3DMM Estimation from Highly Distorted Images

Ananta Bhattarai
Technical University of Munich
Munich, Germany
ananta.bhattarai@tum.de

Magdy Mahmoud
Technical University of Munich
Munich, Germany
magdy.mahmoud@tum.de

Maqarios Saleh
Technical University of Munich
Munich, Germany
maqarios.saleh@tum.de

Abstract

In this paper, we estimate a 3D Morphable Model (3DMM) from a highly distorted image i.e. 360-degree fisheye image. Our method first applies the fisheye distortion to MICC Florence dataset face images and then fits 3DMM to the distorted image by jointly optimizing for parametric face model parameters, the illumination parameters, the rigid transformation, the camera parameters, and the distortion parameters. We also compare the estimated mesh with the ground truth mesh and study the effect of including distortion parameters in the optimization.

1. Motivation and Idea

Recovering the 3D model of the human face from a single 2D image is a challenging task and has many applications such as facial animation [9] and face recognition [6]. In recent years, there is a rise of interest in image-based reconstruction using the 3D Morphable Model (3DMM) [5] [22] [20]. Reconstructing the 3D face model from the image involves estimating the coefficients of the 3DMM that can best explain the observation. Modern techniques use deep Convolutional Neural Networks (CNN) to predict 3DMM coefficients given the input image [11] [12] whereas traditional methods solve this by minimizing the difference between the observed image and the synthesis of an estimated 3D face [5] [20]. Lately, numerous methods have been proposed for an accurate 3D face reconstruction from a single RGB image [11] [21] [18]. However, to the best of our knowledge, face reconstruction in 360° images has not been explored yet. 360° cameras offer the possibility to capture everything around itself and they are often used in panoramic photography and robotics. However, such cameras make computer vision problems more complex due to the distortions introduced by their lenses.

Our goal in this paper is to address face reconstruction in 360° fisheye images. To this end, we first apply fisheye-like distortion to MICC Florence face images [3] as described in Section 3.1. Then, we fit Basel Face Model (BFM-2019) [15] to the fisheye image using the analysis-by-synthesis technique, jointly optimizing for parametric face model coefficients, the illumination parameters, the rigid transformation, the camera parameters, and the distortion coefficients.

2. Related Work

Analysis-by-synthesis techniques have been explored to fit 3DMM to multiple image modalities. This ranges from using multi-view to monocular images and videos. Although multi-view methods produce high-quality reconstructions, acquiring such data is time-consuming and requires expensive setups. Therefore, a lot of recent research focuses on using a single image to obtain similar quality reconstructions.

Estimating 3DMM coefficients from a single RGB image is a most challenging scenario and an extremely ill-posed problem. The early work of Blanz and Vetter [5] proposed an analysis-by-synthesis solution for fitting 3DMM to a single image. However, their approach requires manual initialization for the optimization problem. In recent years, several robust approaches have been proposed to make reconstruction automatic [2] [4] [19] [13]. Our approach is similar to that of [5]. In addition, we first minimize the landmarks-based energy for the initialization of the parameters and account for distortion coefficients in the optimization.

Face detection in 360° images has been investigated by Jianglin Fu *et al* [14]. They create a fisheye-looking version of FDDB [16] and train a face detector on it to detect the face in fisheye images. Therefore, we adopt the simi-

lar approach outlined in the paper [14] to apply fisheye-like distortion to the face images.

3. Method

3.1. Fisheye Distortion

We apply fisheye-like distortion to the face images from the MICC Florence dataset [3]. We use the same formula as used in [14] to distort the original image. At first, the square or rectangle image is mapped to the circular image using Equation 1. Here, (x, y) is the normalized coordinates of the image, such that image center is $(0, 0)$ and the four corners have coordinates $(\pm 1, \pm 1)$. This form of distortion introduces radial distortion.

$$(x', y') = \left(x\sqrt{1 - \frac{y^2}{2}}, y\sqrt{1 - \frac{x^2}{2}} \right) \tag{1}$$

$$(x'', y'') = \left(x' e^{-\frac{r^2}{4}}, y' e^{-\frac{r^2}{4}} \right) \tag{2}$$

To add barrel distortion, we further apply Equation 2 to the distorted coordinates, where $r = r_s\sqrt{(x')^2 + (y')^2}$. We use r_s to scale the distortion and set it to 1.3 in our experiments. The effect of fisheye distortion is shown in Figure 1.

3.2. Landmark Detection

We extract facial landmarks from the fisheye image to match the landmarks with the 3D model and perform the optimization. The key landmark points normally include the facial regions like the nose tip, eye corner, eyebrows, and chin tip. With the recent advances in deep learning, the problem of landmark detection in 2D images has largely been solved. Some examples of these works are [17] and [10]. In this paper, we use the work of [8] to detect 68 landmarks of the face. An example of landmarks detection is shown in Figure 2.

3.3. 3DMM Fitting

We use (BFM-2019) [15] and fit it to the given 360° fisheye image. The first two dimensions of the parametric face model represent shape, and texture respectively. Therefore, the resulting parametric face model is

$$\mathcal{M}_{geo}(\boldsymbol{\alpha}) = \boldsymbol{a}_{shape} + E_{shape} \cdot \boldsymbol{\alpha} \tag{3}$$

$$\mathcal{M}_{tex}(\boldsymbol{\beta}) = \boldsymbol{a}_{tex} + E_{tex} \cdot \boldsymbol{\beta} \tag{4}$$

This prior assumes a multivariate normal probability distribution of shape and texture around the mean shape $\boldsymbol{a}_{shape} \in \mathbb{R}^{3n}$ and texture $\boldsymbol{a}_{tex} \in \mathbb{R}^{3n}$. The shape $E_{shape} \in \mathbb{R}^{3n\times 199}$, and texture $E_{tex} \in \mathbb{R}^{3n\times 199}$ PCA basis and the corresponding standard deviations $\sigma_{id} \in \mathbb{R}^{3n\times 199}$, and $\sigma_{tex} \in \mathbb{R}^{3n\times 199}$ are given. There are 47439 vertices and 94464 faces in the model. New faces are synthesized from the model as a linear combination of principal components. Fitting the model to given RGB image involves optimization of the shape $\boldsymbol{\alpha}$ and texture $\boldsymbol{\beta}$ coefficients along with a set of rendering parameters in an analysis-by-synthesis loop. Rendering parameters include the illumination parameters $\boldsymbol{\gamma}$, the rigid transformation $\boldsymbol{R}$, $\boldsymbol{T}$, the camera parameters $\boldsymbol{K}$, and the distortion parameters $\boldsymbol{D}$. To account for the lens distortion, we use OpenCV [7] fisheye camera model.

Our optimization objective includes three energy terms

$$E(\boldsymbol{\mathcal{P}}) = \underbrace{E_{col}(\boldsymbol{\mathcal{P}}) + E_{lan}(\boldsymbol{\mathcal{P}})}_{data} + \underbrace{E_{reg}(\boldsymbol{\mathcal{P}})}_{prior} \tag{5}$$

where $E_{col}(\boldsymbol{\mathcal{P}})$ is the photo consistency error between observed image and the model, $E_{lan}(\boldsymbol{\mathcal{P}})$ is the landmarks feature similarity in observed image and the model and $E_{reg}(\boldsymbol{\mathcal{P}})$ is the regularization term. E_{col} is given by

$$E_{col}(\boldsymbol{\mathcal{P}}) = \sum_{\boldsymbol{p}\in\mathcal{V}} ||C_{\mathcal{S}}(\boldsymbol{p}) - C_{\mathcal{I}}(\boldsymbol{p})||_2 \tag{6}$$

where $C_{\mathcal{S}}$ is the synthesized image, $C_{\mathcal{I}}$ is the input RGB image, and $\boldsymbol{p} \in \mathcal{V}$ denote all visible pixel positions in $C_{\mathcal{S}}$. Similarly, E_{lan} is defined as

$$E_{lan}(\boldsymbol{\mathcal{P}}) = \sum_{\boldsymbol{f}_i\in\mathcal{F}} ||\boldsymbol{f}_i - \boldsymbol{K}\boldsymbol{D}(\boldsymbol{R}(\boldsymbol{v}_j) + \boldsymbol{T})||_2^2 \tag{7}$$

Each landmark feature $\boldsymbol{f}_i$ is detected by the landmark detector as described in Section 3.2 and associated with a unique vertex $v_j = \mathcal{M}_{geo}(\boldsymbol{\alpha}) \in \mathbb{R}^3$ of our face model. We include two priors terms to account for the deformation of the faces and fisheye distortion given by

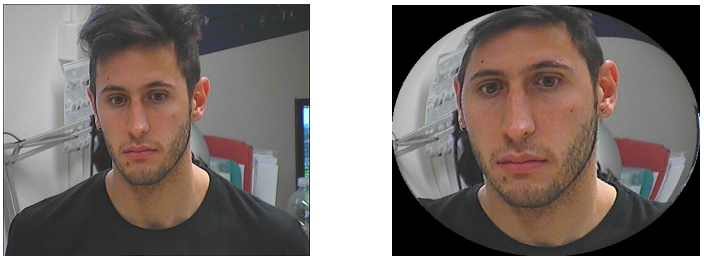
$$\begin{aligned} E_{reg}(\boldsymbol{\mathcal{P}}) = &\sum_{i=1}^{80} \left[\left(\frac{\boldsymbol{\alpha}_i}{\sigma_{shape,i}} \right)^2 + \left(\frac{\boldsymbol{\beta}_i}{\sigma_{tex,i}} \right)^2 \right] \\ &+ \sum_{j=1}^{4} ||\boldsymbol{D}_j - \boldsymbol{D}_{ini,j}||^2 \end{aligned} \tag{8}$$

where $\boldsymbol{D}_{ini}$ is the initial distortion parameters.

We use Ceres Solver [1] to solve this unconstrained nonlinear optimization problem. Since reconstruction from a single RGB image is an extremely ill-posed problem, we propose optimization in five steps. This helps to avoid local minima in the highly-complex energy landscape. Our five steps of optimizations are:

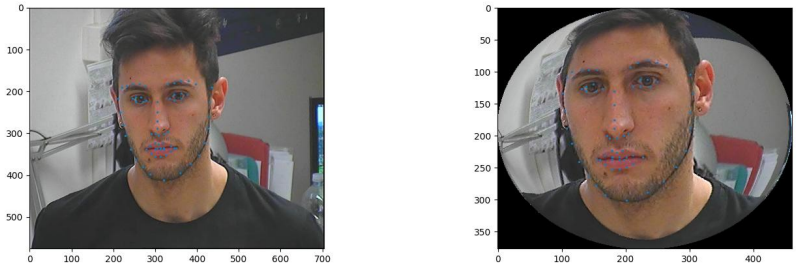
1. We start by minimizing $E_{lan}(\boldsymbol{\mathcal{P}})$ and optimize for the rigid transformation parameters, the camera parameters, and the distortion parameters, keeping other parameters constant.

2. In the second step, we randomly sample 10K vertices from the model and minimize $E_{col}(\boldsymbol{\mathcal{P}})$ and $E_{reg}(\boldsymbol{\mathcal{P}})$.



(a) Image before applying fisheye distortion. (b) Image after applying fisheye distortion.

Figure 1. Example of a fisheye distortion.



(a) Landmarks before applying fisheye distortion. (b) Landmarks after applying fisheye distortion.

Figure 2. Example of landmarks detection.

Here, we optimize for the first 20 shape and texture coefficients along with all the rendering parameters.

3. We randomly sample 30K vertices and optimize for additional 30 shape and texture coefficients along with other parameters by minimizing $E_{col}(\boldsymbol{\mathcal{P}})$ and $E_{reg}(\boldsymbol{\mathcal{P}})$.

4. In the fourth step of optimization, we use all the vertices of the model and optimize for additional 30 shape and texture coefficients along with other parameters by minimizing $E_{col}(\boldsymbol{\mathcal{P}})$ and $E_{reg}(\boldsymbol{\mathcal{P}})$.

5. In the last step, we just optimize for the shape and texture coefficients, keeping all the other parameters constant. This helps in refining the facial details.

Due to the memory constraint of our PC, we only optimize for the first 80 shape and texture coefficients.

4. Results and Analysis

To evaluate our reconstruction pipeline, we take 4 subjects from the MICC Florence face dataset [3]. Each subject is associated with a ground truth scan in neutral expression and video sequences. We extract a specific frame where the face is zoomed. Applying fisheye distortion to a zoomed image enhances the distortion and makes the face highly distorted. As our evaluation metric, we use MeshLab[1] *Distance From Reference Mesh* function to compare the reconstructed mesh and the ground truth mesh. Figure 3 shows the qualitative comparison between the reconstructed meshes from both undistorted and distorted images and the ground truth mesh for 4 subjects. We can see that the middle segment of the face has relatively low error compared to the side segments. The reason for this might be the clear visibility of just the frontal part of the face in the images. Although reconstructed meshes from distorted images are still affected by distortion, the difference between the mean distance from both reconstructed meshes to the ground truth mesh is marginal as shown in Table 1. In some cases, the mean distance of the reconstructed mesh from the distorted image is lower than from the undistorted image. To perform reconstruction from the undistorted image, we run all five steps of optimization for 100 iterations. Therefore, for some meshes, facial details might not have been recovered. Specifically, for subject 41, we observed that the variation between the reconstructed mesh from the distorted image and the ground truth mesh is high. We assume that the coefficients need to be altered relatively more to represent the woman's face. Therefore, as the mean face of the model represents a man's face, adding distortion and face prior constraints at the same led to the optimization stuck in the local minima.

We also investigated the effect of adding distortion prior to the optimization. Figure 4 shows the comparison between ground truth mesh with reconstructed mesh without distortion coefficients/distortion prior and with distortion

[1] https://www.meshlab.net

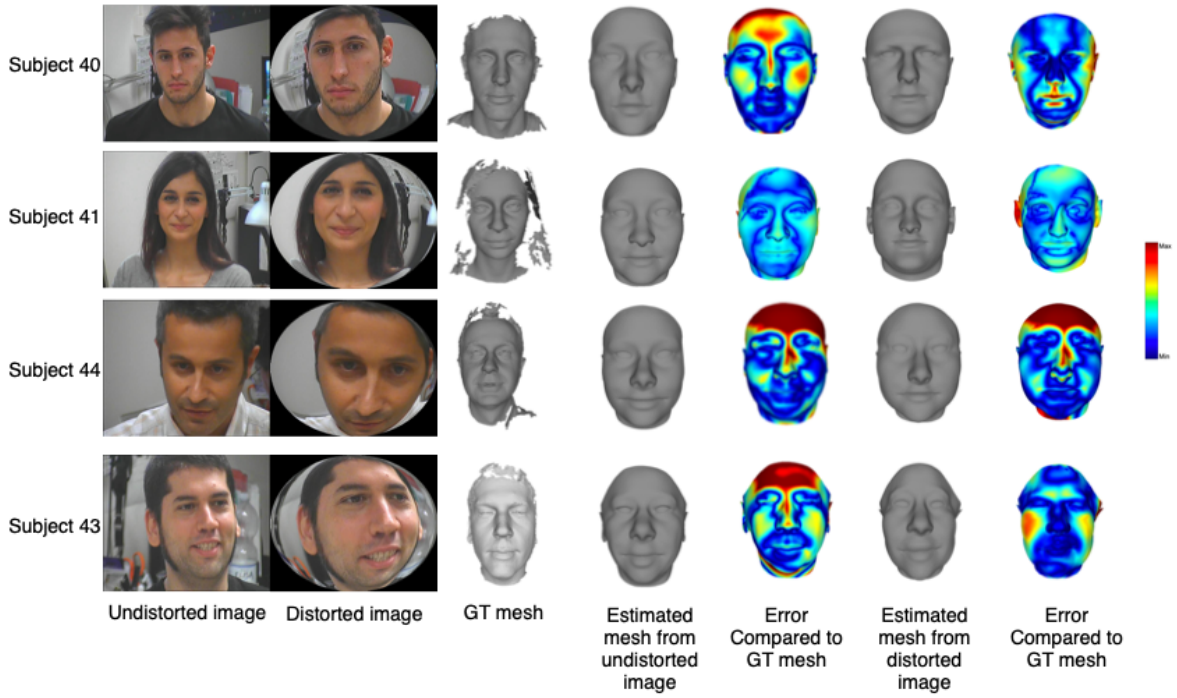


Figure 3. Figure comparing estimated mesh from the undistorted and distorted image of the same subject with ground truth mesh. The error shows the distance between ground truth mesh and estimated mesh where blue is minimum and red is maximum. Error is plotted with respect to the mean distance given in Table 1.

Subject	Mean distance from GT mesh of estimated meshes	
	Undistorted image	Distorted image
40	3.15	3.30
41	10.56	9.30
44	4.49	4.36
43	4.47	5.80

Table 1. Mean Distance from GT mesh of estimated meshes using the undistorted and distorted image of the same subject.

prior in the optimization. We can see that adding distortion prior stops the reconstructed mesh from getting oval like in the distorted image. However, without the distortion prior, the model can't model the distortion correctly. In addition, without the distortion coefficients in the optimization, the model doesn't deviate too much from the prior mean face.

5. Conclusion

We showed that adding distortion prior is crucial while reconstructing from a single distorted image. Moreover, our results verify that reconstruction from fisheye images is an extremely ill-posed that can be further investigated in the future. Future work can be done to reconstruct the face from multi-view fisheye images to model the distortion accurately. Since our landmark detector is not ideal for fisheye images, we use the landmark cost function to initialize the parameters in the first step of optimization. To add landmark cost to other steps of the optimization, we need to accurately detect landmarks on fisheye images. Therefore, fine-tuning existing landmark detectors on fisheye images can also be investigated in the future.

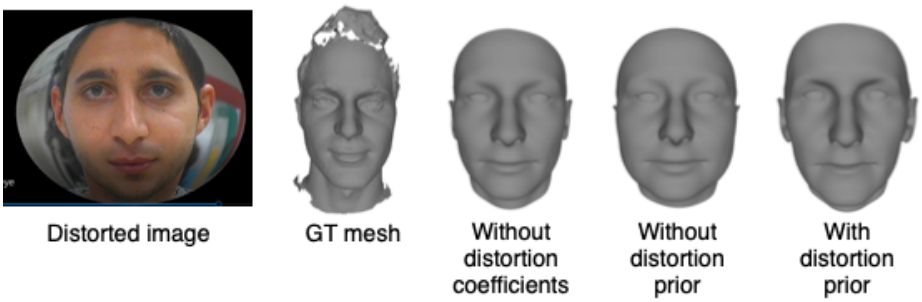


Figure 4. Figure comparing ground truth mesh with reconstructed mesh without distortion coefficients/distortion prior and with distortion prior in the optimization.

References

[1] Sameer Agarwal, Keir Mierle, and The Ceres Solver Team. Ceres Solver, 3 2022. 2

[2] Oswald Aldrian and William A. P. Smith. Inverse rendering in suv space with a linear texture model. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 822–829, 2011. 1

[3] Andrew D. Bagdanov, Alberto Del Bimbo, and Iacopo Masi. The florence 2d/3d hybrid face dataset. In *Proceedings of the 2011 Joint ACM Workshop on Human Gesture and Behavior Understanding*, J-HGBU '11, page 79–80, New York, NY, USA, 2011. Association for Computing Machinery. 1, 2, 3

[4] Anil Bas, W. Smith, Timo Bolkart, and Stefanie Wuhrer. Fitting a 3d morphable model to edges: A comparison between hard and soft correspondences. In *ACCV Workshops*, 2016. 1

[5] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3d faces. In *SIGGRAPH '99*, 1999. 1

[6] Volker Blanz and Thomas Vetter. Face recognition based on fitting a 3d morphable model. *IEEE Trans. Pattern Anal. Mach. Intell.*, 25:1063–1074, 2003. 1

[7] G. Bradski. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*, 2000. 2

[8] Adrian Bulat and Georgios Tzimiropoulos. How far are we from solving the 2d & 3d face alignment problem? (and a dataset of 230,000 3d facial landmarks). In *International Conference on Computer Vision*, 2017. 2

[9] Chen Cao, Yanlin Weng, Stephen Lin, and Kun Zhou. 3d shape regression for real-time facial animation. *ACM Transactions on Graphics (TOG)*, 32:1 – 10, 2013. 1

[10] Cunjian Chen. PyTorch Face Landmark: A fast and accurate facial landmark detector, 2021. Open-source software available https://github.com/cunjian/pytorch_face_landmark. 2

[11] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW)*, pages 285–295, 2019. 1

[12] Pengfei Dou, S. Shah, and I. Kakadiaris. End-to-end 3d face reconstruction with deep neural networks. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1503–1512, 2017. 1

[13] Ohad Fried, Eli Shechtman, Dan B. Goldman, and Adam Finkelstein. Perspective-aware manipulation of portrait photos. *ACM Transactions on Graphics (TOG)*, 35:1 – 10, 2016. 1

[14] Jianglin Fu, Saeed Ranjbar Alvar, Ivan V. Bajić, and Rodney G. Vaughan. Fddb-360: Face detection in 360-degree fisheye images. *2019 IEEE Conference on Multimedia Information Processing and Retrieval (MIPR)*, pages 15–19, 2019. 1, 2

[15] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schoenborn, and Thomas Vetter. Morphable face models - an open framework. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 75–82, 2018. 1, 2

[16] Vidit Jain and Erik G. Learned-Miller. Fddb: A benchmark for face detection in unconstrained settings. 2010. 1

[17] Vahid Kazemi and Josephine Sullivan. One millisecond face alignment with an ensemble of regression trees. In *2014 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1867–1874, 2014. 2

[18] Elad Richardson, Matan Sela, Roy Or-El, and Ron Kimmel. Learning detailed face reconstruction from a single image. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5553–5562, 2017. 1

[19] Andreas Schneider, Bernhard Egger, and Thomas Vetter. A parametric freckle model for faces. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 431–435, 2018. 1

[20] Justus Thies, Michael Zollhöfer, Marc Stamminger, Christian Theobalt, and Matthias Nießner. Face2face: Real-time face capture and reenactment of rgb videos. 2020. 1

[21] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model, 2018. 1

[22] Yu Yanga, Xiao-Jun Wu, and Josef Kittler. Landmark weighting for 3dmm shape fitting, 2018. 1