Single Image 3D Reconstruction of Human Faces

Muhammad Ahmed Riaz
mriaz@ucsd.edu

Abstract

We propose an algorithm to convert a single 2D image of a human face into a full 3D model. In general the Single View Reconstruction problem is under-constrained. However a human face cannot take any arbitrary shape, this provides additional constraints on the 3D shape. Faces have many common features which can be used to make the problem trackable. Our algorithm uses a parametric face model to fit an approximate 3D shape to the image. This coarse shape estimate helps in estimating the scene lighting. Spherical harmonic coefficients are used to compactly represent this lighting. 3D Shape obtained from parametric face model lacks accurate features and high quality details present on the face. We apply shape from shading constraints on the image to get accurate shape with high level of detail. An important component of our system is cast shadow estimation. Cast shadows can be a source of error in depth reconstruction, if not accounted for. We predict the shadow intensity in different parts of the image and compensate for it during shape estimation. This enables our algorithm to handle a wide variety of incident lightings. Key to the success of our method are accurate shading models which we employ, handling effects like diffuse albedos, specular highlights and cast shadows. We provide both quantitative and qualitative results on images from face datasets and on those downloaded directly from the internet. Our system is completely automatic with no user input required, which makes it a scalable solution.

1. Introduction

3D reconstruction of a face is not only an interesting problem in itself, but also many other computer vision applications can make use of this information to make their job easier. Applications of 3D reconstruction can be broadly categorized into two categories; image synthesis and analysis. With the widespread use of cameras and images in the world of internet, content aware image editing is an important field. Users want to interact with their pictures rather than treat them as static entities. If we can extract 3D information from an image, the user can be given options like view point changing and image-based relighting [23]. This can be used to insert someone into a virtual environment. This can also provide a physically plausible image editing system, e.g., to apply virtual makeup on face [17, 14]. Aforementioned applications fall in the category of image synthesis. 3D information can also help with better image analysis. State of the art face recognition algorithms of today construct 3D face models to improve their accuracy. Such reconstruction has applications in security and surveillance as well because it provides better scene understanding. Even though 3D information can be captured directly using a 3D sensor, conventional cameras remain to be the most widely used image capture devices. Over a billion pictures are uploaded to the internet every day. If we want to tap into this vast data reserve, single view reconstruction has to play an important role.

Figure 1: 2D image and its automatically generated 3D Reconstruction

Single View Reconstruction is inherently an ill-posed problem. There are infinitely many possible shapes which can generate the exact same image. During the image capture process there is a loss of dimensionality which makes shape recovery ambiguous. In the literature, a number of cues have been used to resolve this ambiguity. Some of them include Shape from X [25, 19] , vanishing

point based techniques [15] and symmetry cues. In our algorithm we exploit the fact that human faces lie in a low dimensional subspace. The variation between faces of different people is always constrained. We use prior cues about human faces and constrain our solution to lie close to this subspace. This makes the problem tractable. There has been extensive work in the field of single view reconstruction of faces. But most of the work assumes that the face follows a lambertian shading model and there are no cast shadows on the face. These assumptions make the problem easier but the solution obtained under these assumptions is less practical.

Our proposed algorithm takes a single face image as input and computes its 3D shape in the form of a depth map. Specular highlights from the input are removed using color-space rotation as a pre-processing step leaving behind the lambertian component only. We then estimate a coarse face shape by fitting a bilinear 3D morphable model into the image which accounts for both the identity and expression variation in the face. This model also provides an albedo estimate of the face. The initial estimates of shape and albedo are collectively called reference model. We use this reference model to estimate lighting seen in the image in the form of spherical harmonics. Because spherical harmonics can compactly express a linear combination of single light sources, we show that the recovered coefficients are accurate even though the depth and albedo used in their calculation lack details.

Using the estimates of lighting and shape, cast shadows are computed and added to albedo to account for their darkening effect. This makes sure darker regions due to shadowing does not effect the rest of the algorithm. The face shape is improved upon using shape from shading constraints along with estimated shadows, lighting and reference model albedo. A wedge like regularization term ensures that large abrupt changes in depth are penalized heavily while allowing for small high-frequency variations in shape at the same time. The following are our key contributions:

- Dealing with specularity in captured image
- Handling of cast shadows on face, making the algorithm robust
- Use of a piecewise wedge regularizer to preserve high frequency details of shape

2. Related Work

The existing methods of Single View Reconstruction for faces broadly fall into two categories. Those which use a parametric model to explain the observed data. And others imposing physically based constraints like shape from X. Luckily these both complement each other quite well. Which is why more recently people have used a combination of the two techniques to obtain more accurate results.

The use of parametric models for face reconstruction became popular after it was shown to give promising results by Blanz and Vetter [4]. Followup work was done to recover 3D shapes using only a small number of landmark points and statistical modeling in [3]. It sparked interest of a lot of people in this field. FaceWarehouse[6] uses a bilinear model instead which can capture effects of both identity and expression of a person. It is the most widely used face model today. The key idea behind these techniques is to look at a large number of example face geometries and find an orthogonal shape basis. These basis can describe majority of the data in a small number of shapes. For reconstruction, a linear combinations of these shape basis can be used to synthesize a human face and fit it back into the given image. The advantage of using this scheme is that the number of parameters to estimate is fairly small. But the problem with these methods is that they can not express all variations of faces accurately.

The other class of methods, often called shape from X, compute the depth separately for every pixel location.These models make use of physically based models that underline image formation. They estimate the geometry which would produce the given image, so that it agrees with the defined physical princinples. However often it is not possible to infer depth just using image values because these system are highly under-constrained. That is why often these methods take Lambertian reflectance assumption and other similar simplifying assumptions to make the problem tractable. These methods include [25] which gives a survey of other SfS methods, and Johnson et al.[11]. More recent work in this field includes Barron and Malik[1], Han et al.[9].

3. Overview

Given an image $I(x, y)$, 3D reconstruction requires us to estimate the depth $z(x, y)$ at each pixel $\mathbf{p}(x, y)$. Integrability constraints along with Shape from Shading can be used to estimate surface normals $\vec{\mathbf{n}}(x, y)$ as shown by Frankot and Chellapa [8]. These normals can be integrated to get the final depth. We however compute depths directly by expressing surface normals in terms of $z(x, y)$ and applying SfS. More precisely we use the following parametrization

$$\vec{n}(x, y) = \frac{(p, q, -1)^\top}{\sqrt{p^2 + q^2 + 1}}, \tag{1}$$

where p and q are the discrete forward difference derivatives of depth.

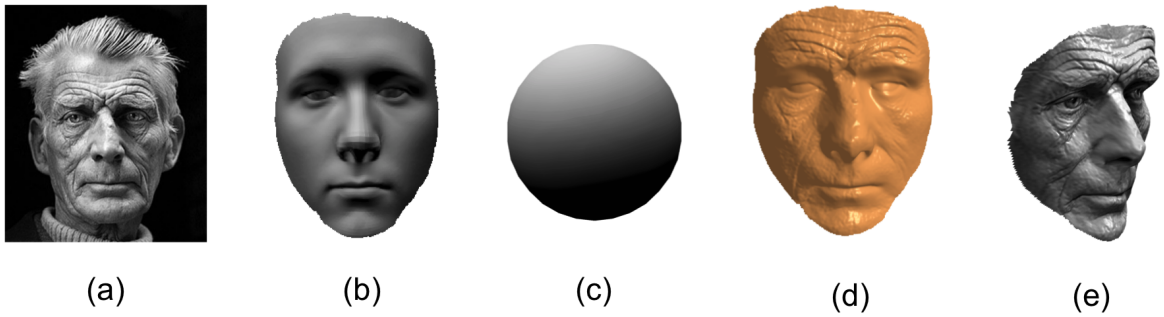


Figure 2: Main stages of the pipeline. (a) is the input image, (b) is the fitted reference model, (c) shows a sphere illuminated under estimate lighting, (d) estimated shape, and (e) shows the texture mapped depth from a novel view-point.

As a preprocessing step we remove specularities from the input, getting an intermediate image that closely follows Lambertian assumption. This is important because Shape from Shading (SfS) constraints for Lambertian surfaces have been studied far more extensively and are simple to apply.

We start the reconstruction by computing a rough estimate of face shape, fitting a parametric model into the image [4, 6]. This estimated shape is used as reference model in rest of the algorithm. The reference model provides us with additional constraints to make the problem trackable and also acts as a regularizer. The regularizer becomes important because real world images contain noise. Without the use of such a reference, noisy pixels introduce large errors in final reconstructed depth. The reference shape makes sure our final output lies close to this initial estimate.

In order to compute the face depths using SfS, we need to estimate the environment lighting. For surfaces that exhibit close to Lambertian behavior, it is sufficient to compute low frequency components of lighting only, using a small number of parameters to express the complete environment map. It has been shown that by using just 9 coefficients of spherical harmonic lighting, average error is under 3% [2, 20]. We use a lighting model employing 13 parameters to specify the lighting (section 4.3) which account for quadratic dependence on surface normals. We use the low frequency estimated normals from reference shape to compute these parameters.

Cast shadows prove to be problematic for SfS constraints because they assume direct lighting. The darkening of image observed due to cast shadows introduces errors in final depth. Given the reference face shape and estimated lighting, we estimate cast shadows before depth refinement. To be precise, for every pixel $\vec{\mathbf{p}}(x, y)$, we compute a darkening factor which informs the algorithm how much darkening is expected due to cast shadows. These are accounted for in rest of the pipeline steps.

Now that we know about the lighting as well as an approximate face shape, we apply SfS constraints on the given image to find out detailed face geometry. For this purpose we set up the a non-linear least square optimization problem with the following cost function:

$$C(z) = \sum_{(x,y)\in I} (I(x,y) - I_{rendered}(\vec{n}(z)))^2 + f(z - z_{ref})^2, \tag{2}$$

where the first term is the data term, quantifying how closely our estimated depth matches input image. Second term is the regularization, which is expressed as some function $f(\cdot)$ of estimated depth z and reference depth z_{ref}. We will discuss this cost function in detail in section 4.4, along with regularization function used.

4. Algorithm Pipeline

We here discuss in detail, each module of our system pipeline.

4.1. Pre-processing

Most subsequent modules of our system assume that our surface (face) loosely follows labertian shading model. Lambertian shading means that apparent brightness of a surface does not depend on the viewing direction. Only contributing factor is the angle between light source and surface normal. All matte surfaces follow this model, whereas surfaces exhibiting specular highlights cannot be explained by this model. Faces mostly behave in a lambertian manner but have small intensity of specular highlights in some regions. In order to identify these

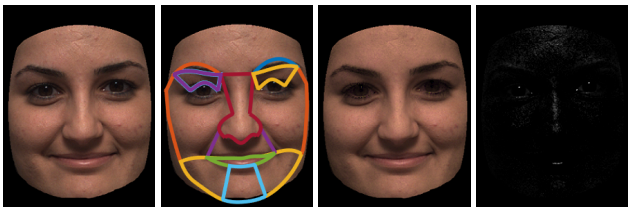


Figure 3: Specularity removal as a preprocessing step. Left to right: Input image, Regions of face, Diffuse component, Specular component

regions and remove these highlights, we use the algorithm described in Li et al.[16].

We run a face landmark detector on input image and divide it into 10 basic regions as shown in 3, second from left image. For each of these regions, we compute the mean color of pixels which are not either too bright or too dark. Our hope is that these pixels do not suffer from effects like shadowing and specularities. This average color is assumed to be the approximate albedo of that face region. These pixels are projected into RGB space and a plane is fitted into them. Each face region is described by a different RGB plane. The common axis of all these planes is found by minimizing the following cost function:

$$C_{light}(\vec{\mathbf{n}}) = \sum_{i=1}^{10} \vec{\mathbf{n}} \times \vec{\mathbf{n_i}} + ||\vec{\mathbf{n}}|| \tag{3}$$

where $\vec{\mathbf{n}}$ is the vector being optimized for and $\vec{\mathbf{n_i}}$ are the normals to planes fitted in each face region. If there is a difference in albdeo of different face regions, the color vector found as a result of the optimization would be pointing in light source color. This follows from the fact that when a surface of albedo color $\vec{\mathbf{c_1}}$ is lit by a light source of color $\vec{\mathbf{c_2}}$, the surface appears to be of a color, that is a linear combinations of the two color vectors [13]. So if each face regions has a different albedo, the only common axis of all those planes should be the light source color vector.

4.2. Reference Model

The image formation equation that gives us a relationship between image intensity and surface normals has a lot more unknowns than the number of constraints provided by a single image. Other than plane normals of surface depth, surface albedo (texture) and lighting are also unknown. The problem is ill-posed if we do not apply any additional constraints. In order to provide those constraints, a reference model is fitted into the image. This model provides a low resolution depth depth map of the face. This captures the overall shape and structure of face and provides the rest of the algorithm with an estimate of the

(a) Image with contours (b) Reference shape

Figure 4: Reference model fitting

final depth to be recovered. This reference model can be used to apply additional constraints on the solution and to make the problem trackable. For this purpose, use of a parametric model to estimate the depth is the natural choice.

Fitting a parametric face model into a 2D image to estimate the face shape was shown to work successfully in the seminal work of Blanz and Vetter[4]. The most recent and prominent work following the same train of logic [6] uses face scans of 150 subjects with various expressions to come up with a bilinear face basis. These basis capture both the identity and facial expression of a face. We make use of this face model to get the reference.

We start by finding landmark points on the input face image using Active Shape Models with Stasm [18]. This gives us landmark points both inside the face (such as eyes, nose, lips etc) and at face contours. Corresponding landmarks points on the 3D face model are also detected and the projection error of these model points is minimized using the following error function:

$$E_k = \frac{1}{2}||s\mathbf{R}(C_\mathbf{r} \times_2 \mathbf{w}_\mathbf{id}^\top \times_\mathbf{3} \mathbf{w}_\mathbf{exp}^\top)||^2 \tag{4}$$

where $s, \mathbf{R}$ are the pose parameters (scale and Rotation) and $\mathbf{w}_\mathbf{id}^\top, \mathbf{w}_\mathbf{exp}^\top$ are internal parameters of the model. $C_\mathbf{r}$ are the internal vertices of model bilinear basis. The optimization is performed using coordinate descent method as described in [24]. To make sure we do not inherit any issues from the bilinear model itself, for some experiments we use Basel face model[10] in place of the Facewarehouse. Once a parametric face model has been fit into the image, the ambiguity in pose and size of face is resolved. Also some aspects of the shape of the face are captured at this stage. The model not only gives rough information about the shape of face but also its albedo. We will use this albedo estimate for SfS in later stage of the pipeline. An example model fit into an image can be seen in fig 4.

Use of a parametric model to estimate face shape is a tangential method to Shape from Shading. In some respect it is also complimentary to the SfS technique. Parametric model does a good job at capturing overall structure and its solution is always plausible face shape. However, it fails to capture the unique details which may be specific to a particular person. Because it uses a linear combination of some basis shapes, it can capture only limited amount of variations in shape. This is a general limitation of all parametric models.

4.3. Lighting Estimation

Accurate estimation of lighting is essential for depth estimation using SfS. Any errors incurred at this stage of the algorithm would directly effect the final recovered depth. We use spherical harmonics (SH) for representation of environment lighting. The SH coefficients are computed such that when reference model is rendered using our lighting model, it matches the input image. We infer the lighting based on the shading seen in input image. But as a human face mostly follows Lambertian assumptions, it is hard to accurately predict higher order lighting effects. If has been shown by [2] that if we use spherical harmonic basis to represent illumination on Lambertian surfaces, using just 4 coefficients captures 70% of the pixel values correctly. Likewise, using 9 coefficients captures more than 97% of the pixel values accurately. We test our system under two different lighting models, a first order and a modified second order spherical harmonic system. The following subsections contain more detail about these models.

4.3.1 1st order SH lighting

Use of spherical harmonic lighting basis and coefficients is a compact way of representing environment lighting. Under the Lambertian assumption, shading equation for SH model can be written as:

$$I(x,y) = \rho(x,y)(\vec{l} \cdot \vec{\mathbf{Y}}(\vec{n})) \tag{5}$$

where $\vec{l}$ are the lighting coefficients and $\vec{\mathbf{Y}}$ are the SH basis functions. We find the values of these SH coefficients by minimizing the difference between the two sides of the equation. This can be solved linearly as matrix multiplication as:

$$\vec{l} = I \cdot (\rho_{ref} \mathbf{Y}(\vec{n}_{ref}))^{-1} \tag{6}$$

For 1st order approximation of the above system,

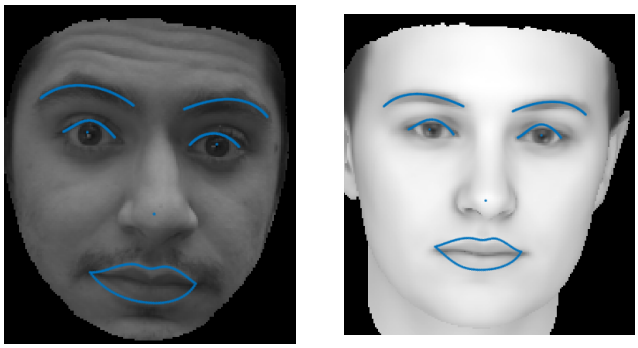
$$\mathbf{Y}(\vec{n}_{ref})) = [1, n_x, n_y, n_z] = [1, \frac{p, q, -1}{\sqrt{p^2+q^2+1}}] \tag{7}$$

4.3.2 Quadratic SH lighting

It has been shown in [20, 11] that for accommodating more complex lighting scenarios than a linear combination of directional light sources, we can use the following quadratic parametrization for SH lighting:

$$I = \rho(x,y)(\vec{\mathbf{n}}^\top A \vec{\mathbf{n}} + \vec{\mathbf{n}}^\top b + c) \tag{8}$$

where A, b and c are model parameters. We have tested both of the above lighting systems and the difference between the results of the two are minimal. This is due to the fact that Lambertian surfaces act as low pass filters on light falling on them and they filter out the higher frequency component of light. More details on how to solve for this optimization can be found in [7].



(a) Input image (b) Reference model

Figure 5: Landmarks for Moving Least Squares Morphing. As it can be seen from the figures, we only retain landmark points which are inside the face and discard contour points. Also after the landmarks are obtained, we fit splines into them to get smooth and even morphing, independent of the location of individual landmarks detected.

4.4. Shape From Shading

4.4.1 Reference Alignment

The parametric model fitting provided a coarse alignment between the image and reference model. This level of alignment was acceptable for computing lighting coefficients because the system was highly over-constrained. Only 4 to 13 unknowns (lighting coefficients) were to be computed using thousands of constraints (one for every pixel). Any errors in alignment were averaged out and their effect was cancelled out in the complete optimization. But in the application of SfS, we have as many number of unknown as many pixels. For every single face pixel, we need to compute a depth value. Any misalignment between the reference and input image could introduce serious errors in the final depth, specially in areas of high variations (e.g. around the nose). To resolve this problem we add a dedicated alignment stage in the pipeline. This alignment is achieved by a two step process.

(a) Input image (b) Predicted shadows

Figure 6: Fractional cast shadow prediction for an input 2D image.

First we perform morphing between landmark points of reference model and input image. Ideally these landmarks should already be lined up after the reference fitting stage. But it is possible certain configurations were highly unlikely and the morphable model was not able to align all landmark points between the two sets of images. In this case corresponding sets of landmark points (identified using Stasm tracker) are aligned. In doing so, we also move the pixel near those landmark points to morph the image realistically and smoothly. We use Moving Least Squares[21] for performing this operation. Landmarks detected on image and reference for morphing can be seen in fig 5.

We then compute a dense optical flow between image and reference. This flow is applied on reference model to establish tight pixel to pixel correspondence with the input image. We perform multi-scale optical flow with multiple iterations at each level of the pyramid. For the results shown in this paper, we used 3 levels deep pyramids with 2 iterations of flow at each level. We are able to recover large variations in shape using this technique. The computed flow is applied on both the reference shape and albedo. For flow computation we use Large Displacement Optical Flow[5].

4.4.2 Cast Shadow Estimation

In the real world pictures, it often happens that some parts of the face cast shadows on other areas. The shadow of nose can often be seen in portraits. This self shadowing changes the intensity of certain pixels and they no longer follow the SfS constraints. If we do not account for this darkening, it can introduce errors in recovered depth.

Cast shadows in the domain of 3D Face Reconstruction have been overlooked. Almost all the existing literature ignores cast shadows and directly applies SfS to compute depths. One reason for this simplifying assumption is that cast shadow prediction is an expensive task. So to keep the execution time short, often they are ignored. We introduce an efficient algorithm to compute these shadows using PRT (Precomputed Radiance Transfer) framework [22]. PRT allows you to do pre-computations, promising faster rendering speeds at runtime. It is a standard technique used by high-end games in the industry. We make use of the pre-computation trick to quickly estimate fractional shadowing even when lighting is not known before-hand.

In order to compute the shadowing, we do pre-computations for rendering the face reference under arbitrary lighting with and without shadowing. At run-time, we render the face under both conditions (with and without cast shadows) and compute a ratio of the two to find fraction of light blocked due to shadowing. We compensate for this expected darkening by multiplying this darkening factor with albedo and merging them together. The rest of the algorithm runs as usual and is not effected by shadowing. If the darkening factor is S, it can be computed as:

$$S = \frac{\vec{l} \cdot M_s}{\vec{l} \cdot M_{ns}} \tag{9}$$

$$M_s = \int_\Omega V_p(s) cos(N_p, s) ds \tag{10}$$

$$M_{ns} = \int_\Omega cos(N_p, s) ds \tag{11}$$

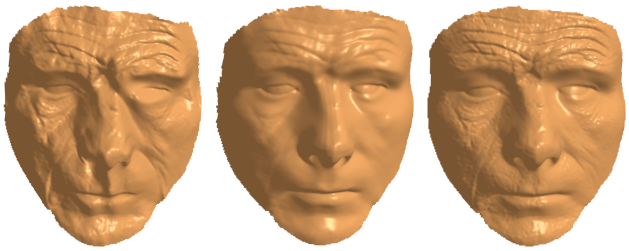
where M_s is the precomputed coefficients for shadowed rendering and M_{ns} are precomputed coefficients for non-shadowed rendering. Refer to [22] for more details. The ratio of the two, S, provides with the relative shadowing. The factor is incorporated into albedo ρ' to get the updated albedo ρ as:

$$\rho(x, y) = S(x, y)\rho'(x, y) \tag{12}$$

Relative shadowing for a specific example can be seen in fig 6. It is worth noticing here that we perform the shadow estimation based on reference shape, which is the initial estimate of the face shape. We assume that during the optimization, the changes in depth do not change the shadowing pattern significantly. This assumption helps us do the pre-computation once and use the same results at each iteration of optimization.

4.4.3 Depth Estimation

SfS constraints along with reference model and estimated lighting are used to estimate the refined depth of the face at each pixel. We use the depth-map representation of face shape in our work. This representation makes it easier to form correspondence between given image and estimated depth. This means there is a $1 - 1$ correspondence between



(a) High reg (b) Low reg (c) Piecewise Wedge

Figure 7: Affects of different regularization terms on depth.

pixel values and depths.
To compute the shape, we set up the following optimization:

$$\min_{z} \sum_{(x,y)\in I} (I - \rho_{ref}(\vec{n}^\top A\vec{n} + \vec{n}^\top b + c))^2 + \lambda_1(\triangle G * d_z)^2 + \lambda_2(d_z)^2,$$

where d_z is the difference between estimated and reference depths. First term is the data term which enforces SfS constraints. Second and third terms are for regularization. Second term makes sure that if z goes away from z_{ref}, it does so in a smooth manner. The last term makes sure our final estimate is not very far away from initial estimate. We solve this optimization in terms of z directly. That ensures we do not have to enforce integrability constraints separately. They get baked into the optimization inherently.

As our data term involves surface normals, these are computed using forward differences. Hence each data term depends on more than one depth variables. We are not able to compute the data term at the face boundaries. At these boundary pixels we apply boundary constraints. The constraint we choose for these pixels is the following:

$$\nabla(\nabla z \cdot \vec{n}) \cdot \vec{n} = 0$$

This constraint means that at boundary pixels, if you travel towards the center of the face, the slope of depth in that direction should no change. This ensures that surface remains smooth at boundary. We take the motivation of this boundary condition from Kemelmacher 2011[12].

4.4.4 Piecewise Wedge Regularizer

The optimization of depth involves a regularization weight. Setting this weight to one specific value does not give ideal results. If we set the regularization very large, estimated depth follows the overall shape of the game nicely but most of the details on face are smoothed out and blurry. On the other extreme, if we set to too low, details are preserved nicely but structure of the face break down. There are undesirable artifacts on the recovered depth. A comparison of the two extreme weights can be seen in 7(a), 7(b).
We propose to use a piecewise wedge shaped regularization term instead. The penalty imposed by this term is small bellow a certain threshold δ. In this range, the penalty function is scaled down by a factor α, as seen in figure 8. This allows for small high frequency changes to be permitted, but discourages larger irregularities. The depth result generated using the proposed regularizer is shown in 7(c).

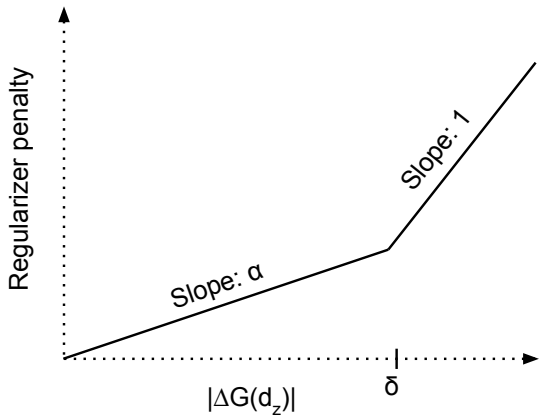


Figure 8: Piecewise Wedge Regularizer

5. Results

Qualitatively evaluating the accuracy proved to be a challenging task. For quantitative analysis, there must be images of faces along with ground truth depth maps of faces available. Collection such data is not straight forward. Not many datasets exist which have both of these quantities available. We have reported our results on USF face dataset. But it should be noted that the dataset itself is not completely accurate. The errors shown in results could arise due to our algorithm as well as the errors in ground-truth itself.

5.1. USF Dataset depths

The dataset contains depth maps and albedo maps of around 80 subjects. The the subjects are in neutral poses which makes the data uninteresting. Also the albedo maps provided are not accurate. Subject to these errors, we generate images using the given depth maps and albedos by simulating a combination of 3 light sources projected from random directions. Each image is scaled to size 480×360. Errors are reported in average error in millimeters per pixel and percentage errors in figure 9. For almost all the images in the dataset, the mean error is less than 5mm per pixel.

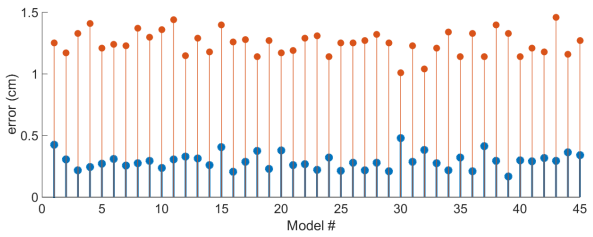


Figure 9: Errors in reconstruction of USF dataset faces, reported in cm. The red lines denote errors in reference faces. Blue lines are the errors after optimization. It can be seen that in every model, the error in depth reduces substantially.

5.2. Internet Images

We downloaded face images directly from the internet, to test our algorithm on uncontrolled data. The system was given no information about the pose or lighting of the image. Fully automated results on some internet images can be seen in figure 10.

5.3. USF Dataset lighting estimation

We generated images using single light source synthetically from USF dataset. From these synthetic images, light source direction is estimated and compared with ground truth direction. For each direction should here, the reported are mean errors computed over whole dataset in degrees. The model of lighting used for this experiment was 1st order SH lighting. Errors can be seen in fig 11. We find that our lighting estimation error is reasonably low. It is always less than 10° mean error.

5.4. Cast Shadow Estimation

In figure 12 it can be seen that without cast shadow estimation, recovered depth is visually incorrect. This problem is resolved when we take into account cast shadows.

6. Applications

We have developed applications over our system to test its reliability. We can successfully perform view-point changing and light source changing. Accurate images can be generated under entirely new lighting environment maps. However one application that we are still looking to make is a web interface that can provide users with the ability that they can upload their 2D image on the server and get back their 3D face model within minutes.

7. Conclusions

We provide with a completely automatic end-to-end face reconstruction system that is robust enough to handle images under varied situations and conditions with complex effects like specular highlights and cast shadows. We provide the novel way of handling cast shadows and combining multiple depth maps together to get more accurate results.

8. Acknowledgments

I want to thank Sony for their funding towards the project. Also special thanks for William Smith for helping be figure out parametric face models and providing me with his code as well.

References

[1] J. T. Barron and J. Malik. Color constancy, intrinsic images, and shape estimation. In *Computer Vision–ECCV 2012*, pages 57–70. Springer, 2012.

[2] R. Basri and D. W. Jacobs. Lambertian reflectance and linear subspaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(2):218–233, 2003.

[3] V. Blanz, A. Mehl, T. Vetter, and H.-P. Seidel. A statistical method for robust 3d surface reconstruction from sparse data. In *3D Data Processing, Visualization and Transmission, 2004. 3DPVT 2004. Proceedings. 2nd International Symposium on*, pages 293–300. IEEE, 2004.

[4] 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*, pages 187–194. ACM Press/Addison-Wesley Publishing Co., 1999.

[5] T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(3):500–513, 2011.

[6] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou. Facewarehouse: A 3d facial expression database for visual computing. *Visualization and Computer Graphics, IEEE Transactions on*, 20(3):413–425, 2014.

[7] M. Chai, L. Luo, K. Sunkavalli, N. Carr, S. Hadap, and K. Zhou. High-quality hair modeling from a single portrait photo. *ACM Transactions on Graphics (TOG)*, 34(6):204, 2015.

[8] R. T. Frankot and R. Chellappa. A method for enforcing integrability in shape from shading algorithms. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 10(4):439–451, 1988.

[9] Y. Han, J.-Y. Lee, and I. Kweon. High quality shape from a single rgb-d image under uncalibrated natural illumination. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1617–1624, 2013.

[10] IEEE. *A 3D Face Model for Pose and Illumination Invariant Face Recognition*, Genova, Italy, 2009.

[11] M. K. Johnson and E. H. Adelson. Shape estimation in natural illumination. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, pages 2553–2560. IEEE, 2011.

[12] I. Kemelmacher-Shlizerman and R. Basri. 3d face reconstruction from a single image using a single reference face shape. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(2):394–405, 2011.

Figure 10: Depth reconstruction on internet images. Column 1,3 show input images and column 2,4 show the corresponding depth maps.

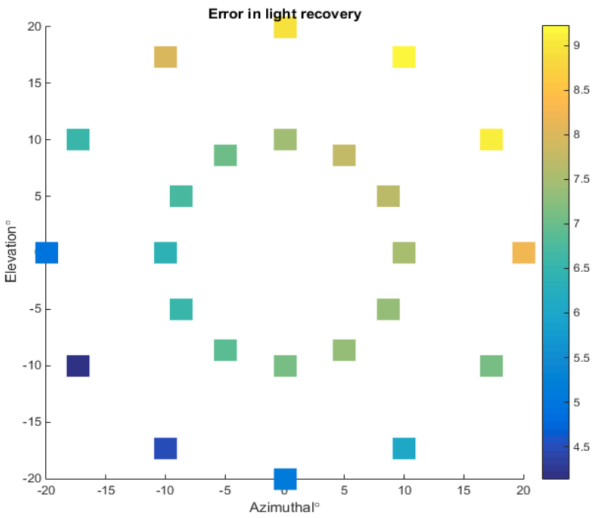


Figure 11: Errors in light source direction estimation averaged over all USF dataset.

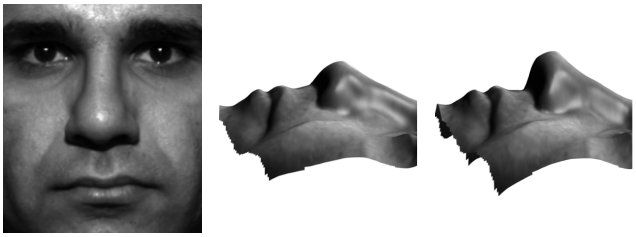


Figure 12: Improvement in depth due to cast shadow estimation.Left to right: Input image, depth w/o shadow estimation, depth w/ shadow estimation. Artifacts are noticeable under the nose in middle image.

[13] G. J. Klinker, S. A. Shafer, and T. Kanade. The measurement of highlights in color images. *International Journal of Computer Vision*, 2(1):7–32, 1988.

[14] D. Kriegman. TAAZ virtual makeover and hairstyles. `http://taaz.com`. Accessed: 07-14-2016.

[15] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2136–2143. IEEE, 2009.

[16] C. Li, K. Zhou, and S. Lin. Intrinsic face image decomposition with human face priors. In *Computer Vision–ECCV 2014*, pages 218–233. Springer, 2014.

[17] S. P. Mallick, T. Zickler, P. N. Belhumeur, and D. J. Kriegman. Specularity removal in images and videos: A pde approach. In *European Conference on Computer Vision*, pages 550–563. Springer, 2006.

[18] S. Milborrow and F. Nicolls. Active Shape Models with SIFT Descriptors and MARS. *VISAPP*, 2014.

[19] S. K. Nayar and Y. Nakagawa. Shape from focus. *IEEE Transactions on Pattern analysis and machine intelligence*, 16(8):824–831, 1994.

[20] R. Ramamoorthi and P. Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the 28th annual conference on Computer graphics and interactive techniques*, pages 497–500. ACM, 2001.

[21] S. Schaefer, T. McPhail, and J. Warren. Image deformation using moving least squares. In *ACM transactions on graphics (TOG)*, volume 25, pages 533–540. ACM, 2006.

[22] P.-P. Sloan, J. Kautz, and J. Snyder. Precomputed radiance transfer for real-time rendering in dynamic, low-frequency lighting environments. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 527–536. ACM, 2002.

[23] B. Tunwattanapong, A. Ghosh, and P. Debevec. Practical image-based relighting and editing with spherical-harmonics and local lights. In *Visual Media Production (CVMP), 2011 Conference for*, pages 138–147. IEEE, 2011.

[24] D. Vlasic, M. Brand, H. Pfister, and J. Popović. Face transfer with multilinear models. In *ACM Transactions on Graphics (TOG)*, volume 24, pages 426–433. ACM, 2005.

[25] R. Zhang, P.-S. Tsai, J. E. Cryer, and M. Shah. Shape-from-shading: a survey. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 21(8):690–706, 1999.