

Robust 3D Morphable Model Fitting by Sparse SIFT Flow

Xiangyu Zhu, Dong Yi, Zhen Lei and Stan Z. Li

Center for Biometrics and Security Research & National Laboratory of Pattern Recognition
Institute of Automation, Chinese Academy of Sciences
95 Zhongguancun East Road, Haidian District, Beijing, China.

Abstract—3D Morphable Model (3DMM) has been widely used in face analysis for many years. The most challenging part of 3DMM is to find the correspondences between 3D points and 2D pixels. Existing methods only use keypoints, edges, specular highlights and image pixels to complete the task, which are not accurate or robust. This paper proposes a new algorithm called Sparse SIFT Flow (SSF) to improve the reconstruction accuracy. We mark a set of salient points to control the shape of facial components and use SSF to find their corresponding pixels on the input image. We also incorporate SSF into Multi-Features Framework to construct a robust 3DMM fitting algorithm. Compared with the state-of-the art, our approach significantly improves the fitting results in facial component area.

I. INTRODUCTION

In the research area of face analysis, estimating 3D shape from image data has attracted considerable attentions in recent years. Based on the 3D shape of a face, it is easy to estimate its pose and illumination, which are important for face recognition in the wild [1, 2]. Besides, 3D shape provides many novel approaches for image processing problems like relighting [3] and super-resolution [4].

Given a single face image under unknown pose and illumination, 3D Morphable Model (3DMM) [2, 5–7] could solve its 3D shape, texture, pose and illumination parameters at the same time. These parameters are estimated by fitting 3DMM to the input image following analysis-by-synthesis framework, where Stochastic Newton Optimization is applied to minimize the difference between the synthetic image and the input image. This fitting process is ill-posed and suffers from local minimum problem seriously. Romdhani [8] presents a Multi-Features Framework (MFF) to handle the local minima in which contours, textured edges, specular highlights and pixel intensity are considered jointly. The objective function of MFF is the weighted sum of the cost functions produced by each feature. Zhang [9] and Aldrian [10, 11] concentrate on simplified linear illumination models. In their works, human faces are regarded as Lambertian objects and a 9 dimensional linear subspace called Spherical Harmonic Reflectance is used to describe illumination on human faces. In addition, SUV color subspace and high-order spherical harmonic bases are applied to handle specular reflection.

The central problem of 3DMM fitting is to find the correspondences between 3D points and image pixels based on some features. Until now, the most popular features are keypoints [11], pixel intensity [2] and edges [8]. The keypoint

feature directly provides sparse correspondences, which make the fitting procedure convex and robust. However, the performance of keypoint fitting relies on the accuracy of keypoint detection and the fitted shape is always inaccurate due to the severe sparsity. The pixel intensity feature could provide dense correspondences by fitting the synthetic image to the input image, but the correspondences are unstable and easily influenced by expression and occlusion. Edge is another feature used to get sparse correspondences. A set of edge points are marked on the model by hand and they are fitted to their closest Canny edges on the input image. Considering the Canny operator cannot detect all edges stably and the closest point criterion may get wrong correspondences, the feature is also unstable. In many cases, all these features are used together to compensate each other in MFF, but none of them is robust enough for facial component fitting.

In this paper, a more discriminative feature and its robust fitting algorithm, SIFT flow [12], is applied under Multi-Features Framework. SIFT feature is used to describe face images and neighborhood constraints are used to restrict flows of neighboring points. To speed up the fitting process, we only apply SIFT flow on the facial component region. According to [13], facial components are more important than other regions both to human perception and face recognition system. Experimental results show that our algorithm achieves more accurate facial component shapes than existing 3DMM across a wide range of illumination and pose conditions.

II. RELATED WORK

A. 3D Morphable Model

3D Morphable Model [5] is constructed from a set of 3D face scans in dense correspondence. Each scan is represented by a shape-vector $S = (x_1, y_1, z_1, x_2, y_2, z_2, \dots, x_n, y_n, z_n)$ and a texture-vector $T = (r_1, g_1, b_1, r_2, g_2, b_2, \dots, r_n, g_n, b_n)$, which contain the 3d coordinate and color of each point. PCA is applied to decorrelate texture and shape vectors respectively and a 3D face can be described as:

$$S = \bar{s} + \sum_{i=1}^{m-1} \alpha_i \cdot s_i \quad T = \bar{t} + \sum_{i=1}^{m-1} \beta_i \cdot t_i \quad (1)$$

where m is the number of face scans. \bar{s} and \bar{t} are the means of shape and texture respectively. s_i and t_i are the i th eigenvector. $\alpha = (\alpha_1, \alpha_2, \dots, \alpha_{m-1})$ and $\beta = (\beta_1, \beta_2, \dots, \beta_{m-1})$ are shape and texture parameters determining S and T .

After the 3D morphable model has been generated, the Phong illumination model and the weak perspective projection

Stan Z. Li is the corresponding author

are used to produce a synthetic image. The fitting process of 3DMM is to minimize the Euclidean distance between the synthetic and input image:

$$E = \left\| \sum_{x,y} I_{input}(x,y) - I_{syn}(x,y) \right\| \quad (2)$$

where E is the cost function, $I_{input}(x,y)$ is the input image and $I_{syn}(x,y)$ is the image synthesized by projecting the 3D face into image plane. Stochastic Newton method is adopted to minimize the cost function.

The fitting algorithm in [5] only uses the pixel intensity feature. As we know, pixel intensities of a face image can be dramatically influenced by expression, occlusion and complicated illumination. Therefore, the 3DMM is not robust enough to these variations and suffers from local minimum problem seriously. Moreover, the texture description ability of 3DMM is limited due to its low dimensionality. As a result, the synthetic image is always smooth, which makes the fitting procedure unstable if the input face is rich in texture.

B. Multi-Features Framework

To address the limitation of pixel intensity, a Multi-Features Framework (MFF) [8] was proposed. With multiple features, the cost function of MFF is smoother and hence the global optimal solution is easier to be obtained in optimization process.

In the original MFF, three other features including specular highlights, texture constraint and edges are incorporated. Specular highlights imply the relationship between lighting direction and surface normal. Texture constraint is used to separate the contributions of texture from lighting in pixel intensities. The edge feature provides shape information robust to illumination, and both contours and texture edges are utilized. The contours are generated from the overlapping of 3D face and background and the textured edges are defined on the edge and corner of facial components. In the edge fitting process, the edge points marked on the model are fitted to the Canny edges on the image based on the closest point criterion:

$$k(i) = \arg \min_{j=1, \dots, J} \|q_j^e - p_i\| \quad (3)$$

where $k(i)$ is the cost produced by an edge point. p_i is a model edge point. q_j^e is the closest point of p_i on the image. J is the number of image edge points.

Textured edges, which are specially designed for facial component fitting, are very important to the fitting result. However, according to the optimization procedure in [14], the textured edge feature is only used in the first several iterations. The main reason is that textured edges cannot be detected by Canny operator reliably. Moreover, the closest point criterion is not always reasonable to build the correspondences between p_i and q_j^e , which may provide a wrong guide for fitting.

III. SPARSE SIFT FLOW FITTING

In this section, a new fitting algorithm called Sparse SIFT Flow (SSF) is utilized and we pay special attention on facial

component fitting. As in [2], generating a synthetic face image needs the projection:

$$s_{2d} = fPR(\bar{s} + \sum_{i=1}^{m-1} \alpha_i \cdot s_i + t_{3d}) \quad (4)$$

where s_{2d} is the image coordinates of 3D points after projection. f is the scale parameter. P is the orthographic projection matrix. R is the rotation matrix with ϕ, γ, θ as the rotation angles for x, y, z axes. t_{3d} is the translation vector in 3D space and $(\bar{s} + \sum_{i=1}^{m-1} \alpha_i \cdot s_i)$ is the shape vector as (1) where α is the shape parameter. If there exists a vector s_{2d_t} that contains the true image coordinates of 3D points, the pose and shape parameters could be estimated by minimizing the distances of s_{2d} and s_{2d_t} as (5). The central problem will be how to get the coordinates s_{2d_t} .

$$\arg \min_{f, \phi, \gamma, \theta, t_{3d}, \alpha} \|s_{2d_t} - s_{2d}\| \quad (5)$$

3DMM fitting always needs hundreds of iterations and estimating s_{2d_t} for each point is very time-consuming especially by complicated flow algorithm. We hope to just fit a sparse set of points instead of the whole face, while keep the fitting result as accurate as possible. In our algorithm, three steps are involved : 1. Marking a set of salient points to control the facial component shape. 2. Matching the salient points to the input image using SIFT flow. 3. Minimizing the Euclidean distances of corresponding points.

A. Salient Points

Facial components are the most important part for face recognition system [13], thus we hope to get more accurate component shapes in our algorithm. We mark a set of salient points on the 3D face model to control the shape of facial components, as shown in Figure 1. These salient points are located on four regions: eyebrow, eye, nostril and mouth, which are all texture-rich regions.

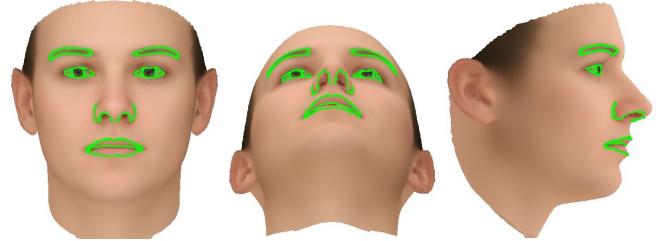


Fig. 1. Salient Points on 3D Morphable Model

With these points marked, we restrict (4) and (5) only on salient points and get a new cost function:

$$\begin{aligned} & \arg \min_{f, \phi, \gamma, \theta, t_{3d}, \alpha} \|s_{2d_sa_t} - s_{2d_sa}\| \\ & s_{2d_sa} = fPR(\bar{s}_{sa} + \sum_{i=1}^{m-1} \alpha_i \cdot s_{sa_i} + t_{3d}) \end{aligned} \quad (6)$$

where \bar{s}_{sa} and s_{sa_i} are the mean and eigenvector on salient points. s_{2d_sa} is the positions of salient points after projection and $s_{2d_sa_t}$ is the true positions. Because salient points are only

a small portion of the whole face, estimating of $s_{2d_sa_t}$ is much faster than s_{2d_t} and more sophisticated feature extraction and fitting algorithm can be applied without consuming a lot of resources.

B. Sparse SIFT Flow Fitting

SIFT Flow [12] is used to estimate $s_{2d_sa_t}$, it is an image alignment algorithm based on SIFT descriptor and could provide dense, pixel-to-pixel correspondences between two images. In the algorithm, a SIFT descriptor is computed for each pixel to form a SIFT image and the SIFT image is aligned with two criteria: the descriptors are matched and the flow field is smooth.

A novel shape fitting method called Sparse SIFT Flow (SSF) is proposed here. The flowchart of SSF is shown in Figure 2. With the initial pose estimated with a set of

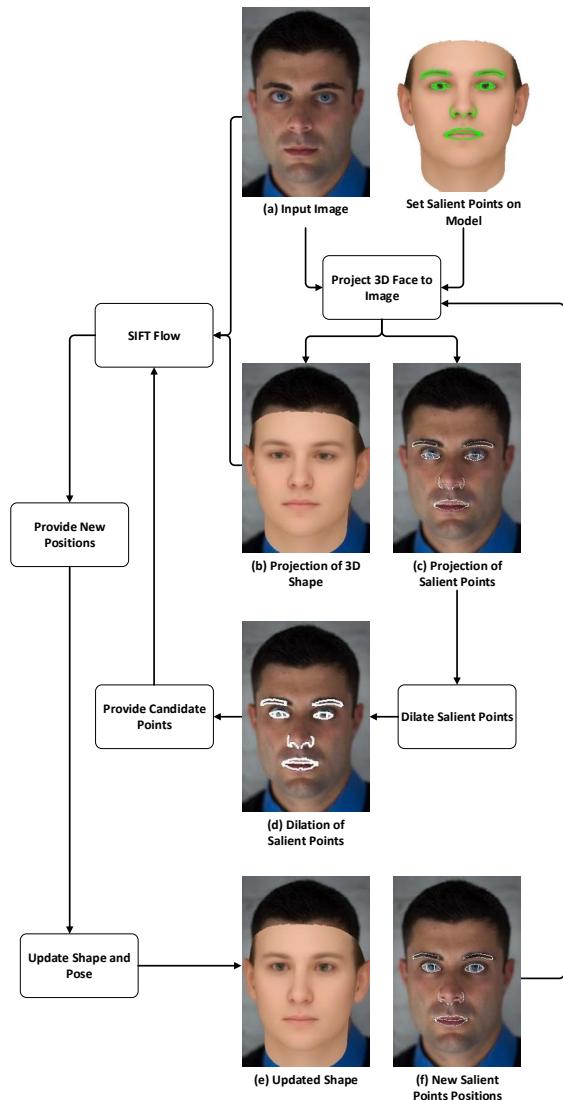


Fig. 2. The flow chart of SSF. (a): input face image. (b): projection of the mean face. (c): the initial positions of salient points. (d): dilation of salient points. (e): updated shape after an iteration. (f): positions of salient points after an iteration

keypoints, we project the mean 3D face to the input image, as

shown in Figure 2(b). Figure 2(c) shows the s_{2d_sa} of salient points after projection. Noted that the neighborhood relations of salient points are not guaranteed because of coordinate discretization, a dilate morphological operator is applied to make sure salient points are connected to their neighbors, as in Figure 2(d). These white points after dilation are candidate points.

The SIFT flow algorithm will run from Figure 2(b) to Figure 2(a) only on candidate points. The cost function of the alignment is defined as:

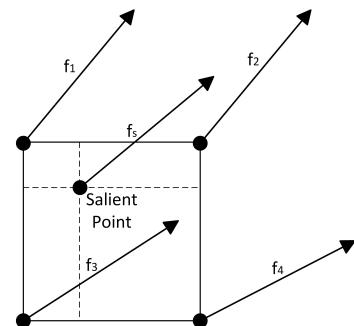
$$E(w) = \sum_p \min(|s_1(p) - s_2(p + w(p))|, t) \quad (7a)$$

$$+ \sum_p \eta(|u(p)| + |v(p)|) \quad (7b)$$

$$+ \sum_{(p,q) \in \epsilon} (\min(\alpha|u(p) - u(q)|, d) + \min(\alpha|v(p) - v(q)|, d)) \quad (7c)$$

where $p = (x, y)$ is the coordinate of a candidate point. $w(p) = (u(p), v(p))$ is the flow vector of p . s_1 and s_2 are the SIFT images of Figure 2(b) and Figure 2(a). ϵ contains all 4-neighborhoods of candidate points. This cost function contains three terms: (7a) is the data term which makes the SIFT descriptors matched along the flow $w(p)$; (7b) is the small displacement term which prevents flows from becoming too large; (7c) is the smoothness term which keeps the flows in neighborhood as similar as possible. The truncated L1 norm is used to deal with discontinuities along object boundaries, with d and t as thresholds respectively. We adopt dual loopy belief propagation and coarse-to-fine matching scheme [12] to optimize the object function. After the algorithm converges, a flow chart of candidate points will be obtained. The algorithm is called “sparse” because the SIFT flow runs only on a sparse set of points.

Considering salient points are not always located on image grids exactly, a bilinear interpolation is used to estimate the flows of salient points $flow_{sa}$. As shown in Figure 3, the translation of a salient point is the weighted sum of its neighbors on the grid. The estimated $flow_{sa}$ is then used to



$$f_s = w_1 * f_1 + w_2 * f_2 + w_3 * f_3 + w_4 * f_4$$

Fig. 3. The bilinear interpolation of SIFT flow, where f_1, f_2, f_3, f_4 are the flows of neighboring points on the grid, w_1, w_2, w_3, w_4 are the corresponding bilinear weights.

get true positions of salient points s_{2d_sat} :

$$s_{2d_sat} = s_{2d_sa} + flow_{sa} \quad (8)$$

After the s_{2d_sat} has estimated, a Levenberg-Marquardt optimization is used to update shape and pose parameters by (6). The fitting result after an iteration is shown in Figure 2(e) and Figure 2(f). Then the projection and fitting process will run another iteration until convergence.

IV. MULTI-FEATURES FRAMEWORK WITH SSF

A more sophisticated approach is incorporating SSF into Multi-Features Framework (MFF). We choose keypoints, contours, pixel intensity and SSF as features in our framework. Compared with Romdhani's method [8], our approach uses SSF instead of the non-robust textured edges to constrain facial component fitting. Moreover, a multi-stage fitting scheme is adopted: in each stage, different parameters are optimized with different features with Levenberg-Marquardt algorithm.

Our fitting process is divided into 6 stages: In the 1st stage, pose parameters are initialized using keypoints on the input image. In the 2nd stage, pose and shape are coarsely estimated with keypoints and contours. After the initialization of pose and shape, in the 3rd stage illumination and texture are updated with pixel intensity. In the 4th stage is an integrated fitting procedure, in which all the parameters are updated using contours, pixel intensity and SSF. In the 5th stage a segmented fitting is adopted, where the 3D face model is divided into 4 subregions (eye, nose, mouth and cheek) and they are fitted respectively. In the last stage, when parameters are close to their optimal values, we use SSF to refine the shape of facial components. Table I lists the 6 stages of our algorithm, where "X" means used, "few" means that only the first few PCA coefficients are updated and "seg" means segmented fitting. Figure 4 shows the fitting result after each stage.

TABLE I. FEATURES AND PARAMETERS USED AT EACH STAGE OF OUR ALGORITHM

Stg	Features				Parameters				Iters
	Key	Cont	Pixel	SSF	Pose	Shape	Illum	Tex	
1	X				X				6
2	X	X			X	few			20
3			X				X	few	30
4		X	X	X	X	X	X	X	20
5		X	X	X		seg		seg	15
6			X			seg			15

V. EXPERIMENT

We use the Basel Face Model (BFM) [15] database which provides a 3DMM constructed by 200 face scans and a testing set including 270 images. The testing set is built by rendering 10 face scans under 9 pose conditions (from frontal to profile) in 3 lighting directions (from frontal to side), giving 270 renderings in total. For each rendering image, 7 keypoints on eye corner, mouth corner, nose tip and earlap are provided.

We use the weak perspective projection containing scaling, rotation and translation to describe pose and the Phong illumination model to describe ambient, diffuse and specular reflectance. In the fitting process, the 7 keypoints are utilized to initialize pose parameters with the method in [16], the Multi-Features Framework with SSF is then used to optimize pose,

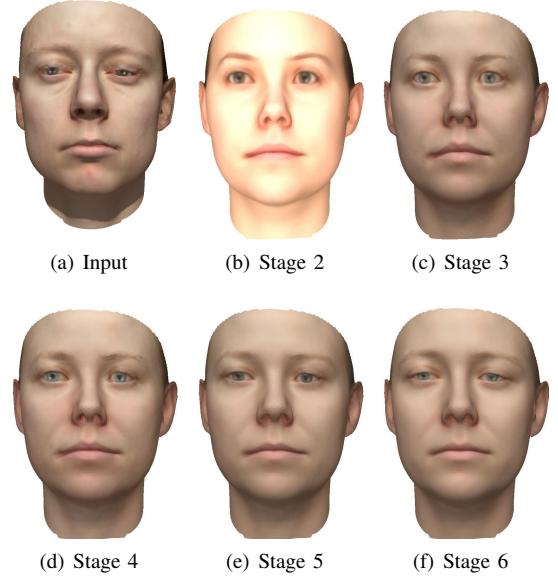


Fig. 4. Fitting result after each stage. (a) input face image; (b) the result of pose and shape initialization after stage 2; (c) the result of illumination and texture initialization after stage 3; (d) the result of integrated fitting after stage 4; (e) the result of segmented fitting after stage 5; (f) the final result

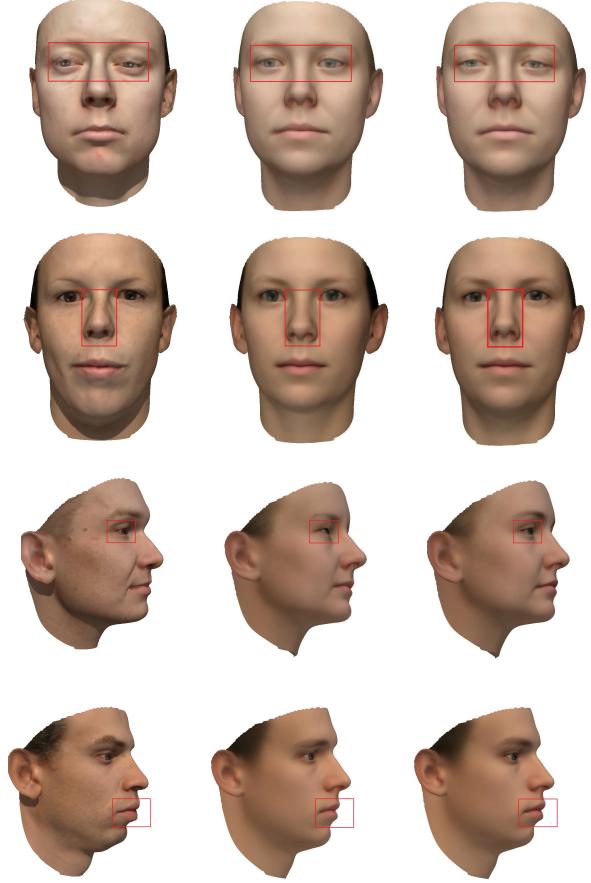


Fig. 5. Left column: input images. Middle column: reconstruction results of Romdhani's MFF. Right column: reconstruction results of ours. Rectangles show the better reconstructed regions of our method

shape, illumination and texture parameters. To be robust, only the 90 most significant PCA coefficients are used.

We compare our method with the state-of-the-art Romdhani's MFF algorithm in [8]. Figure 5 shows the comparison experiment. It can be seen from the rectangles that we reconstruct eyes, nose and mouth better than MFF. In the 6-stage fitting process, we find the fitting results after stage 5 are almost similar with the results of MFF. But in the stage 6 when SSF is used only to refine facial components, the fitted shape improves a lot in a few iterations, just as the differences between Figure 4(e) and Figure 4(f), the eyes are closer due to the fitting of SSF. Figure 6 shows the fitting results of Face No.9 in Illumination No.0 (frontal lighting) and Face No.52 in Illumination No.30 (side lighting) under all pose conditions. Figure 7 shows some fitting results of real-world images. It can be seen that our algorithm also performs well on real-world data, even for cartoon characters.



Fig. 7. The first row: the input real-world images. The second row: the corresponding fitting results.

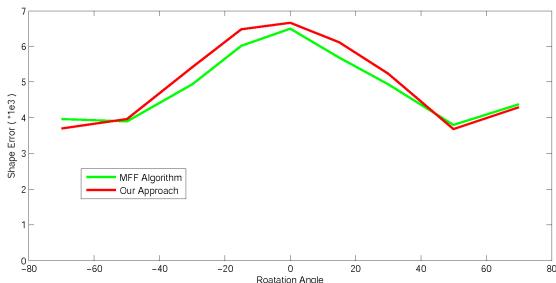


Fig. 8. Root Mean Square Error of the entire face over the rotations in (-70, -50, -30, -15, 0, 15, 30, 50, 70) angles

We also evaluate the precision of shape fitting quantitatively. Given a reconstructed shape and its ground truth, we firstly scale and translate the reconstructed shape to eliminate the error brought by rigid transformation and then measure the Euclidean distances of corresponding points. Root Mean Square Error (RMSE) is used to measure the errors of fitted shapes. Because facial components are more important in many applications, we also measure the RMSE of eyes, nose and mouth, the vertices of which are defined in BFM. Figure 8 shows the RMSE of all face points in each pose. We can see that in the whole face area, our algorithm performs comparable with Romdhani's MFF. While in the facial component

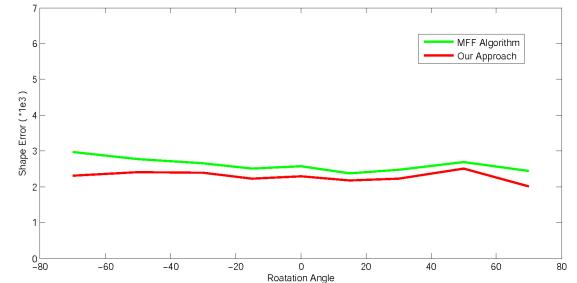


Fig. 9. Root Mean Square Error in facial component area over the rotations in (-70, -50, -30, -15, 0, 15, 30, 50, 70) angles

region, as shown in Figure 9, our method outperforms the MFF owing to the contributions of SSF. Table II shows the RMSE in facial component area of all subjects in each pose and illumination condition. There is a 12.42% improvement overall, which illustrates the accuracy and robustness of our approach.

VI. CONCLUSION

In this paper, a novel robust and accurate shape fitting algorithm using SIFT flow was proposed. We marked a set of salient points around facial components, and fitted them to the input image with a sparse SIFT flow algorithm. We also incorporated SSF into Multi-Features Framework to construct a robust 3DMM fitting algorithm. The accuracy and robustness was compared to the state-of-the-art Romdhani's MFF. Experiment showed that our method could get more accurate and robust facial components under all illumination and pose conditions. Our future work will involve in using 3D Morphable Model and SSF to fit faces in the wild which have complicated poses, illuminations and expressions.

ACKNOWLEDGMENT

This work was supported by the Chinese National Natural Science Foundation Projects #61105023, #61103156, #61105037, #61203267, #61375037, National Science and Technology Support Program Project #2013BAK02B01, Chinese Academy of Sciences Project No. KGZD-EW-102-2, Jiangsu Science and Technology Support Program Project #BE2012627, and AuthenMetric R&D Funds.

REFERENCES

- [1] V. Blanz, S. Romdhani, and T. Vetter, "Face identification across different poses and illuminations with a 3d morphable model," in *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE, 2002, pp. 192–197.
- [2] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 25, no. 9, pp. 1063–1074, 2003.
- [3] L. Zhang, Y. Wang, S. Wang, D. Samaras, S. Zhang, and P. Huang, "Image-driven re-targeting and relighting of facial expressions," in *Computer Graphics International 2005*. IEEE, 2005, pp. 11–18.
- [4] P. Mortazavian, J. Kittler, and W. Christmas, "A 3-d assisted generative model for facial texture super-resolution," in *Biometrics: Theory, Applications, and Systems, 2009. BTAS'09. IEEE 3rd International Conference on*. IEEE, 2009, pp. 1–7.

TABLE II. THE RMSE OF ALL THE FACES IN EACH POSE AND ILLUMINATION CONDITION

Illum	0			30			50			All Illum		
Pose	MFF	Ours	Improvement	MFF	Ours	Improvement	MFF	Ours	Improvement	MFF	Ours	Improvement
-70	2.42	2.03	0.38(15.85%)	2.53	2.07	0.46(18.07%)	2.37	1.95	0.42(17.82%)	2.98	2.31	0.66(22.31%)
-50	2.43	2.24	0.19(7.91%)	2.44	2.20	0.24(9.77%)	2.53	2.25	0.29(11.41%)	2.77	2.41	0.36(12.92%)
-30	2.67	2.36	0.32(11.80%)	2.55	2.28	0.26(10.28%)	2.52	2.25	0.27(10.81%)	2.65	2.39	0.26(9.94%)
-15	2.83	2.38	0.45(15.78%)	2.70	2.41	0.30(10.94%)	2.43	2.38	0.05(2.04%)	2.51	2.22	0.29(11.69%)
0	2.79	2.29	0.50(17.92%)	2.96	2.24	0.72(24.34%)	3.19	2.41	0.77(24.25%)	2.58	2.30	0.28(10.98%)
15	2.61	2.37	0.25(9.44%)	2.89	2.42	0.47(16.42%)	2.79	2.44	0.35(12.56%)	2.38	2.18	0.20(8.50%)
30	2.41	2.22	0.19(8.00%)	2.53	2.22	0.30(12.04%)	2.60	2.21	0.39(14.82%)	2.47	2.23	0.24(9.72%)
50	2.53	2.25	0.27(10.74%)	2.25	2.12	0.13(5.92%)	2.36	2.16	0.20(8.58%)	2.69	2.50	0.19(7.02%)
70	2.52	2.37	0.14(5.73%)	2.79	2.63	0.17(5.91%)	2.75	2.50	0.26(9.31%)	2.44	2.02	0.42(17.26%)
All Pose	2.58	2.28	0.30(11.60%)	2.60	2.29	0.34(12.90%)	2.62	2.28	0.33(12.82%)	2.61	2.28	0.32(12.42%)

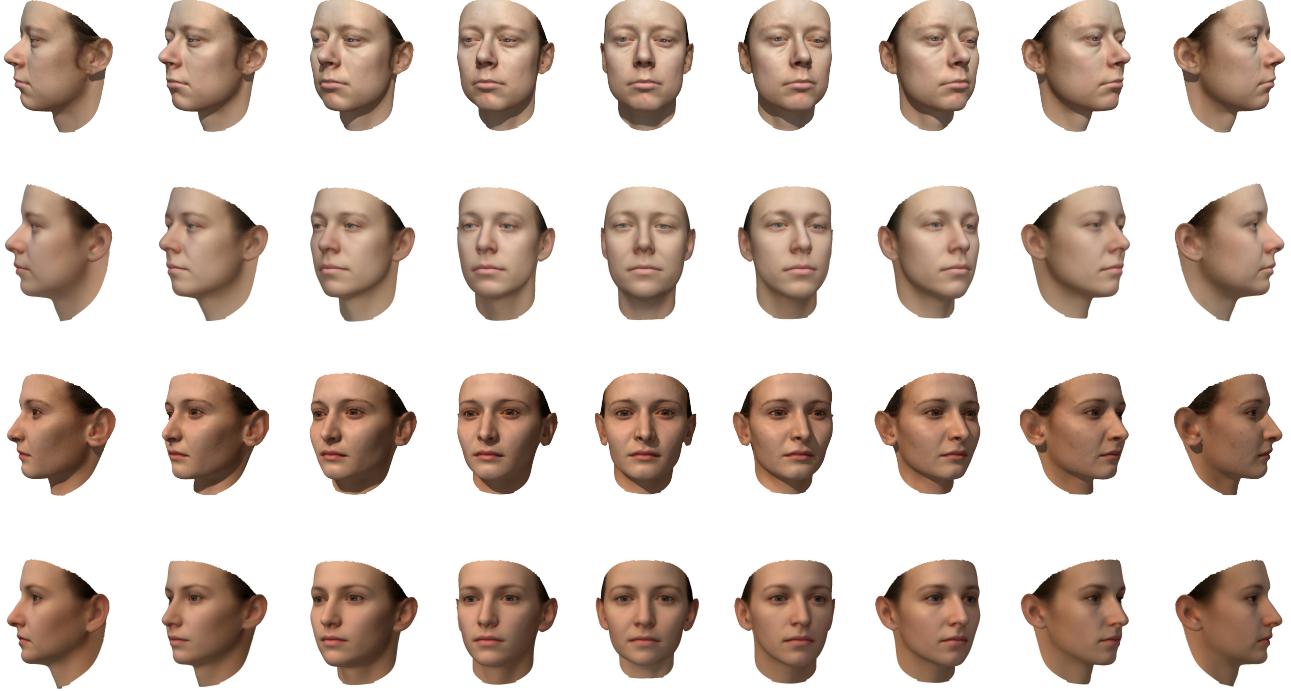


Fig. 6. The first row: the input images of Face No.17 in Illumination 0. The second row: the corresponding fitting results of the first row by our method. The third row: the input images of Face No.52 in Illumination 30. The fourth row: the corresponding fitting results of the third row by our method.

- [5] V. Blanz and T. Vetter, “A morphable model for the synthesis of 3d faces,” in *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*. ACM Press/Addison-Wesley Publishing Co., 1999, pp. 187–194.
- [6] S. Romdhani, V. Blanz, and T. Vetter, “Face identification by fitting a 3d morphable model using linear shape and texture error functions,” in *ECCV (4)*, vol. 2353. Springer, 2002, pp. 3–19.
- [7] S. Romdhani and T. Vetter, “Efficient, robust and accurate fitting of a 3d morphable model,” in *Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on*. IEEE, 2003, pp. 59–66.
- [8] ———, “Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior,” in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, vol. 2. IEEE, 2005, pp. 986–993.
- [9] L. Zhang and D. Samaras, “Face recognition from a single training image under arbitrary unknown lighting using spherical harmonics,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 28, no. 3, pp. 351–363, 2006.
- [10] O. Aldrian and W. Smith, “Inverse rendering of faces with a 3d morphable model,” 2013.
- [11] O. Aldrian and W. A. Smith, “A linear approach of 3d face shape and texture recovery using a 3d morphable model,” in *Proceedings of the British Machine Vision Conference*, pages, 2010, pp. 75–1.
- [12] C. Liu, J. Yuen, and A. Torralba, “Sift flow: Dense correspondence across scenes and its applications,” *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 5, pp. 978–994, 2011.
- [13] K. Bonnen, B. Klare, and A. Jain, “Component-based representation in automated face recognition,” 2013.
- [14] S. Romdhani, “Face image analysis using a multiple features fitting strategy,” Ph.D. dissertation, University of Basel, 2005.
- [15] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, “A 3d face model for pose and illumination invariant face recognition,” in *Advanced Video and Signal Based Surveillance, 2009. AVSS’09. Sixth IEEE International Conference on*. IEEE, 2009, pp. 296–301.
- [16] A. M. Bruckstein, R. J. Holt, T. S. Huang, and A. N. Netravali, “Optimum fiducials under weak perspective projection,” *International Journal of Computer Vision*, vol. 35, no. 3, pp. 223–244, 1999.