Active Morphable Model: An Efficient Method for Face Analysis

Xun Xu[†] xunxu@ifp.uiuc.edu
Changshui Zhang[‡] zcs@mail.tsinghua.edu.cn
Thomas S. Huang[†] huang@ifp.uiuc.edu

[†]*Beckman Institute*
University of Illinois at Urbana-Champaign
Urbana, IL 61801, USA

[‡] *Department of Automation*
Tsinghua University
Beijing 100084, P.R.China

Abstract

Multidimensional Morphable Model is a powerful model to analyze and synthesize human faces. However, the stochastic gradient descent algorithm adopted to match the Morphable Model to a novel face image is not efficient enough. In this paper, a very efficient optimization method devised for Morphable Model matching is proposed, called Active Morphable Model (AMM). The kernel of AMM is an iterative algorithm directly utilizing the heuristic information provided by the novel image, and updating the model parameters in a computationally economic fashion. AMM is more efficient than general optimization methods in matching a Morphable Model, it has much higher convergent rate and matching speed. Furthermore, it is insensitive to the initial estimation of the face pose, and is robust when used to match novel faces with large variations in translation, rotation and scaling. Experimental results are given to validate the efficiency and robustness of the proposed method.

1. Introduction

Face analysis is one of the most important tasks in computer vision and pattern recognition. During the recent years, model based approaches have become prevailing in face analysis. Many shape and/or appearance models for interpreting human faces have been proposed, including Active Shape Model (ASM)[3][5], Active Appearance Model (AAM)[4][5], Direct Appearance Model (DAM)[7], and Multidimensional Morphable Model [8][9] etc.

Multidimensional Morphable Model [8][9] (also referred as *Morphable Model*, or *Linear Morphable Model*), is a linear model for images of objects of a certain class. It separates and vectorizes texture and shape in an ingenious way, and represents shape as a dense flow, which makes it an expressive and powerful tool to analyze and synthesize human faces. In practice, it has been successfully applied to many problems including face recognition [8], gaze estimation [11] and 3D face reconstruction [2] and speech video synthesis [6] etc.

The Morphable Model possesses many merits such as efficiency of training and accuracy of matching. However, in the original framework of the Morphable Model, the matching problem, i.e. fitting the Morphable Model to a novel image, is treated as an general function minimization problem, and solved through stochastic gradient descent algorithm. As we will show, it is not a very efficient method, since such a common optimization algorithm does not efficiently exploit the heuristic information provided by the novel image, thus has a low convergent rate.

In this paper, an efficient optimization method devised for Morphable Model matching is proposed, which is called Active Morphable Model (AMM). The basic ideas of AMM are: (1) Exploiting the information of the novel image efficiently to guide the optimization and speed the search. (2) Updating the model parameters at each iteration in a computationally economic way. AMM gets its name because it is more "active" than a common optimization method.

The virtues of AMM include: (1) Through direct utilizing the heuristic information in the input image, it can guide the search towards the optimum, so that high convergent rate is achieved. (2) AMM is a pure linear method, throughout the matching process, no non-linear optimization methods are adopted, all parameters are updated in a computationally efficient fashion. (3) As our experiments will show, AMM is insensitive to the initial estimation of model parameters, and is robust when applied to match novel faces with large variations in translation, scaling and rotation, in other words, AMM has a larger "capture range".

The rest of this paper is organized as follows: In Section 2, we will briefly review the original Morphable Model. After that, we will propose the Active Morphable Model, firstly introducing its motivation and key points, then giving the algorithm in detail. Section 3 will show the experimental results to validate the efficiency and robustness of AMM, and give a short analysis on AMM's performance. In Section 4, a discussion on AMM is given by comparing it with other face modeling and analyzing methods. Section 5 concludes.

2. **Active Morphable Model**

2.1. Morphable Model

In this section, a brief review of Morphable Model is given, for detailed description, the reader can refer to [8] or [9].

In Morphable Model, shape S_j is given as a pixel-wise correspondence between the reference image I_0 and each example image I_j. Since S_j is a dense mapping, it is often referred to as *correspondence field*. In the training phase, the correspondence fields are established through an optical flow algorithm [13][14].

The texture is defined as:

$$T_j(x,y) = I_j\left(S_j(x,y)\right) \triangleq I_j \circ S_j(x,y)$$

where $\circ$ is a warping operator. T_j is a set of shape free images, called *texture vectors*.

Morphable Model is defined as I^{model}, the set of images produced by the model with a set of parameters.

$$I^{model} \circ \mathcal{A}\left(\sum_{i=0}^{N} c_i S_i\right) = \sum_{j=0}^{N} b_j T_j \tag{1}$$

where $\mathbf{c} = [c_0, c_1, \ldots, c_N]$ and $\mathbf{b} = [b_0, b_1, \ldots, b_N]$ are linear combination coefficients of prototype shapes S_i and textures T_j, respectively. $\mathcal{A} : : \mathcal{R}^2 \rightarrow \mathcal{R}^2$ is an affine transform parameterized by $\mathbf{p}$, which is called *pose parameter*. Given $\mathbf{c}$, $\mathbf{b}$, and $\mathbf{p}$, a model image can be rendered. It can be noted that in Morphable Model, the parameters for shape and texture are independent, endowing the model with great expressiveness.

Applying the Morphable Model to interpret a novel image is done in an analysis-by-synthesis fashion, through matching the model to the novel image, i.e. finding $\mathbf{c}$, $\mathbf{b}$ and $\mathbf{p}$ for the model so that the rendered model image best resembles the novel image. The matching process is regarded as an optimization problem with the error function defined as:

$$E(\mathbf{c},\mathbf{b},\mathbf{p}) = \frac{1}{2}\sum_{\hat{x},\hat{y}}\left[I^{novel}(\hat{x},\hat{y}) - I^{model}(\hat{x},\hat{y})\right]^2 \tag{2}$$

For computational feasibility, the error is calculated in the model frame instead of the novel image frame as in (2), therefore the error function is written as:

$$E(\mathbf{c},\mathbf{b},\mathbf{p}) = \frac{1}{2}\sum_{x,y}\left[I^{novel} \circ \mathcal{A}\left(\sum_{i=0}^{N} c_i S_i(x,y)\right) - \sum_{j=0}^{N} b_j T_j(x,y)\right]^2 \tag{3}$$

To minimize this error function, any optimization algorithm can be used, in most works applying Morphable Model [8][11][12][2], the stochastic gradient descent algorithm is adopted.

2.2. Active Morphable Model

Morphable model is an expressive and powerful model. It separates and vectorizes texture and shape in an ingenious way, and represents shape as a dense flow field, which extends the number of shape feature points to its limit and can capture very subtle changes in face shape (even wrinkles, for example) [2]. However, treating Morphable Model matching as a general optimization problem and solving it through a common algorithm, such as stochastic gradient descent, is actually not the most efficient way. The reason is that it does not efficiently utilize the information provided by the novel image. At each iteration, the information exploited to update model parameters is only the *algebraic* difference between the novel image and model image. Therefore, the parameters cannot be updated effectively, leading to the low convergent rate.

The above analysis motivates us to devise a more specific and efficient method for the matching problem, thus the Active Morphable Model (AMM) is proposed. The basic ideas of AMM are: (1) Exploiting the information of the novel image effectively to guide the optimization process and thus speed the search. (2) Updating the model parameters at each iteration in a fashion that is computationally efficient.

In fact, the heuristic information that can be provided by the novel image is much richer than the algebraic difference between it and the model image. For example, texture can be directly updated using the pixel information in the novel image, as DAM [7] does. Furthermore, although the novel image does not contain shape information directly, an optical flow calculation between it and the model image can partly recover the shape of the novel face to be matched [1][10], which will guide the updates of shape and pose parameters. AMM intends to utilize all these information directly, and effectively update the model parameters in an efficient way at each iteration.

In error function (3), all parameters appear together and are tightly coupled with each other, which makes it difficult to update them efficiently. Note that the objective of model matching is to find the shape and texture parameters with which a face best fitting the novel face can be synthesized. Therefore, equivalently to minimizing (3) under the assumption that shape and texture are independent of each other, we can turn to minimize the shape error:

$$E_s = \sum_{x,y}\left\|S^*(x,y) - \mathcal{A}\left(\sum_{i=0}^{N} c_i S_i(x,y)\right)\right\|^2 \tag{4}$$

and the texture error:

$$E_t = \sum_{x,y}\left\|T^*(x,y) - \sum_{j=0}^{N} b_j T_j(x,y)\right\|^2 \tag{5}$$

at the same time.

The true optimum of shape S^* and texture T^* can never be known. However, we are able to estimate them using the heuristic information extracted from the novel image. S^* can be measured through an optical flow algorithm, and T^* can be estimated through warping the novel image back onto the reference face using the current pose and shape parameters.

As mentioned above, shape and texture parameters are independent in Morphable Model, which makes it feasible to separate the updates of shape and texture, and interleave them to minimize E_s and E_t iteratively.

2.2.1. Model Definition in AMM. The original Morphable Model synthesizes model faces through linearly combining prototype shape and texture vectors, a straightforward modification is to replace this with a PCA (Principal Component Analysis) model to remove the linear dependency among the prototypes and reduce the dimensionality of both the shape and texture parameters. We represent the shape and texture in vector form:

$$S = [x_1, x_2, \ldots x_L \mid y_1, y_2, \ldots y_L]^\tau$$

and $T = [R_1, G_1, B_1, \ldots, R_L, G_L, B_L]^\tau$

where L is the number of the pixels in the reference face image. (x_i, y_i) is the location of the i^{th} pixel with a value of $(R_i\ G_i\ B_i)$. PCA representation of shape is given as:

$$S = \mu + \Phi\vec{\alpha}$$

where μ is the mean shape vector and Φ is an $L \times l$ matrix constructed with the first l shape eigen vectors. The l dimensional shape parameter vector $\vec{\alpha}$ gives the compact representation of shape. Similarly, texture is given as:

$$T = \nu + \Omega\vec{\beta}$$

where $\vec{\beta}$ is the m dimensional texture parameter vector.

For 2D face matching, affine transform $\mathcal{A}$ degenerates into a similarity transform, with scaling parameter s, in-plane rotation θ and translation (t_x, t_y). The shape generated by the model is:

$$S^{model} = \mathbf{R}(\mu + \Phi\alpha) + \mathbf{t} \tag{6}$$

where $\mathbf{R}$ is a $2L \times 2L$ block scaling and rotation matrix:

$$\mathbf{R} = s\left[\begin{array}{c|c} \cos\theta \cdot \mathbf{I}_L & -\sin\theta \cdot \mathbf{I}_L \\ \hline \sin\theta \cdot \mathbf{I}_L & \cos\theta \cdot \mathbf{I}_L \end{array}\right]$$

and $\mathbf{t}$ is the $2L$ dimensional translation vector:

$$\mathbf{t} = [\underbrace{t_x \ \cdots \ t_x}_{L} \mid \underbrace{t_y \ \cdots \ t_y}_{L}]^\tau$$

In order to model global intensity variation of images and also the color sensing attributes of different imaging devices, a color transform, parameterized by R, G, B gains $(\gamma_r, \gamma_g, \gamma_b)$ and shifts $(\sigma_r, \sigma_g, \sigma_b)$, is applied to the texture generated by the PCA model, thus gives the model texture T^{model} as:

$$T^{model} = \mathbf{C}(\nu + \Omega\vec{\beta}) + \mathbf{s} \tag{7}$$

where $\mathbf{C}$ is a $3L \times 3L$ diagonal color gain matrix:

$$\mathbf{C} = \mathrm{diag}(\gamma_r, \gamma_g, \gamma_b \mid \ldots \mid \gamma_r, \gamma_g, \gamma_b)$$

and $\mathbf{s}$ is the $3L$ dimensional color shift vector:

$$\mathbf{s} = [\sigma_r, \sigma_g, \sigma_b \mid \ldots \mid \sigma_r, \sigma_g, \sigma_b]^\tau$$

2.2.2. Shape Update. With the vector form of the PCA shape (6), the shape error (4) can be rewritten as:

$$E_s = \left\| \mathbf{R}(\mu + \Phi\alpha) + \mathbf{t} - S^* \right\|^2$$

and in order to update the shape and pose parameters we should minimize E_s, i.e. solve:

$$\arg\min_{s,\theta,t_x,t_y,\vec{\alpha}} \left\| \mathbf{R}(\mu + \Phi\vec{\alpha}) + \mathbf{t} - S^* \right\|^2 \tag{8}$$

Unfortunately, it is nonlinear with respect to s, θ and $\vec{\alpha}$. However, we can solve this problem efficiently with a two-stage linear approach. At the first stage, we combine s and θ into $s_x \equiv s\cos\theta$ and $s_y \equiv s\sin\theta$, then define $\overrightarrow{\alpha_x} \equiv s_x\vec{\alpha}$ and $\overrightarrow{\alpha_y} \equiv s_y\vec{\alpha}$. Now (8) becomes a standard least-square problem:

$$\arg\min_{\mathbf{x}} \left\| \mathbf{P}\mathbf{x} - S^* \right\|^2$$

where $\mathbf{P} = \left[\mu \quad \mathbf{U}\mu \quad \mathbf{V}_\mathbf{x} \quad \mathbf{V}_\mathbf{y} \quad \Phi \quad \mathbf{U}\Phi\right]$, with:

$$\mathbf{U} \triangleq \left[\begin{array}{c|c} \mathbf{O} & \mathbf{I}_L \\ \hline \mathbf{I}_L & \mathbf{O} \end{array}\right], \mathbf{V}_\mathbf{x} \triangleq \left[\begin{array}{c} \mathbf{1}_L \\ \hline \mathbf{0}_L \end{array}\right], \mathbf{V}_\mathbf{y} \triangleq \left[\begin{array}{c} \mathbf{0}_L \\ \hline \mathbf{1}_L \end{array}\right]$$

and $\mathbf{x} = \left[s_x \quad s_y \quad t_x \quad t_y \quad \overrightarrow{\alpha_x}^\tau \quad \overrightarrow{\alpha_y}^\tau\right]^\tau$ is the $(4+2l)$ dimensional solution vector. A least-square solution (pseudo-inverse solution) is simply given as:

$$\mathbf{x} = \mathbf{P}^\dagger S^* \equiv (\mathbf{P}^\tau\mathbf{P})^{-1}\mathbf{P}^\tau S^* \tag{9}$$

From which (t_x, t_y) can be acquired directly, and s, θ can be calculated immediately by their definition. In order to get the shape parameter $\vec{\alpha}$, we solve another linear least-square problem at the second stage:

$$\arg\min_{\vec{\alpha}} \left(\left\| s_x\vec{\alpha} - \overrightarrow{\alpha_x} \right\|^2 + \left\| s_y\vec{\alpha} - \overrightarrow{\alpha_y} \right\|^2 \right)$$

It can be shown that:

$$\vec{\alpha} = \frac{s_x\overrightarrow{\alpha_x} + s_y\overrightarrow{\alpha_y}}{{s_x}^2 + {s_y}^2} = \frac{1}{s}\left(\cos\theta \cdot \overrightarrow{\alpha_x} + \sin\theta \cdot \overrightarrow{\alpha_y}\right)$$

Our experiments have shown that the solution obtained through this two-stage linear approach is constantly reliable. With this approach, we can update the shape and pose parameters in a computationally very efficient way at each iteration.

2.2.3. Texture Update. Similarly to the shape update, now rewriting texture error function (5) with (7):

$$E_t = \left\| \mathbf{C}\left(\nu + \Omega\vec{\beta}\right) + \mathbf{s} - T^* \right\|^2$$

and we should solve:

$$\arg\min_{\gamma_r,\gamma_g,\gamma_b,\sigma_r,\sigma_g,\sigma_b,\vec{\beta}} \left\| \mathbf{C}\left(\nu + \Omega\vec{\beta}\right) + \mathbf{s} - T^* \right\|^2 \qquad (10)$$

Notice that (10) shares exactly the same form as (8), therefore, it can be solved through the two-stage linear approach as we do in 2.2.2. Note that in this texture update step, the color transform parameters are updated as well as the texture parameters. It is much more efficient than solving them separately in different steps.

2.3. AMM Search

The basic idea of utilizing the heuristic information in the novel image and the shape and texture updating methods detailed in 2.2 lead to the AMM search algorithm, summarized below:

1. **Initialization:** Set shape parameter $\vec{\alpha}$, texture parameter $\vec{\beta}$, pose and color transform parameters with initial estimates. Render the model face I^{model} with $\vec{\alpha}$ and $\vec{\beta}$. Typically, $\vec{\alpha}$ and $\vec{\beta}$ are set to zero vectors so that the mean shape and texture are used, color transform is set to an identity one, pose is given by the output of a face detection algorithm.

Do 2 through 4 until convergence:

2. **Shape Update**

   2a. Map the novel image I^{novel} onto the model face frame as $\tilde{I}^{novel}$, using current pose parameters. Apply color transform to I^{model}, obtaining $\tilde{I}^{model}$.

   2b. Calculate the optical flow between $\tilde{I}^{model}$ and $\tilde{I}^{novel}$.

   2c. Estimate the optimal shape S^* through combining the optical flow with current pose and shape parameters.

   2d. Update the pose and shape parameters through the two-stage linear approach in 2.2.2.

3. **Texture Update**

   3a. Estimate the optimal texture T^* through back-warping I^{novel} onto the reference face, with the updated pose and shape parameters.

   3b. Update the texture and color transform parameters by the two-stage linear approach.

4. **Model Face Update:** Re-render I^{model} with the updated $\vec{\alpha}$ and $\vec{\beta}$.

Usually, the AMM search converges in less than 10 iterations. Its efficiency will be discussed in Section 3.

2.4. Model Training

In our implementation of AMM, a modified version of the bootstrapping algorithm [12] is used to train the Morphable Model. At each basic iterative step, instead of the stochastic gradient descent algorithm, AMM is applied to approximate each prototype image with the current model, which considerably accelerates the training phase. Secondly, through the bootstrapping iterations, the power preservation ratio of the shape and texture PCA models, instead of the number of principal components, is increased to enhance the expressive power of the model. For instance, for the 4 iterations, we can pick principal components so that 5%, 20%, 50% and 80% of the total power (variance) is retained, respectively. To control the bootstrapping process in this manner is more reasonable since it is independent of the number of the training samples.

2.5. Background Independence

Since in the novel image, the face to be matched may appear on a background that is very different to that of the training samples. In order to be a practical tool for face analysis, our model should be *background independent*. It can be noted that in a trained Morphable Model, all the shapes and textures are *vectorized*, i.e. pixel-wise aligned. Therefore, after regular Morphable Model training above, we apply a "mask", indicating the face region, to the aligned texture images and shape fields and obtain the masked versions. In this manner, the background information in the training images is entirely removed, without introducing any information losses to the faces, as shown in Figure 1.

In the AMM search phase, I^{model} is rendered by overlaying the "background-less" model face on the novel image (mapped onto the model frame), so that the required optical flow can be calculated reasonably. As a background-independent method, AMM can be used to analyze face images with arbitrary background.

(a) (b)

Figure 1. (a)Reference face image. (b)Face mask used to remove the background.

3. Experiments and Analysis

The dataset for our experiments contains 80 color images of frontal Asian faces. The faces are roughly normalized manually, with respect to scaling, rotation and translation. 60 out of the total 80 images are chosen as prototypes from which a Morphable Model is built, as discussed in Section 2.4 and 2.5.

3.1. Face matching with AMM

The AMM matching experiments on the test images show that the proposed method is efficient. A typical scenario of AMM search is shown in Figure 2. Usually the search converges in less than 10 iterations. It is not surprising because AMM effectively exploits the heuristic information provided by the novel image, and guides the search directly towards the optima of the model parameters, which makes AMM superior to a general optimization algorithm.

Since the parameter updating algorithm employed at each iteration is computationally efficient, typically the search ends in about 30 seconds on a Pentium 4 level PC. The optical flow calculation embedded in the AMM algorithm *seems* to be time-consuming, however, it can be noticed that in step **2b** the optical flow is calculated between two images within the *model frame*. That means even if the input novel image is as large as 1024x768, the optical flow should be calculated only between two 256x256 (which depends on the size of the model face) images. With the fast, gradient-based optical flow algorithm [13], this can easily be done in less than 200ms.

3.2. Robustness of AMM

It is necessary to point out that AMM works well even if the initial estimation of model parameters is bad, and can be used to match faces with large variations in translation, scaling and rotation. Figure 3 gives examples which show the insensitivity of AMM to the initial guess of face pose.

This may be explained from two aspects. First, the extraction of the heuristic information in the image is reliable, thanks to the effectiveness of the optical flow algorithm [13] adopted to recover the shape. Although a multi-resolution (coarse-to-fine) version of AMM is

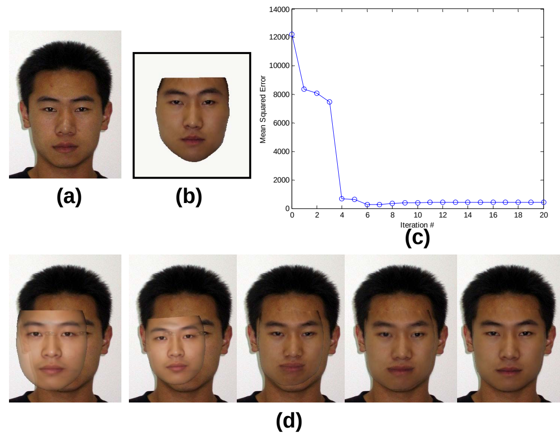

Figure 2. Typical scenario of AMM search. (a) Input novel face. (b)Matched model face. (c)Evolution curve of the pixel error. (d)The AMM search process.

not (actually need not to be) implemented explicitly, this hierarchical optical flow algorithm gives AMM a large search range in an implicit way. Second, the least-square method used to update the model parameters is trustworthy since it benefits from the redundancy of the information provided by tens of thousands of pixels (recall that in Morphable Model both shape and texture are dense).

To numerically evaluate the robustness of AMM to the initial estimation of translation, rotation and scaling, we systematically control the variation of the initial values of these parameters in a range around the ground-truth ($t_x, t_y \simeq 0$, $\theta \simeq 0^\circ$, $s \simeq 1$), and record the minimized pixel error achieved by AMM searches for each initial guess. The results are shown in Figure 4. The range around the ground truth with very small errors indicates the "capture range" of AMM searches: about -40~+40 pixels for translation (the model face image is about 300x300), -40~+40 degrees for rotation and 0.6~3.0 for scaling, which is superior to that of the stochastic gradient descent algorithm, as reported in [8].

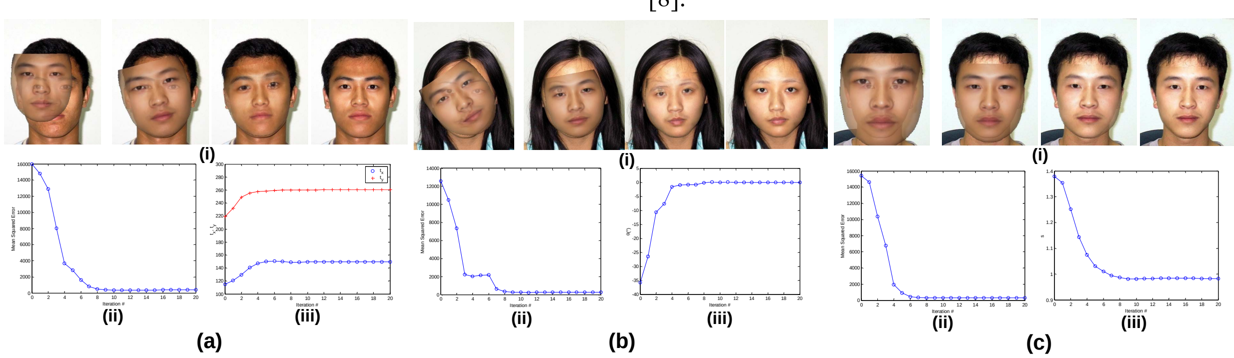

Figure 3. AMM's robustness to the initial estimation of pose. (a)Translation (b)Rotation and (c)Scaling. For each case:(i)The AMM search process (ii)Evolution curve of the pixel error (iii)Evolution curve of the relevant pose parameter(s).

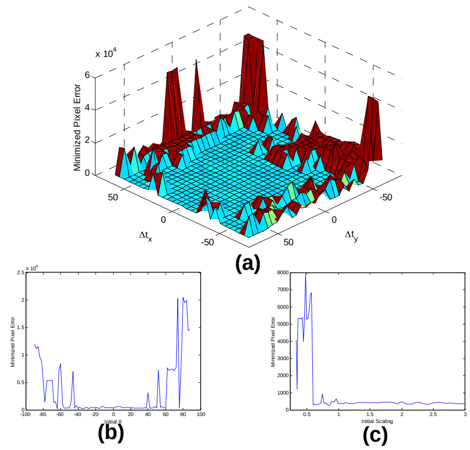

Figure 4. Robustness evaluation of AMM. (a)Translation. (b)Rotation. (c)Scaling.

4. Discussion

It may be helpful to compare AMM with other face analysis methods for a better understanding of its ideas.

In AAM, the updates of all parameters are predicted from image difference, which is not very efficient and results in relatively low convergent rate [7]. DAM [7] partly overcomes this drawback, updating texture parameter directly utilizing the image information, and then updating shape parameter under the assumption that shape is constrained by the texture it encloses. Sharing a similar idea with DAM, AMM updates texture parameter by directly minimizing the difference between model face and the input image. Additionally, AMM updates pose and shape parameters in a single step, from the dense correspondence, recovered by optical flow, between current model image and the novel image. In such manner, AMM efficiently utilizes the heuristic information provided by the novel image.

Another point which makes AMM different from AAM and DAM is that in AAM and DAM, shape and texture are correlated in some manner to reduce the dimensionality of the solution space. On the contrary, the Morphable Model assumes the independence between shape and texture, which gives it the additional expressiveness. Although this assumption increases the dimensionality of the solution space, the optimization scheme of AMM evades this difficulty by updating shape and texture parameters iteratively independent of each other.

ASM also updates shape and pose parameters through point correspondences. AMM is distinguished from ASM not only because shape is described by a dense flow instead of sparse landmark points. More importantly, ASM finds new suggested shape by independent search for each landmark point since it lacks a texture model for the whole face; whereas AMM recovers the suggested shape through optical flow, which utilizes the information provided by the texture, therefore constrains the estimated shape and avoids generating extremely eccentric shapes.

5. Conclusion

In this paper, we proposed Active Morphable Model (AMM), an efficient optimization method for face analysis with Morphable Model. AMM effectively utilizes the heuristic information provided by the input image, and updates all model parameters in a computationally efficient fashion. It has higher convergent rate and matching speed, and a larger capture range. As future work, we plan to apply AMM to practical face analysis problems, such as that in face recognition.

References

[1] David Beymer, "Vectorizing Face Images by Interleaving Shape and Texture Computations", A.I. Memo No. 1537, A.I. Lab, MIT, 1995

[2] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3D faces", *In SIGGRAPH'99 Conference Proceedings*, pages 187--194, 1999.

[3] T. F. Cootes, D. Cooper, C. J. Taylor and J. Graham, "Active Shape Models - Their Training and Application", *Computer Vision and Image Understanding*. Vol. 61, No. 1, Jan. 1995, pp. 38-59.

[4] T. F. Cootes G.J. Edwards and C. J. Taylor, "Active Appearance Models", *In Proceedings of ECCV'98*, v2, pp484-498, 1998.

[5] T. F. Cootes and C. J. Taylor, "Statistical Models of Appearance for Computer Vision", Technical Report, University of Manchester, 2001

[6] Ezzat, T., G. Geiger and T. Poggio. Trainable Videorealistic Speech Animation. *In: Proceedings of ACM SIGGRAPH 2002*, San Antonio, Texas, pp388-398, 2002.

[7] X. Hou, S.Z. Li, H. Zhang, Q. Cheng, "Direct Appearance Models", *In Proceedings of CVPR 2001*

[8] M. J. Jones, "Multidimensional Morphable Models: A Framework for Representing and Matching Object Classes", PhD Thesis, Massachusetts Institute of Technology, 1997

[9] M. J. Jones and T. Poggio, "Multidimensional Morphable Models", *In Proc. of the International Conference on Computer Vision,* pages 683-688, 1998

[10] Sami Romdhani, Volker Blanz, and Thomas Vetter, "Face Identification by Fitting a 3D Morphable Model using Linear Shape and Texture Error Functions", *In Proc. ECCV'2002*

[11] T. D. Rikert and M. J. Jones. "Gaze Estimation using Morphable Models". *In Proc. of the 3rd Intl. Conf. on Automatic Face and Gesture Recognition*, Nara, Japan, 1998

[12] T. Vetter, M. J. Jones and T. Poggio, "A Bootstrapping Algorithm For Learning Linear Models Of Object Classes", *In Proc. CVPR '97*, pp40-46, 1997

[13] J.R. Bergen, P. Anandan, K. J. Hanna and R. Hingorani, "Hierarchical Motion-based Motion Estimation", *In Proc. ECCV'92*

[14] B. D. Lucas and T. Kanade, "An Iterative Image Registration Technique with an Application to Stereo Vision", *In Proc. of Image Understanding Workshop*, pp.121-130, 1981