A Robust Iterative Shape-From-Shading Algorithm with Modified Transforming Matrix

Osamu Ikeda
Faculty of Engineering, Takushoku University
815-1 Tate-machi, Hachioji, Tokyo 193-0985 Japan

Abstract

We present a new simple iterative shape-from-shading algorithm, which gives greater numerical stability and as a result more accurate shapes than the previous one. The Jacobi's iterative method is applied to the difference between the image and the reflectance function of the three depth parameters to get an iterative relation, which is then modified so that all the eigenvalues of the inverting matrix consist of three squared terms. This prevents the determinant of the matrix from being null to a great degree, resulting in a more accuracy of the shape estimate. We also rotate the coordinates by an arbitrary angle, when needed, to improve the reconstruction with no such shape distortions as stripe-like ones and with a resolution enhancement. Computer experiments were made using several synthetic and real images to show its effectiveness.

1 Introduction

Since Horn initiated the research on shape-from-shading [1], significant developments have been made [2], [3]. They may be classified to local [4], [5], minimization [6], [7], linear [8], [9], propagation [10], [11], and deformable model based [12] approaches. Minimization approaches are based upon minimizing a given energy criterion to estimate the shape. Zheng and Chellappa [6], for example, introduced image gradient and integrability constraints to obtain fine details. Linear approaches linearize the reflectance map in tilts or depth. Tsai and Shah [8] linearized the map in the depth and used the Jacobi iterative technique to derive an iterative relation. We presented a nonlinear approach by generalizing their method and by carrying out the bi-directional estimation to minimize shape distortions [13]. Propagation approaches obtain a shape starting from some initial curve, which uses such special points as the brightest or the darkest. Kimmel et al., for example, show that good shape reconstruction is possible using some boundary conditions [11]. The deformable model based method literally uses such a model in combination with the scene irradiance as a constraint [12].

These approaches have pursued accuracy in reconstructed shape, but numerical stability has been the issue as well. Horn and Brooks addressed it in the minimization approach [14]. The above-mentioned approaches are also vulnerable to numerical

BMVC 2003 doi:10.5244/C.17.54

instability. To avoid it, the iteration is limited literally [6] or effectively [8] in number, or the image is normalized to a value less than unity [11], [13]. The limitation may often deteriorate the accuracy. In our previous algorithm the bi-directional estimation was introduced to partially compensate for the deterioration. The deformable model based method appears to be more stable, but it appears to lack accuracy for complex images such as real ones.

In this paper we improve our previous algorithm in numerical stability, which is very simple to implement and successfully reconstructs shapes from images for illuminant vectors other than $\approx(a,-a,1)$. We modify the inverting matrix in the iterative relation to make it more robust, where the matrix transforms the change in depth to that in image intensity. We also rotate the coordinates by an arbitrary angle to maximize the quality of shape. Computer experiments show that the new algorithm is able to give more accurate shapes for a variety of objects and illuminating conditions.

2 Principle

The object is illuminated from a direction of light to obtain a shading image. Given an appropriate reflectance function $R(p,q)$, it may be equal to the image $I(x,y)$:

$$R(p,q) = I(x,y) \tag{1}$$

where $x, y = 1, \ldots, N$, p and q are local tilts of the surface, and the image is normalized to unity. Here we assume that the object has parts whose surface normal **P** is parallel to the illuminant vector **S**. The two unit vectors are given by their components as

$$\mathbf{P} = \frac{(p,q,1)}{\sqrt{p^2+q^2+1}} \tag{2}$$

$$\mathbf{S} = \frac{(s,t,1)}{\sqrt{s^2+t^2+1}} \tag{3}$$

Using the depth $z(x,y)$ of the object, p and q are given by $-\partial z/\partial x$ and $-\partial z/\partial y$, respectively. This description is different from that used by Woodham [15], but it may be more intuitive and convenient. Then, if we assume the *Lambertian* surface, the reflectance function, normalized by the *albedo*, is given by their scalar product:

$$R(p,q) = \frac{1+ps+qt}{\sqrt{1+p^2+q^2}\sqrt{1+s^2+t^2}} \tag{4}$$

The normalization stands on the same assumption as that for the image. If we use the following discrete forms for p and q

$$p = z(x-1,y) - z(x,y), \quad q = z(x,y-1) - z(x,y) \tag{5}$$

then R can be regarded to be a function of three variables, $z(x,y)$, $z(x\text{-}1,y)$ and $z(x,y\text{-}1)$. We define the function $f(x,y)$ as the difference between the image and the reflectance function:

$$f(x,y) \equiv I(x,y) - R(p,q) \tag{6}$$

Then, applying the Jacobi's iterative method to the function f results in the relation:

$$\begin{aligned} -f(x,y)^{(n-1)} &= \left(\frac{\partial f(x,y)}{\partial z(x,y)}\right)^{(n-1)} \left(z(x,y)^{(n)} - z(x,y)^{(n-1)}\right) \\ &+ \left(\frac{\partial f(x,y)}{\partial z(x-1,y)}\right)^{(n-1)} \left(z(x-1,y)^{(n)} - z(x-1,y)^{(n-1)}\right) \\ &+ \left(\frac{\partial f(x,y)}{\partial z(x,y-1)}\right)^{(n-1)} \left(z(x,y-1)^{(n)} - z(x,y-1)^{(n-1)}\right) \end{aligned} \tag{7}$$

where n is the number of iterations. Eq. (7) can be rewritten in matrix form as

$$-\mathbf{f}^{(n-1)} = \mathbf{g}^{(n-1)}(\mathbf{z}^{(n)} - \mathbf{z}^{(n-1)}),\ n = 1, 2, \ldots, \tag{8}$$

where $\mathbf{f}$ is a vector of N^2 elements of $f(x,y)$, $\mathbf{z}$ is a vector of N^2 elements of $z(x, y)$, and $\mathbf{g}$ is a matrix of N^2xN^2 elements that are made of the terms of $\partial f(x,y)/\partial z(x,y)$, $\partial f(x,y)/\partial z(x\text{-}1,y)$ and $\partial f(x,y)/\partial z(x,y\text{-}1)$.

In our previous algorithm, the shape is reconstructed iteratively using the relation in Eq. (8), beginning with null values $\mathbf{z}^{(0)} = \mathbf{0}$. In this case the determinant of $\mathbf{g}$ is given by the product of all its diagonal terms given from Eq. (7) as

$$\lambda(x,y) = \frac{\partial f(x,y)}{\partial z(x,y)},\ x, y = 1,\ldots,N \tag{9}$$

They become null for the image parts where $\mathbf{P}$ is parallel to $\mathbf{S}$, as seen by inserting Eq. (4) in Eq. (6) and the resulting expression in Eq. (9). Existence of even one such element nullifies the determinant. In this sense, numerical instability is unavoidable in this method if the object has such parts.

Eq. (8) can nominally be rewritten for $\mathbf{z}^{(n)}$ as

$$\mathbf{z}^{(n)} = \mathbf{z}^{(n-1)} - \mathbf{G}^{(n-1)^{-1}}\mathbf{F}^{(n-1)},\ n = 1,2,\ldots \tag{10}$$

$$\mathbf{G}^{(n-1)} = \mathbf{g}^{(n-1)^T}\mathbf{g}^{(n-1)} \tag{11}$$

$$\mathbf{F}^{(n-1)} = \mathbf{g}^{(n-1)^T}\mathbf{f}^{(n-1)} \tag{12}$$

Here let $\mathbf{z}'$, $\mathbf{G}'$ and $\mathbf{F}'$ have the elements of $\mathbf{z}$, $\mathbf{G}$ and $\mathbf{F}$, respectively, in a reduced shape reconstruction area of $1 \le x \le N\text{-}1$ and $1 \le y \le N\text{-}1$. Then Eq. (10) is rewritten as

$$\mathbf{z}'^{(n)} = \mathbf{z}'^{(n-1)} - (1/D)\mathbf{G}'^{(n-1)^{-1}}\mathbf{F}'^{(n-1)},\ n = 1,2,\ldots \tag{13}$$

where D is a newly introduced factor of de-acceleration. The eigenvalues and the elements of $\mathbf{F}'$, $F'(x,y)$, are given, respectively, by

$$\lambda(x,y) = \left(\frac{\partial f(x,y)}{\partial z(x,y)}\right)^2 + \left(\frac{\partial f(x+1,y)}{\partial z(x,y)}\right)^2 + \left(\frac{\partial f(x,y+1)}{\partial z(x,y)}\right)^2 \tag{14}$$
$$\text{for } 1 \le x \le N-1,\ 1 \le y \le N-1$$

$$F'(x,y)=\frac{\partial f(x,y)}{\partial z(x,y)}f(x,y)+\frac{\partial f(x+1,y)}{\partial z(x,y)}f(x+1,y)+\frac{\partial f(x,y+1)}{\partial z(x,y)}f(x,y+1) \quad \text{for } 1\le x\le N-1, 1\le y\le N-1 \tag{15}$$

First, it is seen that the eigenvalues consist of three squared terms in contrast to one in Eq. (9). When we use $\mathbf{z}^{(0)}=\mathbf{0}$, those in Eq. (14) are not null except for $\mathbf{S}=(0,0,1)$, while those in Eq. (9) are null for the class of (*a, -a,* 1), where *a* is a positive or negative real number including zero. Hence we may be able to expect greater stability with the modification. Second, it may be seen that the seven depth terms, $z(x\text{-}1,y+1)$, $z(x,y+1)$, $z(x\text{-}1,y)$, $z(x,y)$, $z(x+1,y)$, $z(x,y\text{-}1)$ and $z(x+1,y\text{-}1)$, contribute to both the eigenvalue in Eq. (14) and $F'(x,y)$ in Eq. (15) and that they are distributed fairly symmetric around (x,y). This property may make us expect to reconstruct similar shapes in the forward and backward directions on the coordinates. Third, it is seen that we still need **G** and **F** in the entire area of $1\le x\le N$ and $1\le y\le N$ to get **G′** and **F′**. Since $z(x,y)$ are not estimated along the two boundary lines, (x,N) and (y,N), where $1\le x\le N$ and $1\le y\le N$, we need to assume values of p and q along these two lines in addition to those along the other two boundary lines, $(x,1)$ and $(y,1)$, where $1\le x\le N$ and $1\le y\le N$. We use the two assumptions depending on whether or not the image varies along the boundaries. On one hand, for the case where the image varies, $p=0$ is assumed along the vertical boundary line and $q=0$ along the horizontal one. On the other hand, for the case where the image does not vary, we assume that $p=-z(1,y)$ or $z(N\text{-}1,y)$ along the vertical boundary line and $q=-z(x,1)$ or $z(x,N\text{-}1)$ along the horizontal line in correspondence to $\mathbf{z}^{(0)}=\mathbf{0}$.

3 Rotation of Coordinates

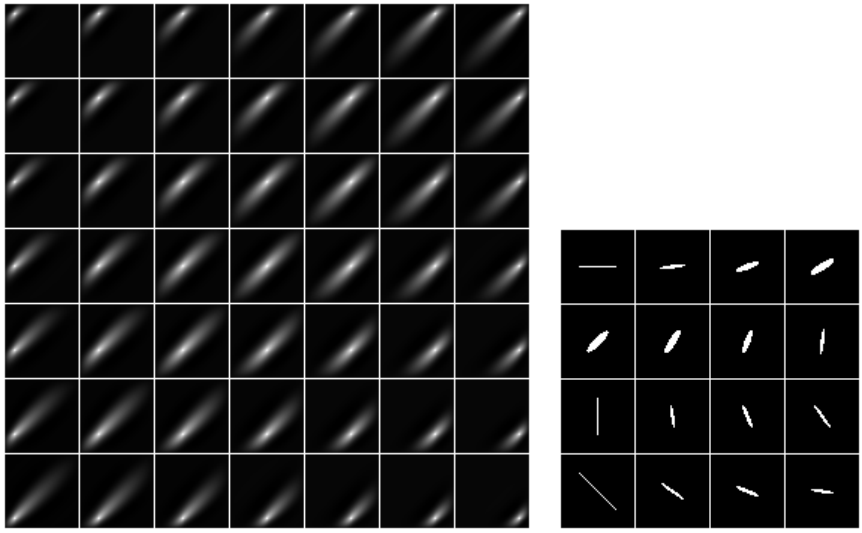
In [13] we proposed to estimate the shape in the forward and backward directions to reduce the distortions. This can be extended to rotating the coordinates by an arbitrary angle to maximize the quality of shape. To explain it, first, let us rewrite Eq. (13) as

$$z'^{(n)}(x,y)=z'^{(n-1)}(x,y)-(1/D)\sum_{x'}\sum_{y'}h^{(n-1)}(x',y';x,y,\sigma,\tau)F'^{(n-1)}(x',y') \quad n=1,2,\ldots \tag{16}$$

When we use $\mathbf{z}^{(0)}=\mathbf{0}$, the function h for $n=1$ may approximately be given by

$$h^{(0)}(x',y';x,y,\sigma,\tau)=A^{(0)}(x,y,\sigma,\tau)p_f^{(0)}(x-x',y-y';\tau) \tag{17}$$

where $A^{(0)}$ takes a maximum at the center of the coordinates and gradually decreases towards the borders, and it varies depending on σ and τ. $p_f^{(0)}$, normalized to unity, has very similar patterns for a variety of (x,y) values, so that it may be regarded to be a space-invariant function. $p_f^{(0)}$ changes with respect to τ but not so with respect to σ, as far as numerical evaluation is concerned. For $n>1$ the function h varies depending on (x,y) so that it is not space-invariant any more, as will be shown, but its basic patterns are still persistent to a varying degree depending on n, (x,y) and the image. It may be seen from Eqs. (16) and (17) that $p_f^{(0)}$ represents the characteristics of the reconstruction system. In view of this, the function may be desired, in general, to be isotropic and to be a monotonically decreasing function with a peak at $x'=x$ and $y'=y$, where the significant area of the function results in an effective averaging area for $F'(x,y)$.



(a) (b)

Fig. 1 (a) Patterns of $p_f^{(0)}$ in Eq. (17) for (x,y) = (6,42), (12,42), …, (42,42), …, (42,6) from the top left to bottom right, where τ = 45, and (b) those for τ = 0, 11.25, 22.5, …, 168.75 from the top left to bottom right, where (x,y) = (25,25).

Some patterns of $p_f^{(0)}$ are shown as a function of (x,y) in Fig. 1(a) for τ = 45 degrees, and binary patterns drawn in white for the parts of more than 50% magnitude of the maximum are given as a function of τ in Fig. 1(b) for $(x,y) = (N/2, N/2)$, N = 50. In Fig. 1(b), where τ ranges from 0 to 168.75 degrees, the pattern for τ + 180 is the same as that for τ. It is seen from Fig. 1(a) that the patterns are similar to each other, and it is seen from Fig. 1(b) that the pattern for $\tau \approx 45$ is the best among them. Hence, it may be reasonable to rotate the coordinates so that the lighting direction is equivalently $\tau \approx 45$. Otherwise, typically the line pattern may generate stripe-like shape noise, as will be shown.

When the coordinates are rotated by an angle other then a multiple of 90 degrees, an image value has to be assumed where the image does not exist. In our case it is given so that the resulting shape is flat. The $p_f(x,y;\tau)$ pattern for the case of $\tau \approx 45$ implies that possible shape distortions may also accumulate along the direction. So when two shapes reconstructed in the opposite directions are averaged, we may expect to use, in general, the weights that vary along this direction.

The overall procedure is as follows:

(i) When $\mathbf{S}=(\sigma, \tau)$ has a tilt angle outside the desired range, say, $30<\tau<60$, we rotate the coordinates by τ' so that it holds that $30<\tau+\tau'<60$.

(ii) Two shapes are reconstructed in two opposite directions on the rotated coordinates.

(iii) Average gradients in x and y are evaluated for the shapes, to determine in which direction the weights on the two shapes linearly vary in the averaging.

(iv) The two shapes are averaged with the weights and the image is mapped pixel by pixel on the shape.

4 Computer Experiments

Four synthetic images and one real image were used. The four shapes and their synthetic images for **S**=(5,5,7) or (5,5,2) are shown in Fig. 2; a semi-sphere, the Mozart sculpture [16], a ring, and a computer mouse. In Fig. 2(c) the cross-sectional surface profiles of the center part and the ring are given by half the periods of the cosine functions. In Fig. 2(d) the shape of the mouse was measured using a laser range scanner. As you can observe, the measured shape has some structural noise and, as a result, the shading image is noisy. All of them have the size 50x50 elements. In the estimation the iteration was basically stopped when the gradients of the shape being reconstructed took jumps, to obtain smooth shapes in a shorter time. Since, the shape tends to be jagged after a certain number of iterations, although the shape remains roughly the same in many cases. In some cases, however, the shape is significantly distorted without the stopping.

First in Fig. 3, reconstructed shapes are compared between the new method and the previous one for two of the objects. It is seen that the difference in shape is significant between the two estimates for the previous method, while the two shapes are relatively similar for the new method. Averaging them may reduce the distortions to a great degree, but it would be much more desirable to get as accurate shapes as possible for each of the estimates. The corresponding patterns of $h^{(n-1)}$ to the case of the semi-sphere in Fig. 3(a) are shown in Fig. 4. It is seen that they are different from those in Fig. 1, reflecting the image, but still the basic patterns in Fig. 1 are persistent.

Shapes obtained with the new algorithm for the four shading images in Fig. 2 are given in Fig. 5, where they are averaged ones and the images are mapped pixel by pixel on the shapes as textures. In the case of the semi-sphere, where averaging was made using the weights that vary linearly along the line $y=x$, the gradients of the estimated shape are not large enough, judging from the part meeting the flat background. In the case of the Mozart, where there is almost no difference between the averaged shapes obtained with the optimal weights that linearly vary in x and with the weights that vary linearly along $y=x$, the part of the chest is lower than the original shape. This may be correlated to the property of the h pattern; that is, its magnitude decreases towards the boundaries. In the case of the ring, where the optimal weights vary linearly along $y=x$, the reconstructed ring is lower in height than the center part due to the same property of the h and a limited spatial resolution that may also have to do with the h pattern. In the case of the mouse, the shape obtained with the optimal weights that linearly vary in y is very close to the one obtained with the weights that linearly vary along $y=x$. We can observe that the shape is relatively good, although it has some minor distortions.

Figure 6 shows the effects of rotating the coordinates and averaging the two estimates on improving the shape for a complex real image. The lighting direction for the real image in Fig. 6(a) was estimated to have σ=58 and τ=5 in degree using the method presented in [6]. Then, two shapes were reconstructed, and they were averaged with the optimal weights that linearly vary in y. The result in Fig. 6(b) shows that stripe-like noise is noticeable as anticipated. Next, the coordinates were rotated clockwise by 40 degrees to get the shape in Fig. 6(c), where the optimal weights on the rotated coordinates vary linearly in x, although the shape obtained with the weights that vary along $y=x$ correspondingly to the h pattern is very close to the one in Fig. 6(c). The image is mapped pixel-by-pixel on the shape in Fig. 6(d) as texture to give a three-dimensional representation of the shape. Two shapes obtained before the

averaging with and without mapping of the image are also shown in Fig. 6(e)-(h). As clearly seen from the comparison of Fig. 6(f) and (h) with (d), the averaging is very significant in the new algorithm, too, when it comes to a complex real image.

5 Conclusions

We presented a robust iterative shape-from-shading algorithm, which is easy to implement, more accurate, and applicable to a wider range of objects and illuminating conditions. We also showed the effects of rotating the coordinates on improving the shape. Computer experiments showed the usefulness of the new algorithm. Shapes obtainable with the new algorithm, however, still appear to lack accuracy, so more sophisticated an algorithm is now under study.

References

[1] B.K.P. Horn, Obtaining Shape from Shading Information, in *The Psychology of Computer Vision* P.H.Winston (ed.) (New York: McGraw Hill, 1975), 115-155.

[2] B.K.P. Horn and M.J. Brooks, *Shape from Shading* (Cambridge, MA: MIT Press, 1989).

[3] R. Zhang, P. Tsai, J.E. Cryer, and M. Shah, Shape from Shading: A Survey, *IEEE Trans. PAMI,* 21(8), 1999, 690-705.

[4] P. Pentland, Local shading analysis, *IEEE Trans. PAMI,* 6(2), 1984, 170-187.

[5] C.H. Lee and A. Rosenfeld, Improved Methods of estimating Shape from Shading Using the Light Source Coordinate System, *Artificial Intelligence,* 26(1), 1985, 125-143.

[6] Q. Zheng and R. Chellappa, Estimation of Illuminant Direction, Albedo, and Shape from Shading, *IEEE Trans. PAMI,* 13(7), 1991, 680-702.

[7] P. L. Worthington and E. R. Hancock, New Constraints on Data-Closeness and Needle Map Consistency for Shape-from- Shading, *IEEE Trans. PAMI,* 21(12), 1999, 1250-1267.

[8] P. S. Tsai and M. Shah, Shape from Shading Using Linear Approximation, *J. Imaging and Vision Computing,* 12(8), 1994, 487-498.

[9] A. Pentland, Shape Information from Shading: A Theory about Human Perception, *Proc. Int'l Conf. Computer Vision*, 1988, 404-413.

[10] M. Bichsel and A. Pentland, A Simple Algorithm for Shape from Shading, *Proc. CVPR*, 1992, 459-465.

[11] R. Kimmel and A.M. Bruckstein, Tracking Level Sets by Level Sets: A Method for Solving Shape from Shading Problem, *CVIU,* 62(1), 1995, 47-58.

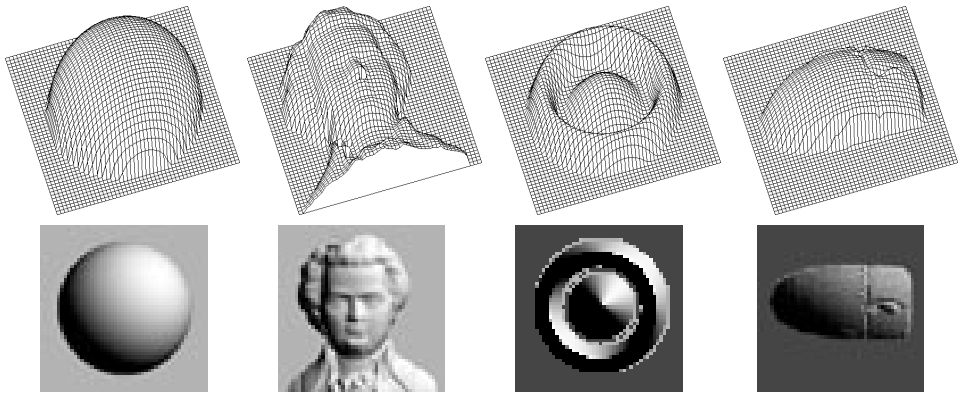
[12] D.Samaras and D.Metaxas, Incorporating Illumination Constraints in Deformable Models, *CVPR*, 1998, 322-329.

[13] O. Ikeda, A Novel Shape-From-Shading Algorithm Using Jacobi Iterative Method and Bi-Directional Estimation, *Proc. IASTED CGIM*, 2002, 56-61.

[14] B.K.P. Horn and M.J. Brooks, The Variational Approach to Shape from Shading, *CVGIP*, 33(2), 1986, 174-208.

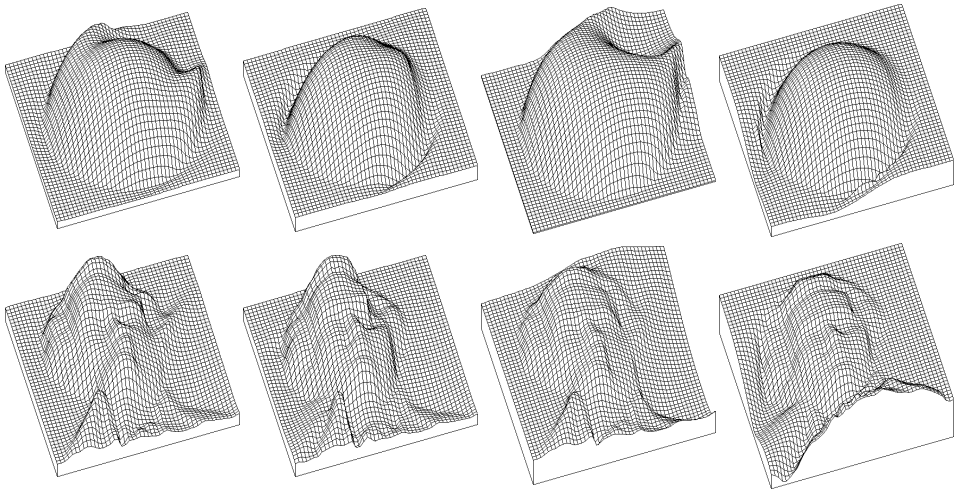
[15] R.J. Woodham, Photometric Method for Determining Surface Orientation from Multiple Images, in *Shape from Shading*(Cambridge, MA: MIT Press, 1989).

[16] eustis.cs.ucf.edu (132.170.108.42)



(a) semi-sphere (5,5,7) (b) Mozart (5,5,7) (c) ring (5,5,2) (d) mouse (5,5,2)

Fig. 2 Four shapes and their synthetic shading images with the values of $\mathbf{S} = (S_x, S_y, S_z)$.



(a) Eq. (13), forward (b) Eq. (13), backward (c) Eq. (8), forward (d) Eq. (8), backward

Fig. 3 Comparison of shapes reconstructed with the new algorithm using Eq. (13) and the previous one using Eq. (8) for two images, where forward and backward mean the directions on the coordinates in the estimation.

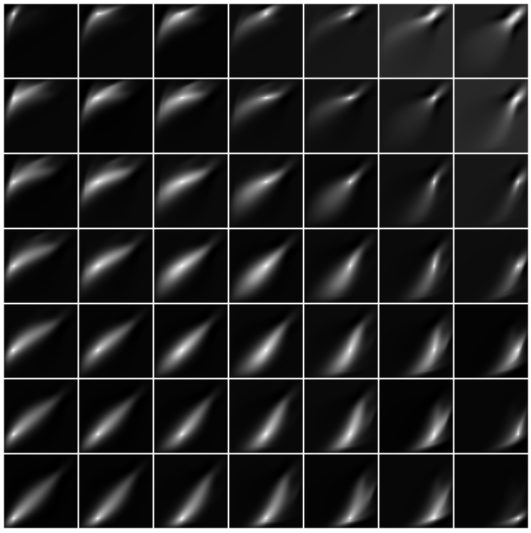


Fig. 4 Normalized patterns of $h^{(n-1)}$ in Eq. (16) for (x,y) = (6,42), (12,42), …, (42,42), …, (42,6) from the top left to bottom right for the semi-sphere image after a number of iterations for which we obtain the shape on the top in Fig. 3(a).

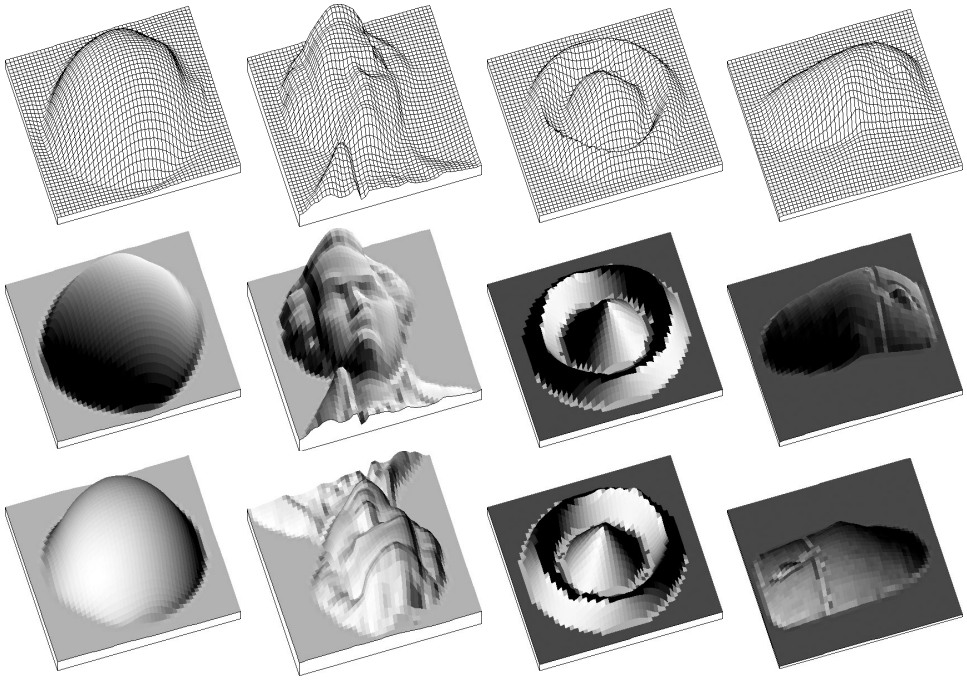


Fig. 5 Reconstructed and averaged shapes (top), front-views of their texture-mapped shapes (medium) and their back-views (bottom). In the case of the ring, the view on the medium row is from the right.

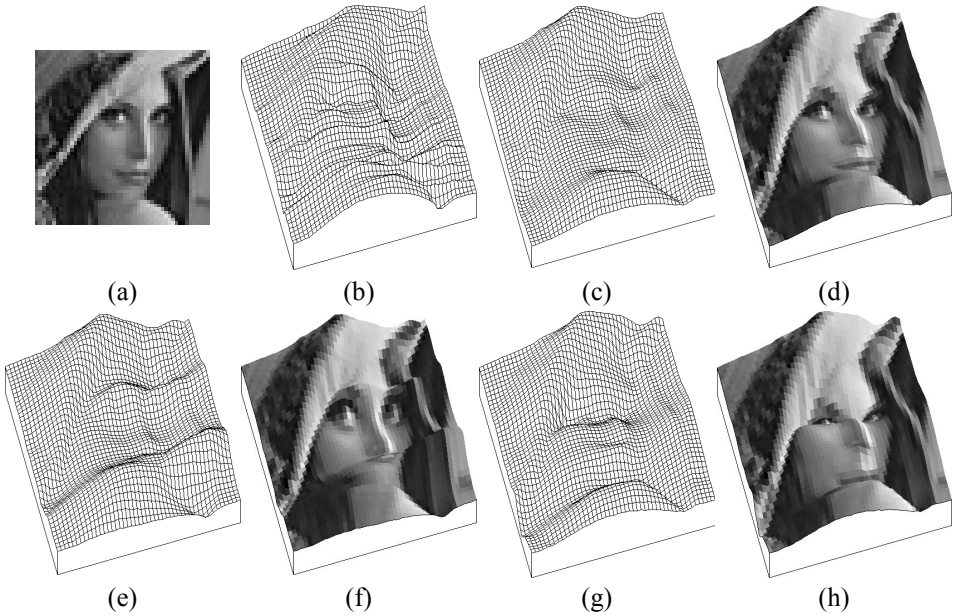


Fig. 6 Effects of rotating the coordinates on improving the shape: (a) image; (b) average of the two reconstructed shapes; (c) average of the two reconstructed shapes after rotating the coordinates clockwise by 40 degrees; (d) the same shape in (c) with the image mapped pixel by pixel; (e) shape reconstructed in the forward direction after rotating the coordinates by 40 degrees; (f) the same shape as in (e) with the image mapped pixel by pixel; (g) shape reconstructed in the backward direction after rotating the coordinates by 40 degrees; and (h) the same shape as in (g) with the image mapped pixel by pixel.