on the dominant variables corresponding of the dominant eigenvectors given by the third PCA of AAM and not on the all appearance parameters.
- The initialization of simplex use Gaussian mixture to initialize dominant appearance variables and uniform random to initialize the remain variables like in [9].
- The constraints on dominant variables use a new operator in simplex which verify the affiliation of each proposed solution to one of the Gaussians of the mixture. The constraints on the remain variables are like in [9].

The method we are suggesting in this paper presents higher convergence rate comparing to classical method based on RM and simplex optimization initialized using Gaussian mixture (SPIGM). The affiliation test applied in our proposition allow us to avoid calculating the errors of all solutions proposed by simplex operators. This test is less time consuming compared to error calculating, that allow us to save about 40% of time needed by RM and SPIGM.

References

1. Zalewski, L., Cong, S.: 2nd statistical models of facial expressions for realistic 3d avatar animation. In: International Conference on Computer Vision and Pattern recognition (2005)

2. Cootes, T.F., Edwards, G.J., Taylor, C.J.: A high resolution grammatical model for face representation and sketching. In: International Conference on Computer Vision and Pattern recognition (2005)
3. Hou, X., Li, S.Z., Zhang, H., Cheng, Q.: Direct Appearance Models. In: International Conference on Computer Vision and Pattern recognition (2001)
4. Cootes, T.F., Walker, K.N., Taylor, C.J.: View based active appearance models. In: International Conference on Automatic Face and Gesture Recognition (2000)
5. Li, S.Z., Zhang, H.J., Cheng, Q.S.: Mutli view face alignment using direct appearance models. In: International Conference on Automatic Face and Gesture Recognition (2002)
6. Cootes, T.F., Edwards, G.J., Taylor, C.J.: Active Appearance Models. In: Burkhardt, H., Neumann, B. (eds.) ECCV 1998. LNCS, vol. 1407, p. 484. Springer, Heidelberg (1998)
7. Cristinacee, D., Cootes, T.F.: Comparison of Shape Constrained Facial Feature Detectors. In: Face and Gesture Recognition (2004)
8. Christoudias, C.M., Darrell, T.: On Modeling Nonlinear Shape-and-Texture Appearance Manifolds. In: Computer Vision and Pattern Recognition (2005)
9. Stegmann, M.B.: Active appearance models Theory, Extension and Cases. Master Thesis IMM-EKS, LYNGBY (2000)
10. Aidarous, Y., Le Gallou, S., Seguier, R.: Simplex Optimization Initialized by Gaussian Mixture for Active Appearance Models. In: Digital Image Computing Techniques and Applications (2007)
11. Nelder, J.A., Mead, R.: A Simplex Method for Function Minimization. Computer Journal, 308–313 (1965)
12. Vlassis, N., Likas, A.: A greedy EM algorithm for Gaussian mixture learning. Neural Processing Letters, 77–87 (2002)