

Using Hybrid Approaches to Solve the Challenges of Shape from Shading

Ryan Murphy and Zoë J. Wood

California Polytechnic State University, Computer Science Department, San Luis Obispo, CA;

ABSTRACT

For over thirty years researchers have been trying to solve the shape from shading problem of determining 3D shape from a single image with a single light source. The basic problem of determining shape from shading is made more difficult due to challenges of light orientation, camera type, ambiguity, multiple materials, and specular highlights. This paper shows how some of these challenges can be overcome through the use of other techniques such as image segmentation and stereopsis. We present a new hybrid method of shape from shading that can be used to autonomously capture 3D information from two 2D images of single objects with multiple peaks and multiple materials with specular components.

Keywords: Shape from shading, stereopsis, image segmentation, surface reconstruction

1. INTRODUCTION

The Shape from Shading (SFS) problem can be described as acquiring 3D information about a surface from the intensity values of a single image.¹ Three dimensional surface acquisition has applications in the computer vision, archaeological, medical, and entertainment fields because it provides a way to study, interact, visualize, and manipulate computational representations of real data. The simple, cheap, fast features of SFS techniques make it very attractive to sectors of industry that cannot afford expensive range scanners.²

In general, SFS techniques work by using intensity values of an image to infer 3D information. This is the inverse of light models, which take 3D information and produce intensity values. For example, the popular Phong Model separates the total intensity visible from an object into an ambient glow, diffuse shading, and specular highlights,³ each produced by a relationship between properties of the incoming light and the 3D scene. In order to simplify the SFS problem, most techniques only address the diffuse component and thus are only applicable to lambertian reflectors.⁴ In brief, SFS solves for depth by using the diffuse intensity value (given by $I = \vec{L} \cdot \vec{N}$,³ where I is the final intensity value, \vec{L} is a 3D vector representing the direction of the incoming light, and \vec{N} is the normal at that location on the surface). Assuming that light direction and at least one normal is known, the value of neighboring normals can be computed using the light direction and corresponding intensity values from an image. Then, using the computed normals for each point, depth can be computed and used to construct a 3D surface. However, SFS techniques that acquire 3D information autonomously from a single 2D image are not without their challenges.

Some of the major challenges that modern SFS techniques face are: light orientation with respect to the camera, perspective versus orthographic cameras, ambiguity (whether a surface region is convex or concave), multiple material surfaces, and specular highlights. The first two challenges have been addressed in previous research⁵⁻⁹ and their general assumptions are acceptable for a number of SFS techniques. This paper focuses on solving the challenges of ambiguity, multiple materials, and specular highlights. The ambiguity problem arises when given a single image with a single light source, SFS cannot determine if a specific section of the surface is concave or convex without additional information or constraints. In fact, Belhumeur, Kriegman, and Yuille⁸ prove that when the lighting direction is unknown and specular highlights are negligible then the same image can be obtained by a continuous family of surfaces. Similarly, most of the SFS research has been done on surfaces with a single material, but little research has been done on surfaces with multiple materials. For example, traditional

Further author information: (Send correspondence to Z.J.W.)

Z.J.W.: E-mail: zwood@calpoly.edu

R.M.: E-mail: drako22@gmail.com

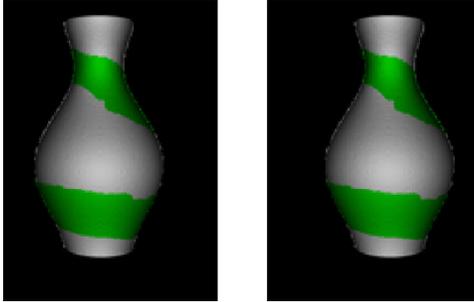


Figure 1.

A stereo pair of images of a vase with multiple materials. This is a sample input to our algorithm. Resulting reconstructions can be seen in Figure 3

SFS approaches would not provide good solutions for images like the one in Figure 1. Likewise, most SFS approaches assume a surface will reflect light consistently as a lambertian reflector and since SFS techniques use the assumption that high intensity regions correspond to flat regions, without knowing how strong the specular component is on a material, the reconstructed surface may suffer from extreme flat spots surrounded by exaggerated slopes. Our algorithm addresses these problems thus broadening the applicability of SFS to a wider range of input images.

We present a new SFS method which addresses the challenges of ambiguity, multiple materials and specularity. Our method:

1. Captures depth information from a single object using a single light source and two views with solving for correspondence for only a handful of points.
2. Allows for surfaces with multiple local minima and local maxima.
3. Allows for surfaces with multiple materials.
4. Captures depth information in real-time.
5. Operates with little to no user input.

This new algorithm allows for the construction of 3D data from a single light source with minimal additional information.

2. PREVIOUS WORK

Berthold Horn discovered that the shape of a surface could be inferred from intensity information found in an image of a surface. Horn originally used a technique that started from one edge of an image and marched along the image constructing a surface based on the observed intensity values at any given point.¹⁰ His approach later evolved to solve for shape by finding the solution of a nonlinear first-order Partial Differential Equation (PDE) called the brightness equation.¹¹ This equation is known as an Eikonal equation.

$$\Delta z(x, y) = \sqrt{\frac{1}{I^2(x, y)} - 1} \quad (1)$$

This equation directly relates the change in depth of a surface to its intensity values, with the depth represented as the magnitude of the gradient of $z(x, y)$ and the intensity values as $I(x, y)$. A brief derivation of this equation as it relates to SFS can be found in the work of Zeng et. al.¹² Since its origins with Horn, the field of SFS has branched into four different directions: minimization, local, linearization, and propagation. Detailed descriptions of these techniques as well as some of the results from them can be found in recent surveys.^{5, 13, 14}

Minimization or optimization approaches work by solving for the depth values of a surface by minimizing an error function. In addition to choosing an error function, a minimization technique has to be chosen.¹³ Although a majority of research in SFS uses minimization techniques, the approaches are still quite limited in the variety of surfaces that a given method can work on¹³ (e.g. surfaces with multiple local minima and maxima). **Local** approaches work by assuming an underlying shape for a region. Generally, this assumption is that the surface is locally spherical.¹⁵ Because of this assumption these solutions perform extremely well on surfaces that are spherical in nature, however, they perform poorly on other surfaces. **Linear** approaches take the non-linear problem introduced by the Eikonal equation and reduce it to a linear equation through the linearization of the reflectance map. A reflectance map is an equation that maps normals on a surface to intensities in an image. Tsai and Shah's work¹⁶ is one of the best examples of linearization.¹

Propagation approaches work by propagating solutions from known points. The first SFS technique that was developed by Horn in 1970 was actually a propagation technique.¹⁰ Propagation techniques start from a single or set of known points, and then propagate outward across the image to form the surface using intensity information from the input image as it marches outward from the given point. Horn's initial solution worked on propagating depth from edges, while Oliensis observed that surface shape can be reconstructed from singular points rather than from edges or known curves.¹⁴ The most well known propagation technique was developed by Kimmel and Sethian.¹⁷ Their method, dubbed Fast Marching SFS (FM-SFS), is able to achieve extremely promising results (refer to Figure 3) with complexity of $O(N \log N)$, where N is the number of pixels in an image.¹⁷ The algorithm works as follows:

1. The algorithm is initialized by adding a single point to a minimum depth sorted priority queue. This initial point is said to be the peak which is the highest point on the surface. The algorithm then loops through the remaining steps until the queue is empty.
2. Remove the head of the queue and compute the depths of its unprocessed neighbors by adding the change in depth of the neighboring pixel to the depth of the head. Where the change in depth is computed by solving Equation 1.
3. The newly computed neighbors are then added to the queue to be processed later.

One disadvantage to this approach is that it works from the assumption that the surface we are trying to compute has only one maximum. For example, an image of a face would result in a surface with a rounded region around the nose, but would not capture the local maxima at the cheeks, lips, etc. Weakness aside this method is popular because it efficiently provides a solution to the Eikonal equation by simply marching around an image. In the most recent survey to date FM-SFS gave an overall performance on both synthetic and real images that proved to be better than all other SFS techniques.¹³

Our algorithm is a propagation method that builds on the work of Zeng, Matsushita, Quan, and Shum,¹² however, their algorithm uses human knowledge to solve for concave versus convex ambiguity. Due to the fact that FM-SFS only works for surfaces with a single maximum (or minimum) point it cannot be used by itself to form an accurate solution to a complex surface with many peaks (or ruts). Due to this limitation Zeng and colleagues' algorithm first locates any local maxima and then computes the relative altitudes between the maxima to then be used in the FM-SFS algorithm. In order to locate the maxima their algorithm relies on user input, where the user provides a number of surface normals and the algorithm traces those normals back to peaks. However, because a user only specifies orientation at points of their choosing, the number of computed peaks is often lower than the actual number of peaks. This weakness is overcome by having the user look at the constructed surface, then iterate specifying more normals to construct a solution that is to their satisfaction. This technique provides aesthetically pleasing results. However, this solution does not lead to an easy way to remove the human factor limiting its application to domains such as computer vision. The method presented in this paper builds on the work of Zeng and colleagues by using a FM-SFS technique that allows for multiple peaks, but without the need for user intervention.

3. ALGORITHM

Our new algorithm for SFS expands upon the work of Zeng and colleagues,¹² by using stereopsis and image segmentation to address some of the open challenges of SFS.

3.1 Solving the Challenge of Ambiguity

The challenge of solving for convex versus concave surface regions can be solved with the help of a user.¹² However, if we want to construct a SFS technique that can be used in fields where user interaction is not feasible (e.g. computer vision), then we must remove the user from the process. In Zeng and colleagues' work, a user specified a series of normals, then those normals were used to determine the location of peaks.¹² They had the user specify normals because it is easier for a human to recognize surface orientation instead of indicating depth. However, it is easier for a machine to indicate depths of certain points using stereopsis rather than specifying a handful of normals.

With this in mind, our new SFS technique works by first capturing two images of an object from slightly different view angles in order to use stereopsis to solve for depth for a small handful of pixels. The algorithm first locates the pixels in one of the images that have greater intensities than all of its neighboring pixels in the same image. These pixels are marked as possible **peaks** and corresponding peaks are identified using epipolar geometry constraints, using a window size of seven by seven pixels. Stereopsis is then used to calculate the depth at each of the marked possible peaks. For more information about stereopsis, see for example the following taxonomy.¹⁸ Finally, FM-SFS is performed from these peaks. The advantage this technique has over the work of Zeng et. al. is that the user is eliminated from the process. To summarize, once the images are captured the algorithm proceeds as follows.

1. Iterate through all intensity values in an image and determine which values are possible **peaks**. Specifically:
 - for each** x, y pixel coordinates in an image
 - do** set neighborhood depth i (values of $i = 3$ most frequently used)
 - do** Loop through all neighboring pixels from $(x-i, y-i)$ to $(x+i, y+i)$
 - if** intensity of all neighboring points increase to center, i.e. $Intensity(x, y)$
 - Mark x, y coordinate as possible peak
2. Use stereopsis to determine the altitudes at each of the possible peaks.
3. Perform FM- SFS using the computed depths as initial values into the depth sorted priority queue.

Because this method uses the FM-SFS algorithm in the same way as,¹² it would yield the same results that are shown in the work of Zeng and colleagues. The only difference between the two techniques is that our technique determines the initial depths of the input points. Our technique could potentially identify more peaks than Zeng et. al.,¹² however, since peaks will already have their depths computed, these values will not have to be recomputed during FM-SF, significantly speeding up this stage of the algorithm. The use of stereopsis to remove the user from the SFS problem allows the technique to be used in a wider number of applications. Some results of this autonomous version of SFS can be seen in Figure 2. The two surfaces generated, both with or without a user's input, are nearly identical demonstrating the usefulness of our algorithm. Note that this solution is also much faster than traditional stereopsis algorithms which needs to compute correspondence for every pixel in an image and thus are much slower. Our algorithm only needs to compute correspondence for a handful of peaks and then uses FM-SFS in order to compute the depths for the rest of the image.

3.2 Solving the Challenge of Multiple Materials

To address the fact that a surface may be composed of multiple materials (i.e. surfaces with starkly contrasting color values like those in Figure 1), our algorithm uses image segmentation to separate a surface into multiple regions. We then modify our SFS algorithm for each region to ensure that the transitions between materials are smooth. The FM-SFS algorithm used by our algorithm relies on an equation that relates intensity values to change in depth (see Equation 1). Intensity values in this equation can be constrained to range from one

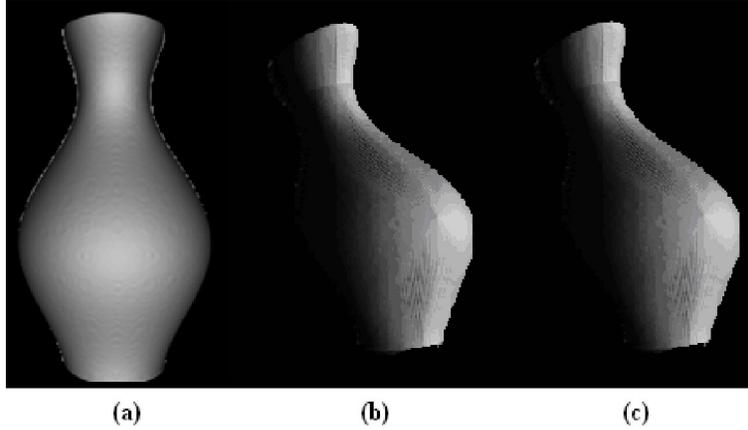


Figure 2.

Using Stereopsis to Solve the Challenge of Ambiguity : (a) Input image of a vase. (b) Surface constructed through the help of user input. (c) Surface constructed autonomously using stereopsis to determine peak depths.

to zero. For completely lambertian surfaces this equation works well since intensity values of one correspond to flat regions. However, for darker materials the maximum intensity is less than one. This poses a problem for SFS algorithms since an intensity less than one implies that the region is sloped. Our technique solves this problem by shifting the intensities for a given region to always have a range from one to zero, which then forces the brightest point of any material to correlate to a flat region.

The preprocess to our FM-SFS technique proceeds as follows.

1. Perform image segmentation to separate the image into materials. See Section 3.2.1.
2. When the segmentation creates a new material, compute the maximum intensity for that material. See Section 3.2.2.
3. Use the maximum intensity for each material and compute transformed intensity values for each region.

Once this process is completed the resulting image will have all intensity values based on a range from one to zero.

3.2.1 Image Segmentation

In order to handle multiple materials, our algorithm relies on an initial preprocessing step of segmenting out differing regions. We use a simple image segmentation approach that works on greedily building up neighboring regions of similar pixel values. In brief, we propagate a queue with all potential peaks as seeds of their own material. Peaks are pulled off the queue and a breadth first traversal is used to grow regions by queueing neighboring pixels if their material values are similar enough to the head of the queue. If a pixel has multiple material regions it could join, it joins the material to which it most closely matches intensity. Any neighboring pixels that are not within the material threshold of their neighbors are processed last in a separate queue. These initially unmatched pixels are processed by again checking for possible neighboring regions to join (after the initial regions have been grown), otherwise a new material is created and propagation proceeds as before. See Figures 3, 4, 5 for examples of the image segmentations created with this method.

3.2.2 Computing the Intensity Range for a Material

Once regions have been created, we must compute the intensity range for each material (or region). Computing the maximum intensity for a material that contains a peak is straight forward, for example, our algorithm simply iterates over every pixel to determine the maximum intensity value. However, if a region does not have a peak, the point with the highest intensity value will not correlate to a flat area as when a peak is present. Most

regions do have peaks, however, to determine maximum intensity values for regions without peaks we have to use information about neighboring regions. For example, we use the below algorithm, assuming the following preconditions: Region A neighbors region B; Region A has a computed maximum intensity, region B does not and assuming that the transition between region A and region B is continuous.

Algorithm:

1. Find two border points on the transition between region A and region B.
2. Whereas the transition between the regions is continuous we formulate the following linear constraint about the intensity values of the points:

$$\frac{A_{bp}}{A_i} = \frac{B_{bp}}{B_i} \tag{2}$$

where A_{bp} is the border point on A and A_i is the brightest point in region A, likewise for B. Since the intensity value of each border point are known and since the brightest point on A is known, using the above constraint, we can solve for what the brightest point on B would be.

With image segmentation and intensity range shifting, our algorithm addresses the challenge of multiple materials. This process does increase the algorithms processing time by approximately 75 percent, however, it yields much better surface results when various materials are present. See Figures 3 and 4.

Note that the challenge of specularity is not that far from the challenge of multiple materials. Like solving for multiple materials we can think of solving for specularity as a preprocessing step. Also like multiple materials, the problem lies within the intensity range. Again our desired intensity range is from one to zero for a lambertian surface with only a diffuse component in the material. If we are dealing with a material with a specular component, this can cause bright regions that are much larger than would have been perceived without a specular component. We found that our multiple materials solution has some capacity to handle the specular component of lighting by breaking the material into two regions. The region affected by the specular component and the region not affected by the specular component. We can then treat these regions as different materials and solve for them separately. While this method is still an approximation, it is an improvement over most SFS techniques which simply ignore specular components.

3.3 Summary of Algorithm

The following is an outline for the combined technique that takes advantage of all the solutions discussed previously. **Preconditions:**

1. The surface to be modeled must be captured into two images with slightly different view directions.
2. The distance between the views (camera position) as well as the focal length must be known.
3. There is a single light source and the scene is captured from a far enough distance away such that the light source is effectively at the optical center of each view.

Algorithm:

1. Process one of the two images for possible peaks as described above.
2. Use stereopsis to compute depths of the possible peaks as in Section 3.1.
3. Mark each peak as coming from a different material.
4. Segment the image to create any multiple material regions as described in Section 3.2.1.
5. Compute shifted intensity range of each material as in Section 3.2.2.
6. Perform FM-SFS with the following modification:
 - (a) When pixels are removed from the priority queue modify their intensity values to fit in the intensity range of their material as described in Section 3.2.

The final result will be a surface constructed of an object that can contain multiple peaks, multiple materials, and specular regions.

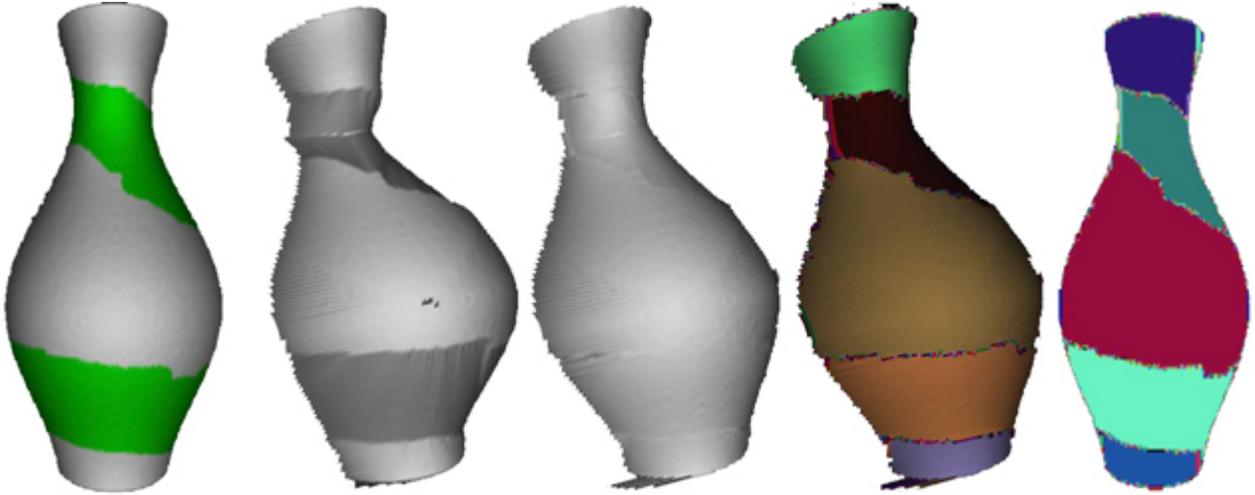


Figure 3. On the left is the input image of a vase composed of multiple materials. Next is the surface constructed without handling multiple materials. Then in the middle we show the surface constructed with handling of multiple materials enabled. On the right, we show the same vase surface showing where the algorithm perceived different materials (on both the reconstructed surface and input image).

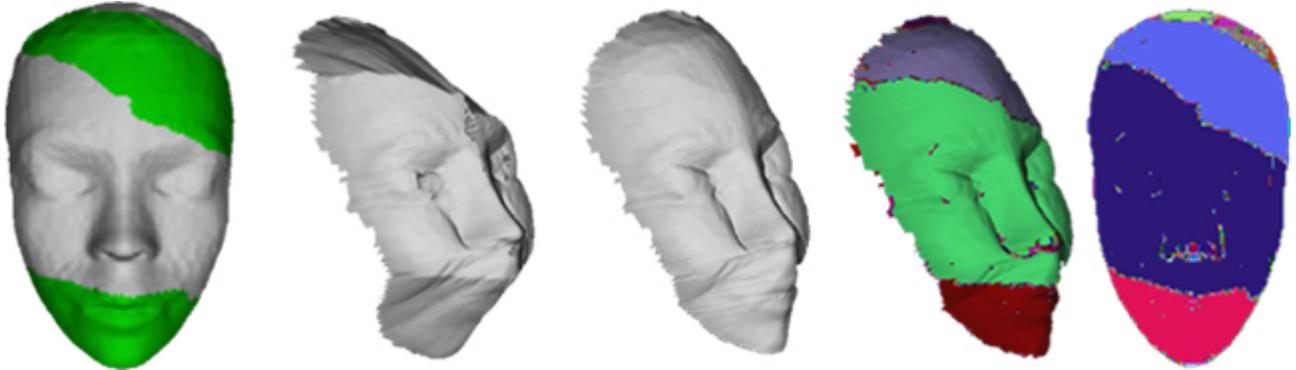


Figure 4. On the left is the input image of a synthetic face dataset composed of multiple materials. Next is the surface constructed without handling multiple materials. Then in the middle we show the surface constructed with handling of multiple materials enabled. On the right, we show the same face surface showing where the algorithm perceived different materials (on both the reconstructed surface and input image).

4. RESULTS

The implementation of our algorithm was done in C# using OpenGL. Tests were run on an AMD Athlon XP 3200+ 2.0 GHz processor using 320 by 240 input images. Figure 3 highlights our reconstruction of a vase surface from two input images with multiple materials. The figure illustrates both our autonomous SFS technique with handling of specular highlights and multiple materials disabled and our complete algorithm. Stereopsis was used on multiple points to compute absolute depth then FM-SFS was executed to construct the surface. As can be seen in Figure 3, without handling multiple materials the green material in the input images produce extremely sloped regions on the constructed surface. In the case where the full algorithm was used, the handling of multiple materials modified the green materials intensity range to use a one to zero scale, and thus, the surface closely resembles the vase where the input images were captured from. The segmentation of the surface to determine where our algorithm perceived different materials is likewise visualized in Figure 3. Runtimes of our algorithm to generate the vase surface were approximately 0.09 seconds for the version without multiple materials and about 0.14 seconds computing with multiple materials.

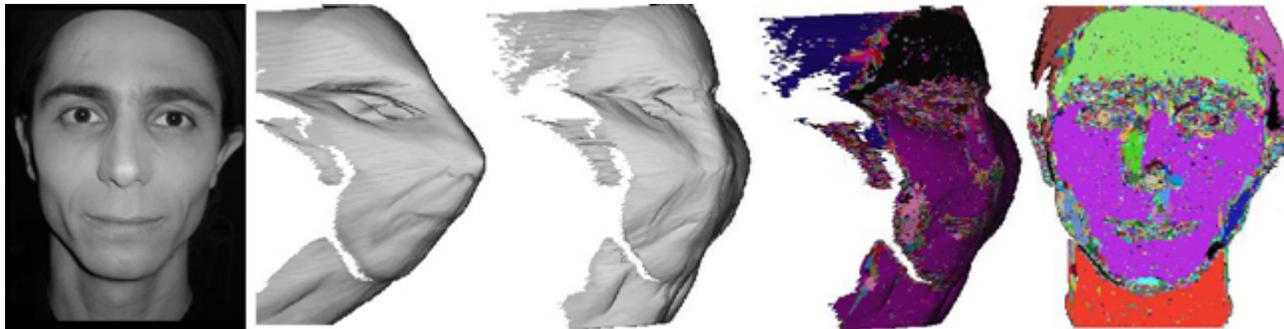


Figure 5. On the left is the input image of a real face image composed of multiple materials. Next is the surface constructed without handling multiple materials. Then in the middle we show the surface constructed with handling of multiple materials enabled. On the right, we show the same face surface showing where the algorithm perceived different materials (on both the reconstructed surface and input image).

In addition to the vase acquisition we provide two more acquisitions which demonstrate the strengths and weaknesses of our approach. These results are shown in Figures 4 and 5 and show our full algorithm run on both synthetic and real face data. Figure 4 shows the results of running our full algorithm on simulated face data with multiple materials shown in grey and green. Runtimes of our algorithm to generate the face surface were approximately 0.07 seconds for the version without multiple materials and approximately 0.13 seconds computing with multiple materials. Again with the synthetic face image, we see the importance of the multiple materials computations as the version run without it suffers from regions with extreme slopes.

When running our FM-SFS with multiple materials on real images like in Figure 5 we see an improvement over unmodified FM-SFS, however, since this data comes from a single image, yet contains multiple peaks, we can no longer rely on stereo depth information to compute accurate peak heights for differing regions. This data demonstrates the usefulness of multiple materials and highlights the utility of stereo correspondence to enhance the output for surfaces with multiple peaks. Although both results do not generate a perfect surface, when our algorithm handles multiple materials, the generated surface more closely matches a desired output. Note that this result also demonstrates the algorithms contributions in terms of specular highlights, by separating out the tip of the nose as a separate region, decreasing its peak height. Runtimes of our algorithm to generate the real face surface were approximately 0.13 seconds without multiple materials and about 0.3 seconds computing with multiple materials.

5. CONCLUSION AND FUTURE WORK

We have shown how techniques from other fields can be used to overcome some of the weakness of SFS. To achieve our goal of an autonomous SFS solution that performs well on real images we have identified the major challenges that occur in real images. We chose to address the challenges of ambiguity, multiple materials, and specularity because previous assumptions that address these problems have been too limiting. Our method uses stereopsis and image segmentation to address these open challenges of SFS. We propose a hybrid technique that allows for an autonomous solution to the SFS problem that can handle multiple peaks, multiple materials, and address specular components.

Our method performs reasonably well when objects are simple enough that our correspondence succeeds and materials boundaries are severe. While we have greatly reduced the reliance on traditional SFS assumptions we have introduced the need for two images rather than one. This restriction is acceptable for newly captured data, but for previous images where only a single image is available the use of stereopsis will not be possible. However, our handling of multiple materials is not coupled with stereopsis and could still be applied to these scenarios (see Figure 5).

Future work includes exploring the applicability of a hybrid method to traditional stereo vision problems. We hope that our hybrid approach of combining SFS, stereopsis, and image segmentation will allow us to move away

from traditional per pixel stereopsis and move towards a much more time efficient per region stereopsis. The idea of per region stereopsis should be researched further using this work as its foundation. Further improvements should be explored in terms of peak detection, as our current algorithm yields an excessively large number of possible peaks and since correspondence in this algorithm is weak, completely autonomous results are sometimes unreliable.

ACKNOWLEDGMENTS

We would like to thank Tawa White and Clarke Turner of Calpoly for their assistance with editing. Real and synthetic images for the shape from shading problem, courtesy of Emmanuel Prados at INRIA Rhone-Alpe.

REFERENCES

1. E. Prados and O. Faugeras, "Shape from shading," in *Mathematical Models in Computer Vision: The Handbook*, N. Paragios, Y. Chen, and O. Faugeras, eds., Springer, 2005.
2. M. Levoy, B. Curless, S. Rusinkiewicz, D. Koller, L. Pereira, M. Ginzton, S. Anderson, J. Davis, J. Ginsberg, J. Shade, and D. Fulk, "The digital Michelangelo project: 3D scanning of large statues," in *ACM SIGGRAPH 2000*, pp. 343–352, 2000.
3. B. T. Phong, "Illumination for computer generated images," *Communications of the ACM* **18**, pp. 311–317, 1975.
4. A. Ortiz and G. Oliver, "Shape from shading for multiple albedo images," in *Proceedings of the 15th International Conference in Pattern Recognition*, 2000.
5. R. Zhang, P. S. Tsai, J. E. Cryer, and M. Shah, "Shape-from-shading: a survey," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 690–706, 1993.
6. E. Prados and O. Faugeras, "Perspective shape from shading and viscosity solutions," in *Proceedings of the Intl. Conference on Computer Vision*, **2**, pp. 826–831, 2003.
7. S. I. Cho, H. Saito, and S. Ozawa, "A divide-and-conquer strategy in shape from shading problem," in *Computer Vision and Pattern Recognition*, pp. 413–419, 1997.
8. P. N. Bellumeur, D. J. Kriegman, and A. L. Yuille, "The bas-relief ambiguity," in *JCV*, **35(1)**, pp. 33–44, 1999.
9. S. Y. Yuen, Y. Y. Tsui, Y. W. Leung, and R. M. M. Chen, "Fast marching method for shape from shading under perspective projection," in *Proceedings of VIIP02*, pp. 584–589, 2002.
10. B. Horn, *Shape from shading: A Method for obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, MIT, 1970.
11. B. Horn, *Obtaining shape from shading information*, McGraw-Hill, 1975.
12. G. Zeng, Y. Matsushita, L. Quan, and H. Y. Shum, "Interactive shape from shading," in *IEEE Proceedings of Computer Vision and Pattern Recognition*, pp. 690–706, 1999.
13. M. F. J. Durou and M. Sagona, "A survey of numerical methods for shape from shading," in *Rapport de recherche 2004-2-R, IRIT*, 2004.
14. R. Kozera, "An overview of the shape from shading problem," *Machine Graphics and Vision*, pp. 291–312, 1998.
15. B. Horn, "Height and gradient from shading," in *International Journal of Computer Vision*, pp. 37–75, 1990.
16. P. S. Tsai and M. Shah, "Shape from shading using linear approximation," in *Image and Vision Computing*, pp. 487–498, 1994.
17. R. Kimmel and J. Sethian, "Optimal algorithm for shape from shading and path planning," in *Journal of Mathematical Imaging and Vision*, pp. 237–244, 2001.
18. D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *International Journal of Computer Vision* **47**, pp. 7–42, 2002.