filtering. The simulation of an electronic neural network setup show that the detection of a moving image is not only possible in real-time but also with high resolution and lower level noise. This demonstrates that biologically inspired quasiparallel processing has advantages in processing moving image information, and provides a new way for hardware implementation for motion processing functions. The proposed algorithm which solves the problem of image-background discrimination by binding current motion information with previously acquired afterimages may be have some usefulness in robot vision.

ACKNOWLEDGMENT

The authors are grateful to our anonymous referee for the most appreciated comments and the substantial help in improving grammar of this paper.

REFERENCES

[1] W. Reichardt, M. Egellhaaf, and A. Guo, "Processing of figure and background motion in the visual system of the fly," *Biol. Cybern.*, vol. 61, pp. 327–345, 1989.
[2] P. Burt and G. Sperling, "Time, distance, and feature trade-offs in visual apparent motion," *Psychol. Rev.*, vol. 88, no. 2, pp. 171–155, 1981.
[3] J. P. H. Santen and G. Sperling, "Elaborated Reichardt detectors," *J. Opt. Amer. A*, vol. 2, pp. 300–321, Feb. 1985.
[4] M. A. Fischler and O. Firschein, *Intelligence, the Eye, the Brain and the Computer*. Reading, MA: Addison-Wesley, 1987, pp. 207–238.
[5] T. Poggio, E. B. Gamble, and J. J. Little, "Parallel integration of vision modules," *Science*, vol. 242, pp. 436–440, 1988.
[6] S. Geman and D. Geman, "Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images," in *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 721–741, 1984.
[7] A. Guo and W. Reichardt, "An estimation of time constant of movement detectors," *Naturwissenschaften*, vol. 74, pp. 91, 1987.
[8] S. Neuenschwander and W. Singer, "Long-range synchronization of oscillatory light responses in the cat retina and lateral geniculate nucleus," *Nature*, vol. 379, pp. 728–733, 1996.
[9] P. Buser and M. Imbert (Translated by R. H. Kay), *Vision*. Cambridge, MA: MIT Press, 1992, pp. 172–174.
[10] H. Lu, X. Wang, S. Liu, M. Shi, F. Liu, and A. Guo, "A model for visual-background discrimination by relative movement," *Sci. China*, series C, vol. 40, no. 1, pp. 79–89, 1997.
[11] M. A. Sivilotti, M. A. Mahowald, and C. A. Mead, "Real-time computations using analog CMOS processing arrays," in *1987 Stanford Conf. VLSI*, P. Losleben, Ed. Cambridge, MA: MIT Press, 1987, pp. 295–312.
[12] C. Koch, "Implementing early vision algorithms in analog networks: An overview," in *Applications of Neural Networks*, H. G. Schuster, Ed. New York: VCH, 1992, pp. 3–23.
[13] H. Ogmen, "A neural theory of retino-cortical dynamics," *Neural Networks*, vol. 6, pp. 254–273, 1993.
[14] J. Aloimonos, "Purposive and qualitative active vision," in *Proc. 10th Int. Conf. Pattern Recognition*, Los Alamitos, CA, 1990, pp. 346–360.

Shape from Intensity Gradient

Ruo Zhang and Mubarak Shah

***Abstract*—Unlike existing global shape-from-shading (SFS) algorithms which involve the brightness constraint in their formulation, we propose a new SFS algorithm which replaces the brightness constraint with an intensity gradient constraint. This is a global approach which obtains the solution by the minimization of an error function over the entire image. Through the linearization of the gradient of the reflectance map and the discretization of the surface gradient, the intensity gradient can be expressed as a linear function of the surface height. A quadratic error function, which involves the intensity gradient constraint and the traditional smoothness constraint, is minimized efficiently by solving a sparse linear system using the multigrid technique. Neither the information at singular points nor the information at occluding boundaries is needed for the initialization of the height. Results for real images are presented to show the robustness of the algorithm, and the execution time is demonstrated to prove its efficiency.**

***Index Terms*— Intensity gradient, multigrid technique, physics-based vision, shape-from-shading.**

I. INTRODUCTION

The goal of shape-from-shading (SFS) is to reconstruct the three-dimensional (3-D) shape of an object from its two-dimensional (2-D) intensity image, assuming a proper reflectance map, which models the relationship between the intensity and surface shape, is given. Surface shape can be represented by height, gradient, surface normal, slant and tilt, or curvature. Since the reflectance map is a nonlinear equation in terms of the shape, simplifications are needed in order to restrict the problem. The first simplification, which is also the biggest for most SFS algorithms, is the assumption of diffuse reflection: that surfaces reflect light equally in all directions. The second simplification is that the illumination is from a point light source at infinity. Other simplifications include assuming a known viewing direction, known light source direction, and orthographic projection. Together, these assumptions introduce a simple Lambertian model which describes the intensity in terms of the cosine value of the angle between the surface normal and the light source direction. This model allows for the use of the information at singular points (points which have maximum brightness in the image) and occluding boundaries (the contour on the object where the gradient is nearly 90° to the viewing direction). In order to obtain a correct solution for SFS, the local spherical assumption [10], the brightness constraint [2], [4]–[9], [11], [14], [16], [17], [20], [22], [23], the brightness derivative constraint [23], the smoothness constraint [4], [6]–[9], [11], [20], and the integrability constraint [7], [8], [20], [23] are used in addition to the information at singular points and occluding boundaries [2], [4]–[7], [14], [20].

The spherical assumption approximates the local surface by a spherical patch. The brightness constraint minimizes the error between the reconstructed intensity and the input intensity, and the brightness derivative constraint minimizes the error between the reconstructed intensity derivatives and the input intensity derivatives. The smoothness constraint requires that the reconstructed surface be smooth. The integrability constraint ensures that the reconstructed surface is integrable.

Manuscript received September 22, 1995; revised November 9, 1995.
The authors are with the School of Computer Science, University of Central Florida, Orlando, FL 32816 USA (e-mail: shah@cs.ucf.edu).
Publisher Item Identifier S 1083-4427(99)01460-5.

There are several simple and efficient SFS approaches which use only local intensity information to derive shape. These include Pentland's methods [16], [17], Lee and Rosenfeld's method [10], and Tsai and Shah's method [22].

Pentland [17] solved for the surface slant and tilt, the radius of curvature, and the light source direction using six equations obtained from the intensity, as well as the first and second derivatives of the intensity. His approach can classify a surface into one of five categories: planar, cylindrical, convex, concave, or saddle surface. However, it is limited to surfaces with equal-magnitude principal curvatures.

Lee and Rosenfeld [10] considered the derivatives of the intensity in both the x and y directions, and found that, in the light source coordinate system, the tilt of the surface was the same as the angle of the intensity gradient. This result was obtained by approximating the local surface with a spherical patch. The slant of the surface was obtained under the assumption that the surface has uniform reflectance, and the brightest point on the surface has its normal pointing in the light source direction. The disadvantage of this approach is its limitation to spherical surfaces.

Another approach by Pentland [16] linearized the reflectance map in terms of the surface gradient, through the Taylor's series. By taking the Fourier transform of the linearized brightness equation and considering the relationship between the Fourier transform of the surface gradient and the Fourier transform of the height, the height can be recovered through the inverse Fourier transform of the intensity. Since no smoothness constraint is needed, this algorithm is applicable to complex natural surfaces. However, it has difficulty with images of quadratic and higher order surface reflectance because of the linearization of the reflectance map.

Instead of linearizing the reflectance map in terms of the gradient, Tsai and Shah [22] employed a discrete approximation to the gradient first, then linearized the reflectance map in terms of the height. Consequently, at each pixel, the intensity can be expressed by a linear function of the height at neighboring pixel, and the Jacobi iterative scheme can be applied to solve the entire linear system. This algorithm breaks down when self-shadows exist in the image.

Although local approaches are simple and fast, they have limitations, especially in the case of noisy real images. Therefore, several SFS algorithms use global information to ensure robustness.

The first two global approaches were by Ikeuchi and Horn [8], and Brooks and Horn [4]. Both combined the brightness constraint and the smoothness constraint to form an error function, then minimized it using variational calculus. In his later approach, Horn [7] added the integrability constraint to the error function. To solve the problem of slow convergence for Horn's approach, Szeliski [20] used the hierarchical and preconditioned conjugate gradient descent method to improve the efficiency. Unlike the above algorithms, which involve the recovery of either the surface normal or the surface gradient, Leclerc and Bobick [9] used a discrete approximation of the surface gradient to introduce height into the error function, which consists of the brightness constraint and the smoothness constraint. Then they directly solved for height by taking the derivative of the error function and applying the conjugate gradient technique. All of the above techniques require known shape information at occluding boundaries in order to enforce correct convergence. Leclerc and Bobick's approach needs the height output from stereo as the initial estimate.

Zheng and Chellappa [23] were the first to consider the first derivative of intensity in the variational calculus approach. However, their derivatives were taken along the x and y directions. Their error function contains the brightness constraint, the brightness derivative constraint, and the integrability constraint. For smooth Lambertian surfaces, since the change of intensity is small, the brightness constraint in their error function still dominates. The Taylor's series was applied to linearize the reflectance map, and discrete approximations for surface gradients, and their derivatives, were used. The iterative scheme was implemented using a hierarchical structure to solve for surface height and gradient simultaneously. The initial values for the height and gradient could be zero. Their results showed inaccuracies around the light source direction, which is due to the brightness derivative term.

Lee and Kuo's approach [11] involves the brightness and smoothness constraints. The linearization of the reflectance map was combined with the triangularization of the surface to express the reflectance map as a linear function of the height. A quadratic error function was minimized by solving a sparse linear system. The multigrid method, with successive linearization, was used to solve this linear system. All height values could be initialized to zero.

Both Zheng and Chellappa's method and Lee and Kuo's method can recover good low frequency information, but high frequency information, the details, are smoothed out. Zheng and Chellappa's results are affected by the background value. Lee and Kuo's method tends to oversmooth the surface and the recovered height is incorrectly slanted upward in one direction.

Rouy and Tourin [19] presented a solution to SFS based on Hamilton–Jacobi–Bellman equations and viscosity solutions theories in order to obtain a unique solution. A link between viscosity solutions and optimal control theories was given via dynamic programming. Moreover, conditions for the existence of both continuous and smooth solutions were provided.

Another approach is by Dupuis and Oliensis [5], [14], [15]. Oliensis [14] observed that the smoothness constraint is only needed at the boundaries if we have initial values at the singular points. Based on this basic idea, Dupuis and Oliensis [5], [15] developed an iterative algorithm to recover depth using discretized optimal control and dynamic programming. The proof of equivalence between the optimal control representation and SFS was illustrated. At first, they did not deal with a general light source and multiple singular points. Later, they removed these restrictions and allowed for a general light source and multiple singular points. However, their initial algorithm [5] requires *a priori* depth information for all singular points. A later extension [15] can determine this information automatically by assuming twice differentiable depth, isolated singular points, and nonzero curvature at singular points.

Bichsel and Pentland [2] simplified Dupuis and Oliensis's approach. They found that a minimum downhill principle could remove the ambiguity introduced by singular points, so that the height information at singular points can be propagated to build a continuous surface. The propagation follows the principle that the height information is only passed to pixels that are farther away from the light source. Among all the pixels that are farther away from the light source, they choose the one that is the closest. The closest point is the one that has maximum height among those that lead away from the light source. This is based on the fact that since both x and y are fixed at each pixel, the distance to the light source is a monotonically increasing function of the surface height if the angle between the light source direction and the optical axis is less than 90°. Eight directions in the image grid are considered, and the image is prerotated to align the projection of the light source direction in the image plane with the x-axis, since the directions for which a solution exists form a very narrow angle with the light source direction, at low brightness points. For each of the eight directions d, if we consider d to be both the p direction (x component of the surface gradient) and the direction of steepest descent, then in the orthogonal direction q (y component of the surface gradient), the reflectance map should

vanish. Consequently, q is obtained by differentiating the reflectance map with respect to q, and solving it for zero. The change of height along direction d is the p value corresponding to this q. This approach used the concept of the derivative of the reflectance map with respect to q, but it does not directly use the intensity gradient information. The problem with this method is that it has difficulty with multiple singular points and is sensitive to noise.

Similar to Horn's and Dupuis and Oliensis's approaches, Kimmel and Bruckstein [3] reconstructed the surface through layers of equal height contours from an initial closed curve. Their method applied techniques in differential geometry, fluid dynamics, and numerical analysis, which enabled the good recovery of nonsmooth surfaces. The algorithm used a closed curve in the areas of singular points for initialization.

None of the above methods deal with interreflections—the mutual illumination between surface facets. Nayar *et al.* [12], [13] addressed the shape-from-interreflection problem using photometric stereo. They observed that the erroneous shape extracted by shape-from-photometric-stereo algorithms in the presence of interreflections was shallower than the real shape. Therefore, they proposed a method to iteratively refine the shape. Similar formulation of interreflection was also discussed by Forsyth and Zisserman [1].

Common problems among the existing SFS algorithms include oversmoothing, lack of robustness, and excessive execution time. To overcome these problems, we introduce a new SFS algorithm, which follows the traditional global approach, but provides more realistic and reliable results with a fast execution time.

In the following, we first present the theory of the proposed method in Section II, then describe its multigrid implementation in Section III. Section IV presents the results. Finally, we provide the conclusion in Section V.

II. Shape Extraction Using the Intensity Gradient

In our approach, we use neither the spherical assumption as in Lee and Rosenfeld's and Pentland's approaches, nor do we base our algorithm on singular points. Unlike Zheng and Chellappa's approach, which considered the intensity derivatives in the x and y directions, the brightness constraint, and the integrability constraint, we drop the traditional brightness constraint and use the intensity gradient constraint. The direction of the intensity gradient is the direction in which the shape of the surface changes the most in the Lambertian model. The directional derivative of the reflectance map, rather than the reflectance map, is linearized using Taylor's series. The discretization of both the surface gradient and its directional derivative, in terms of height, is used in order to express the derivative of the reflectance map as a linear function of height at neighboring pixels. To enforce a unique solution, the smoothness constraint, instead of the integrability constraint, is applied. The resulting nonlinear error function, which includes the smoothness constraint and the simplified intensity gradient constraint, is minimized through the solution of a sparse linear system, which is solved by the multigrid technique.

We use the traditional Lambertian model, based on the assumption of an infinite point light source

$$I_{i,j} = R_{i,j} = \frac{(-p_{i,j}, -q_{i,j}, 1)\cdot \vec{S}}{\sqrt{p_{i,j}^2 + q_{i,j}^2 + 1}} \tag{1}$$

where $I_{i,j}$ is the input intensity at pixel (i, j), which is equal to the reflectance map $R_{i,j}$, $(p_{i,j}, q_{i,j})$ is the surface gradient, and $\vec{S} = (S_x, S_y, S_z)$ is the unit light source direction. We then take the directional derivative of the reflectance map (1) along the intensity gradient direction $d_{i,j}$ as shown in (2), at the bottom of the page, where $p_{d_{i,j}}$ and $q_{d_{i,j}}$ are the directional derivatives of $p_{i,j}$ and $q_{i,j}$ along the intensity gradient direction.

The first order Taylor's expansion around the fixed point $(\overline{p}_{i,j}, \overline{q}_{i,j}, \overline{p}_{d_{i,j}}, \overline{q}_{d_{i,j}})$ yields the following linear approximation to the directional derivative:

$$\begin{aligned} R_{d_{i,j}} \approx\ & R_{d_{i,j}}(\overline{p}_{i,j}, \overline{q}_{i,j}, \overline{p}_{d_{i,j}}, \overline{q}_{d_{i,j}}) \\ & + \frac{\partial R_{d_{i,j}}}{\partial p_{d_{i,j}}}(p_{d_{i,j}} - \overline{p}_{d_{i,j}}) + \frac{\partial R_{d_{i,j}}}{\partial q_{d_{i,j}}}(q_{d_{i,j}} - \overline{q}_{d_{i,j}}) \\ & + \frac{\partial R_{d_{i,j}}}{\partial p_{i,j}}(p_{i,j} - \overline{p}_{i,j}) + \frac{\partial R_{d_{i,j}}}{\partial q_{i,j}}(q_{i,j} - \overline{q}_{i,j}). \end{aligned} \tag{3}$$

By rewriting (3), we obtain

$$\begin{aligned} R_{d_{i,j}} &\approx \alpha_{i,j}p_{i,j} + \beta_{i,j}q_{i,j} + \gamma_{i,j}p_{d_{i,j}} + \phi_{i,j}q_{d_{i,j}} + \eta_{i,j} \\ \alpha_{i,j} &= \frac{\partial R_{d_{i,j}}}{\partial p_{i,j}} \\ \beta_{i,j} &= \frac{\partial R_{d_{i,j}}}{\partial q_{i,j}} \\ \gamma_{i,j} &= \frac{\partial R_{d_{i,j}}}{\partial p_{d_{i,j}}} \\ \phi i,j &= \frac{\partial R_{d_{i,j}}}{\partial q_{d_{i,j}}} \\ \eta_{i,j} &= R_{d_{i,j}}(\overline{p}_{i,j}, \overline{q}_{i,j}, \overline{p}_{d_{i,j}}, \overline{q}_{d_{i,j}}) \\ &\quad - \alpha_{i,j}\overline{p}_{i,j} - \beta_{i,j}\overline{q}_{i,j} - \gamma_{i,j}\overline{p}_{d_{i,j}} - \phi_{i,j}\overline{q}_{d_{i,j}}. \end{aligned} \tag{4}$$

We use the following discrete approximations for $p_{i,j}$, $q_{i,j}$, and their derivatives:

$$\begin{aligned} p_{i,j} &= z_{i,j} - z_{i,j-1} \\ q_{i,j} &= z_{i,j} - z_{i+1,j} \\ p_{x_{i,j}} &= z_{i,j} - 2z_{i,j-1} + z_{i,j-2} \\ p_{y_{i,j}} &= z_{i,j} - z_{i,j-1} - z_{i+1,j} + z_{i+1,j-1} \\ q_{x_{i,j}} &= z_{i,j} - z_{i+1,j} - z_{i,j-1} + z_{i+1,j-1} \\ q_{y_{i,j}} &= z_{i,j} - 2z_{i+1,j} + z_{i+2,j} \\ p_{d_{i,j}} &= p_{x_{i,j}}\Delta x_{i,j} + p_{y_{i,j}}\Delta y_{i,j} \\ q_{d_{i,j}} &= q_{x_{i,j}}\Delta x_{i,j} + q_{y_{i,j}}\Delta y_{i,j} \end{aligned}$$

where $p_{x_{i,j}}$ and $q_{x_{i,j}}$ are derivatives along the x direction, $p_{y_{i,j}}$ and $q_{y_{i,j}}$ are derivatives along the y direction, $\Delta x_{i,j} = \cos\theta$, and $\Delta y_{i,j} = \sin\theta$ [θ is the angle of the intensity gradient at pixel (i, j)]. Then the directional derivative of the reflectance map (4) can be expressed as a linear function of the height at neighboring points

$$\begin{aligned} R_{d_{i,j}} \approx\ & \alpha_{i,j}(z_{i,j} - z_{i,j-1}) + \beta_{i,j}(z_{i,j} - z_{i+1,j}) \\ & + \gamma_{i,j}[(z_{i,j} - 2z_{i,j-1} + z_{i,j-2})\Delta x_{i,j} \\ & + (z_{i,j} - z_{i,j-1} - z_{i+1,j} + z_{i+1,j-1})\Delta y_{i,j}] \\ & + \phi_{i,j}[(z_{i,j} - z_{i+1,j} - z_{i,j-1} + z_{i+1,j-1})\Delta x_{i,j} \\ & + (z_{i,j} - 2z_{i+1,j} + z_{i+2,j})\Delta y_{i,j}] + \eta_{i,j}. \end{aligned} \tag{5}$$

Now compute the height, $z_{i,j}$, by minimizing the following function:

$$\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(I_{d_{i,j}} - R_{d_{i,j}})^2 + \lambda\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}(p_{x_{i,j}}^2 + p_{y_{i,j}}^2 + q_{x_{i,j}}^2 + q_{y_{i,j}}^2) \tag{6}$$

$$R_{d_{i,j}} = \frac{(-S_x p_{d_{i,j}} - S_y q_{d_{i,j}})(p_{i,j}^2 + q_{i,j}^2 + 1) - (S_z - S_x p_{i,j} - S_y q_{i,j})(p_{i,j}p_{d_{i,j}} + q_{i,j}q_{d_{i,j}})}{(p_{i,j}^2 + q_{i,j}^2 + 1)^{3/2}} \tag{2}$$

where n is the image size and λ is the weight of the smoothness term. The first term is the intensity gradient constraint and the second term is the smoothness constraint. Substituting $R_{d_{i,j}}$ from (5) for the first term of (6), we obtain

$$\begin{aligned}
&\sum_{i=0}^{n-1}\sum_{j=0}^{n-1}\{a_{i,j}^2 z_{i,j}^2 + b_{i,j}^2 z_{i,j-1}^2 + c_{i,j}^2 z_{i+1,j}^2 \\
&\quad + d_{i,j}^2 z_{i,j-2}^2 + e_{i,j}^2 z_{i+1,j-1}^2 + f_{i,j}^2 z_{i+2,j}^2 \\
&\quad + 2[a_{i,j}b_{i,j}z_{i,j}z_{i,j-1} + a_{i,j}c_{i,j}z_{i,j}z_{i+1,j} \\
&\quad + a_{i,j}d_{i,j}z_{i,j}z_{i,j-2} + a_{i,j}e_{i,j}z_{i,j}z_{i+1,j-1} \\
&\quad + a_{i,j}f_{i,j}z_{i,j}z_{i+2,j} + b_{i,j}c_{i,j}z_{i,j-1}z_{i+1,j} \\
&\quad + b_{i,j}d_{i,j}z_{i,j-1}z_{i,j-2} + b_{i,j}e_{i,j}z_{i,j-1}z_{i+1,j-1} \\
&\quad + b_{i,j}f_{i,j}z_{i,j-1}z_{i+2,j} + c_{i,j}d_{i,j}z_{i+1,j}z_{i,j-2} \\
&\quad + c_{i,j}e_{i,j}z_{i+1,j}z_{i+1,j-1} + c_{i,j}f_{i,j}z_{i+1,j}z_{i+2,j} \\
&\quad + d_{i,j}e_{i,j}z_{i,j-2}z_{i+1,j-1} + d_{i,j}f_{i,j}z_{i,j-2}z_{i+2,j} \\
&\quad + e_{i,j}f_{i,j}z_{i+1,j-1}z_{i+2,j} + a_{i,j}g_{i,j}z_{i,j} \\
&\quad + b_{i,j}g_{i,j}z_{i,j-1} + c_{i,j}g_{i,j}z_{i+1,j} + d_{i,j}g_{i,j}z_{i,j-2} \\
&\quad + e_{i,j}g_{i,j}z_{i+1,j-1} + f_{i,j}g_{i,j}z_{i+2,j}] + g_{i,j}^2\} \qquad (7)
\end{aligned}$$

where

$$\begin{aligned}
a_{i,j} &= -\alpha_{i,j} - \beta_{i,j} - \gamma_{i,j}(\Delta x_{i,j} + \Delta y_{i,j}) \\
&\quad - \phi_{i,j}(\Delta x_{i,j} + \Delta y_{i,j}) \\
b_{i,j} &= \alpha_{i,j} + \gamma_{i,j}(2\Delta x_{i,j} + \Delta y_{i,j}) + \phi_{i,j}\Delta x_{i,j} \\
c_{i,j} &= \beta_{i,j} + \gamma_{i,j}\Delta y_{i,j} + \phi_{i,j}(\Delta x_{i,j} + 2\Delta y_{i,j}) \\
d_{i,j} &= -\gamma_{i,j}\Delta x_{i,j} \\
e_{i,j} &= -\gamma_{i,j}\Delta y_{i,j} - \phi_{i,j}\Delta x_{i,j} \\
f_{i,j} &= -\phi_{i,j}\Delta y_{i,j} \\
g_{i,j} &= I_{d_{i,j}} + \eta_{i,j}.
\end{aligned}$$

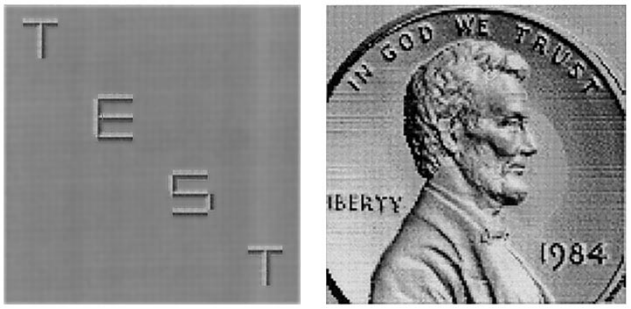
The second term of (6), the smoothness constraint, can be represented by a template, to be applied to the 2-D height, as follows [11], [21]:

$$V: \frac{1}{h^2}\begin{vmatrix} & & 1 & & \\ & 2 & -8 & 2 & \\ 1 & -8 & 20 & -8 & 1 \\ & 2 & -8 & 2 & \\ & & 1 & & \end{vmatrix}.$$

Here, h is the spacing between pixels. The templates for the image boundary can be found in [11] and [21].

Equation (7) can be rewritten in the following matrix form:

$$\tfrac{1}{2}\, z^T U z - \omega^T z + \mu.$$



(a) (b)

Fig. 1. Synthetic input images. (a) *Letters* with light source direction $(-1,\ 1,\ 1)$ and (b) *Penny* with light source direction $(5,\ 5,\ 7)$.

Adding the smoothness term (8) to this quadratic equation, we have

$$\tfrac{1}{2}\, z^T T z - \omega^T z + \mu \qquad (9)$$

where $T = U + \lambda V$. U is an $n^2 \times n^2$ (n is the image size) symmetric, sparse matrix consisting of the second-order terms from (7). U can be formed by using a $(2n-3)\times(2n-3)$ template $\Delta U_{i,j}$ given in (10), shown at the bottom of the page, whose nonzero elements are the properly positioned coefficients from (7). The diagonal elements of $\Delta U_{i,j}$ are the coefficients of the squared terms. The symmetric elements are obtained from the coefficients of the remaining second order terms. Initially, U is zero; then for each $(i,\ j)$, $\Delta U_{i,j}$ is added to the proper position in U where the upper-left corner of $\Delta U_{i,j}$ corresponds to position $(x-2,\ x-2)$ in U, and $x = i*n+j$. This procedure will form a sparse banded matrix which can be represented by an $n^2 \times 19$ matrix.

As an example, given $(i,\ j)$, the term $2a_{i,j}b_{i,j}z_{i,j}z_{i,j-1}$ in (7) will add the value $2a_{i,j}b_{i,j}$ to U at the symmetric positions $(x_1,\ x_2)$ and $(x_2,\ x_1)$ (where $x_1 = i*n+j$ and $x_2 = i*n+j-1$). This corresponds to assigning $2a_{i,j}b_{i,j}$ to $\Delta U_{i,j}$ at the symmetric positions $(x_3,\ x_4)$ and $(x_4,\ x_3)$ [where $x_3 = x_1 - (x-2)$, and $x_4 = x_2 - (x-2)$]. [Note: $(x-2,\ x-2)$ is the position of upper-left corner of $\Delta U_{i,j}$ in U.]

The vector ω is an $n^2 \times 1$ vector consisting of the first order terms from (7). Initially, ω is the zero vector, then for each $(i,\ j)$, $0 \le i,\ j < n$, we locate the corresponding position in ω as $x = i*n+j$, then subtract $d_{i,j}g_{i,j}$, $b_{i,j}g_{i,j}$, $a_{i,j}g_{i,j}$, $e_{i,j}g_{i,j}$, $c_{i,j}g_{i,j}$, $f_{i,j}g_{i,j}$, from positions $x-2$, $x-1$, x, $x+n-1$, $x+n$, $x+2n$, respectively, in ω. Finally, μ is a scalar consisting of the sum of the constant

$$\Delta U_{i,j} = 2\begin{vmatrix}
\frac{1}{2}d_{i,j}^2 & b_{i,j}d_{i,j} & a_{i,j}d_{i,j} & \overbrace{0\cdots0}^{n-2} & d_{i,j}e_{i,j} & c_{i,j}d_{i,j} & \overbrace{0\cdots0}^{n-1} & d_{i,j}f_{i,j} \\
b_{i,j}d_{i,j} & \frac{1}{2}b_{i,j}^2 & a_{i,j}b_{i,j} & \overbrace{0\cdots0}^{n-2} & b_{i,j}e_{i,j} & b_{i,j}c_{i,j} & \overbrace{0\cdots0}^{n-1} & b_{i,j}f_{i,j} \\
a_{i,j}d_{i,j} & a_{i,j}b_{i,j} & \frac{1}{2}a_{i,j}^2 & \overbrace{0\cdots0}^{n-2} & a_{i,j}e_{i,j} & a_{i,j}c_{i,j} & \overbrace{0\cdots0}^{n-1} & a_{i,j}f_{i,j} \\
n-2 & \begin{cases}0\\ \vdots\\ 0\end{cases} & & n-2 & \begin{cases}0\\ \vdots\\ 0\end{cases} & & n-2 & \begin{cases}0\\ \vdots\\ 0\end{cases} \\
d_{i,j}e_{i,j} & b_{i,j}e_{i,j} & a_{i,j}e_{i,j} & \overbrace{0\cdots0}^{n-2} & \frac{1}{2}e_{i,j}^2 & c_{i,j}e_{i,j} & \overbrace{0\cdots0}^{n-1} & e_{i,j}f_{i,j} \\
c_{i,j}d_{i,j} & b_{i,j}c_{i,j} & a_{i,j}c_{i,j} & 0\cdots0 & c_{i,j}e_{i,j} & \frac{1}{2}c_{i,j}^2 & 0\cdots0 & c_{i,j}f_{i,j} \\
n-1 & \begin{cases}0\\ \vdots\\ 0\end{cases} & & n-1 & \begin{cases}0\\ \vdots\\ 0\end{cases} & & n-1 & \begin{cases}0\\ \vdots\\ 0\end{cases} \\
d_{i,j}f_{i,j} & b_{i,j}f_{i,j} & a_{i,j}f_{i,j} & \overbrace{0\cdots0}^{n-2} & e_{i,j}f_{i,j} & c_{i,j}f_{i,j} & \overbrace{0\cdots0}^{n-1} & \frac{1}{2}f_{i,j}^2
\end{vmatrix} \qquad (10)$$

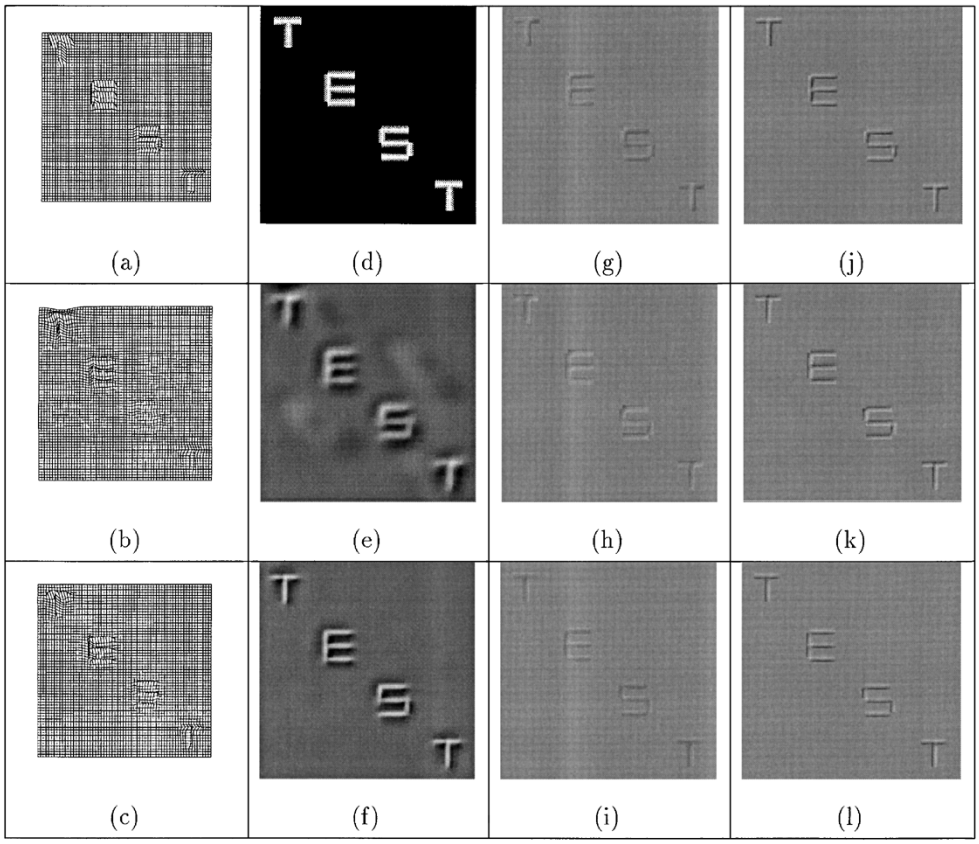


Fig. 2. Results for the *Letters* image. (a) Three-dimensional plot of the true depth, (b) 3-D plot of the recovered depth from Lee and Kuo's algorithm, (c) 3-D plot of the recovered depth from our algorithm, (d) range image of the true depth, (e) range image of Lee and Kuo's recovered depth, (f) range image of our recovered depth, (g) shaded image for (b) using light source $(-1, 1, 1)$, (h) shaded image for (b) using light source $(1, -1, 1)$, (i) shaded image for (b) using light source $(1, 1, 1)$, (j) shaded image for (c) using light source $(-1, 1, 1)$, (k) shaded image for (c) using light source $(1, -1, 1)$, and (l) shaded image for (c) using light source $(1, 1, 1)$.

terms of (7)

$$\mu = \sum_{i=0}^{n-1} \sum_{j=0}^{n-1} g_{i,j}^2.$$

The minimization of (9) is done by solving the linear system $Tz = \omega$.

III. Multigrid Technique

The multigrid method [18] was first introduced by Brandt in the early 1970's for solving elliptic partial differential equations (PDE's). It is a fast technique to solve either linear or nonlinear elliptic problems; we are looking for a solution to a linear problem.

The basic idea behind the multigrid method is to combine a traditional relaxation method with coarse-grid correction, so that the error generated in the finer grid can be corrected in the coarser grid to yield a more efficient, and accurate, solution. The number of grid levels, L, in one iteration of the multigrid method is determined by the size of the image, n, to be $L = \log_2 n - 1$. The multigrid method can be performed iteratively by using the solution from the previous iteration as the initial value for the next. One iteration of the multigrid method, from the finest grid to the coarsest and back to the finest, is called a cycle. There are different structures for the cycle [18]. We use the V-cycle structure since it is simple to implement, yet provides reasonably good results.

The process of one cycle for solving the linear system $Tz = \omega$, can be described as follows.

1) *Initialization:* Start at the finest grid h (h indicates the spacing between pixels).
2) *Presmoothing:* Apply a relaxation method to compute an approximate solution, $\tilde{z}_h$, to $T_h z_h = \omega_h$, on grid h, with z_h initially zero.
3) *Coarse-Grid Correction:*
   a) Compute the residual on grid h: $r_h = T_h \tilde{z}_h - \omega_h$.
   b) Reduce grid h into H.
   c) Restrict r_h (on the fine grid h) into r_H (on the coarse grid H): $r_H = \mathcal{R}(r_h)$.
   d) If H is the coarsest grid, use an exact solver to obtain a solution $\tilde{z}_H$ for $T_H z_H = r_H$ on grid H. Go to (f).
   e) Obtain a solution $\tilde{z}_H$ for $T_H z_H = r_H$ on grid H. Recursively repeat (2) through (4) for coarser grids.
   f) Upon returning from the recursion, prolongate the correction to a finer grid: $\Delta \tilde{z}_h = \mathcal{P}(\tilde{z}_H)$.
   g) Correct the solution on grid h: $\tilde{z}_h = \tilde{z}_h + \Delta \tilde{z}_h$.
4) *Post-Smoothing:* Take $\tilde{z}_h$ as initial value, apply a relaxation method to refine the approximate solution, $\tilde{z}_h$, to $T_h z_h = \omega_h$ on grid h.

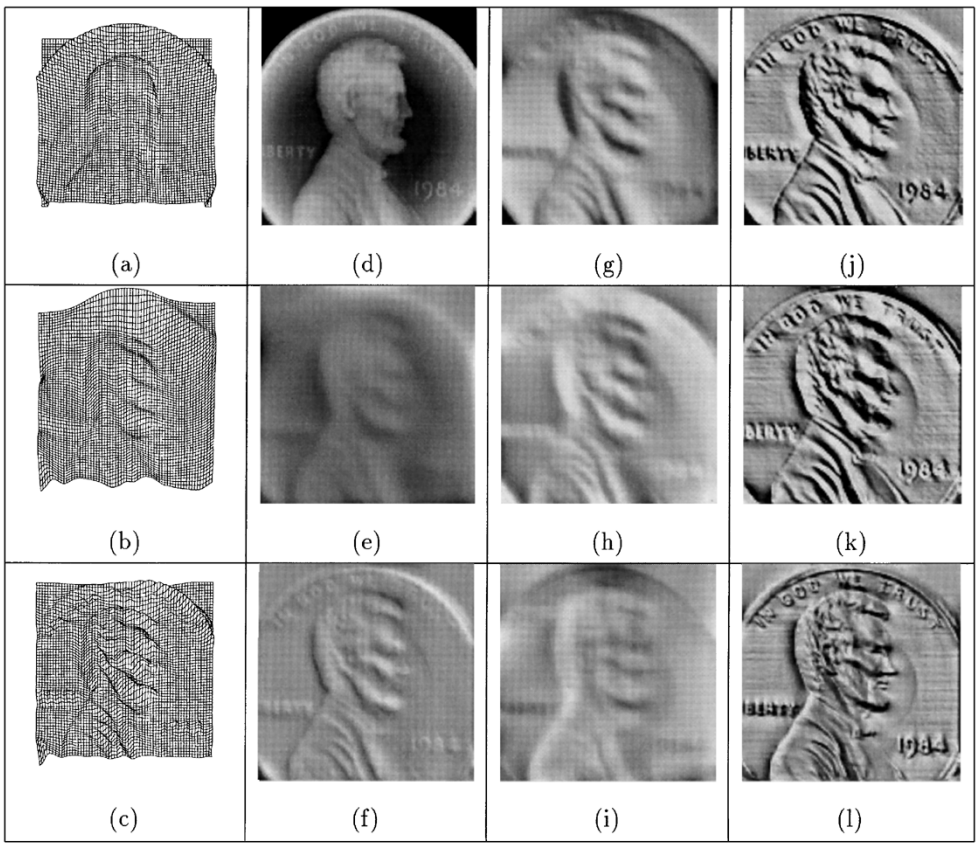


Fig. 3. Results for the *Penny* image. (a) Three-dimensional plot of the true depth, (b) 3-D plot of the recovered depth from Lee and Kuo's algorithm, (c) 3-D plot of the recovered depth from our algorithm, (d) range image of the true depth, (e) range image of Lee and Kuo's recovered depth, (f) range image of our recovered depth, (g) shaded image for (b) using light source $(5, 5, 7)$, (h) shaded image for (b) using light source $(-5, -5, 7)$, (i) shaded image for (b) using light source $(-5, 5, 7)$, (j) shaded image for (c) using light source $(5, 5, 7)$, (k) shaded image for (c) using light source $(-5, -5, 7)$, and (l) shaded image for (c) using light source $(-5, 5, 7)$.

In our multigrid implementation, Gauss–Seidel was used for both the relaxation method and exact solver. Full-weighting restriction

$$\mathcal{R}: \begin{bmatrix} \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \\ \frac{1}{8} & \frac{1}{4} & \frac{1}{8} \\ \frac{1}{16} & \frac{1}{8} & \frac{1}{16} \end{bmatrix} \tag{11}$$

was applied to transfer the residual from finer grids to coarser grids, and bilinear prolongation

$$\mathcal{P}: \begin{bmatrix} \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \\ \frac{1}{2} & 1 & \frac{1}{2} \\ \frac{1}{4} & \frac{1}{2} & \frac{1}{4} \end{bmatrix} \tag{12}$$

was applied to make the correction from coarser grids to finer grids. At each level, the size of the grid is reduced by half.

IV. Results

Among existing SFS techniques, Lee and Kuo's approach is one of the newest and provides very good results. They also applied the multigrid technique. Therefore, we implemented their algorithm and compared the results with ours.

The results for our algorithm are given after one multigrid cycle; since the results after one cycle are already accurate enough, any extra cycles will not yield significant improvement. However, the property of coarse-to-fine-correction in the multigrid technique makes even one cycle meaningful. The smoothing factor λ was chosen as 2000. The maximum number of iterations for Gauss–Seidel is 500. The initial heights were chosen as zero for all tests. In order to show the performance, the top view of the 3-D plots of the recovered heights are presented. In addition, we also compute the mean surface orientation error for results on synthetic images and compare our algorithm with Lee and Kuo's.

TABLE I
Mean p–q Error for Synthetic Images

	Images	
Methods	**Letters**	**Penny**
Lee & Kuo	0.266687	1.135478
Proposed method	0.22	0.47092

We first show the results for the proposed algorithm on two synthetic images: *Letters* and *Penny* (Fig. 1). *Letters* is generated from the synthetic depth of a text string "TEST" with light source direction $(-1, 1, 1)$, and *Penny* is generated from the range data of a penny with light source direction $(5, 5, 7)$. Both images are 128 by 128.

Fig. 2 shows results for the *Letters* image It contains 3-D plots of the true depth, the reconstructed depth from our algorithm and Lee and Kuo's (all of them have the same view); range images for the true

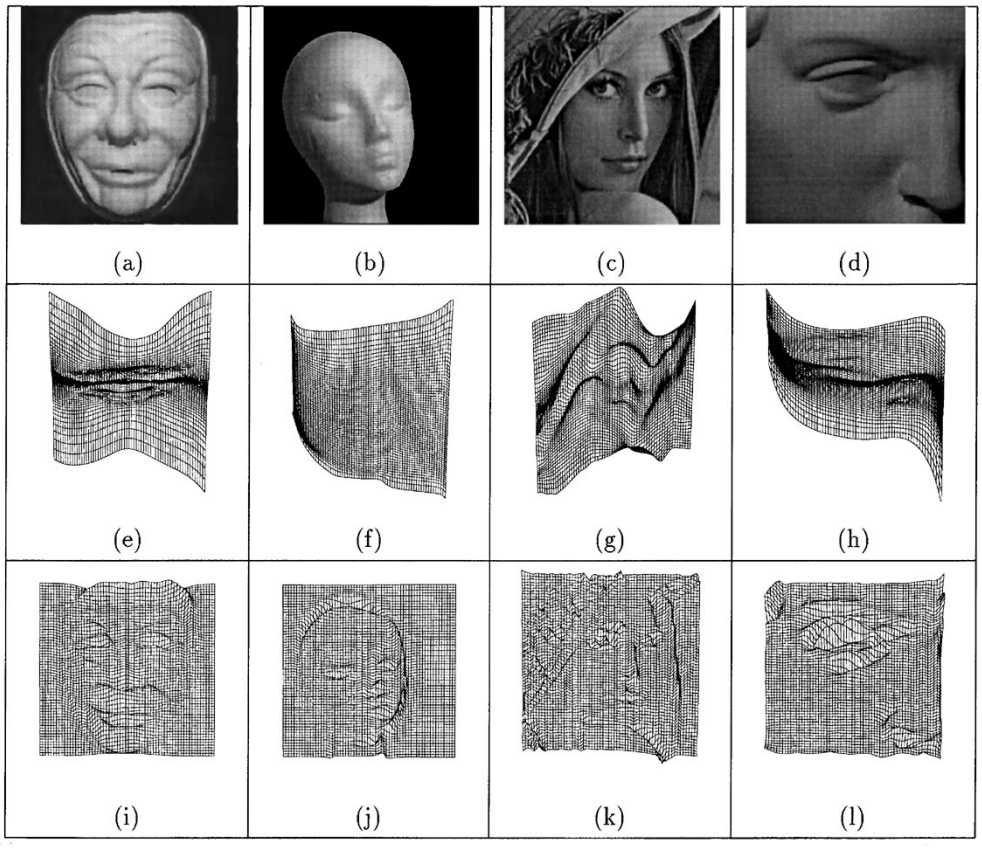


Fig. 4. Results for real images. (a) *Mask* [the estimated light source direction is $(-0.479\,847,\ -0.907\,563,\ 1)$], (b) *Mannequin* [the estimated light source direction is $(-0.345,\ 0.345,\ 0.875)$], (c) *Lenna* [the estimated light source direction is $(1.5,\ 0.866,\ 1)$], (d) *David* [the estimated light source direction is $(-0.707,\ 0.707,\ 1)$], (e)–(l) 3-D plots of the recovered height from Lee and Kuo's algorithm [(e)–(h)], and from our algorithm [(i)–(l)].

TABLE II
CPU TIME (IN SECONDS)

	Images					
Methods	**Letters**	**Penny**	**Mask**	**David**	**Lenna**	**Mannequin**
Lee & Kuo	249.01	2678.29	2543.87	1077.22	14546.79	2366.25
Proposed method	387.14	374.37	397.58	814.39	1215.19	1424.67

depth and the reconstructed depths from both algorithms; and shaded images of the true depth and both reconstructed depths using the original, the opposite, and the orthogonal light sources respectively. Similarly, Fig. 3 shows the results for *Penny*.

From the results for *Letters*, we can see that our algorithm provides a more accurate recovery. This can clearly be seen from the range images, since each range image is scaled from the depth data into the range [0, 255] to provide more contrast. The recovery of the bottom two letters from Lee and Kuo's algorithm can hardly be seen in their 3-D plot due to oversmoothing, while our algorithm recovers all of the letters reasonably well. The shaded images, generated from our reconstructed depth, are also better.

For *Penny*, Lee and Kuo's method loses a lot of detail, which causes the shaded output image to appear blurry. Moreover, the 3-D plot shows that their algorithm recovers a twisted background of the penny. On the other hand, our algorithm does not seem to exhibit any of these problems.

To further analyze the results of the two algorithms, we compare them in terms of the mean surface gradient error (Table I). The reconstructed depth is first rescaled according to the true depth, then a discrete approximation is used to estimate the surface gradient. Table I indicates that our results have approximately 18 to 59% less error than Lee and Kuo's. *Letters* image is quite simple as expected error is much less than the *Penny* image. Also note that the light source directions are different in both cases. The depth error is not computed here since a single peak or valley point (outlier) in the recovered depth may cause big error due to the shift in the renormalization.

The remaining results are given for four real images with the light source directions estimated by Lee and Rosenfeld's method [10]: *Mask* [the estimated light source direction is $(-0.479\,847,\ -0.907\,563,\ 1)$], *Mannequin* [the estimated light source direction is $(-0.345,\ 0.345,\ 0.875)$], *Lenna* [the estimated light source direction is $(1.5,\ 0.866,\ 1)$], and *David* [the estimated light source direction is $(-0.707,\ 0.707,\ 1)$]. The sizes of *Mannequin* and *Lenna* are both 256 by 256, and the sizes of *David* and *Mask* are 128 by 128. The results for Lee and Kuo's algorithm, and for our own algorithm, are shown in Fig. 4. Since a small error in the

depth can cause a big error in the surface orientation, especially in the case of noisy real images, the shaded versions of the recovered depth for the real images are not included here.

For *Mask*, *Mannequin*, and *David*, the recovered heights from Lee and Kuo's algorithm are very flat, even for small λ values. In contrast, our algorithm gives very good, detailed height information for *Mask* and *David*. It also provides much more detail for *Mannequin*, despite the noise at the object boundary. Lee and Kuo's algorithms recovered very good height information for the real image *Lenna*, but details are missing. Our result for *Lenna* shows accurate details. The rough height recovered in the area of *Lenna*'s hair is due to the change in albedo, which violates the constant albedo assumption.

Although both Lee and Kuo's and our algorithms employ the multigrid technique, our method is significantly faster than Lee and Kuo's in all cases except for *Letters*, no matter what threshold is used for Gauss–Seidel. This can be seen in Table II. The speedup ranges from 6.4 for the *Mask* image to 1.6 for the *Mannequin* image. The analysis was done on a Sun SPARC 4.

V. Conclusion

We presented a new SFS algorithm, which replaced the traditional brightness constraint with an intensity gradient constraint based on the fact that the direction of the intensity gradient is the direction in which the shape changes the most. The results have shown that our algorithm has robust performance for different images, and that it is more efficient than the existing multigrid SFS technique.

Acknowledgment

The authors would like to thank Prof. Kuo and Dr. Lee of USC for their helpful discussions and for providing *Penny*, *Lenna*, and *David* images. The authors would also like to thank Dr. Leclerc of Artificial Intelligence Center, SRI International, for providing the *Mannequin* image.

References

[1] D. Forsyth and A. Zisserman, "Mutual illumination," in *Proc. IEEE Computer Vision Pattern Recognition*, 1989, pp. 466–473.
[2] M. Bichsel and A. P. Pentland, "A simple algorithm for shape from shading," in *Proc. IEEE Computer Vision Pattern Recognition*, 1992, pp. 459–465.
[3] R. Kimmel and A. M. Bruckstein, "Shape from shading via level sets," Israel Institute of Technology, CIS Rep. 9209, 1992.
[4] M. J. Brook and B. K. P. Horn, "Shape and source from shading," in *Proc. Int. Joint Conf. Artificial Intelligence*, 1985, pp. 932–936.
[5] P. Dupuis and J. Oliensis, "Direct method for reconstructing shape from shading," in *Proc. IEEE Computer Vision Pattern Recognition*, 1992, pp. 453–458.
[6] B. K. P. Horn, "Shape from shading: A method for obtaining the shape of a smooth opaque object from one view," Ph.D. dissertation, Mass. Inst. Technol., Cambridge, 1970.
[7] B. K. P. Horn, "Height and gradient from shading," *Int. J. Comput. Vision*, pp. 37–75, 1989.
[8] K. Ikeuchi and B. K. P. Horn, "Numerical shape from shading and occluding boundaries," *Artif. Intell.*, vol. 17, no. 1–3, pp. 141–184, 1981.
[9] Y. G. Leclerc and A. F. Bobick, "The direct computation of height from shading," in *Proc. IEEE Computer Vision Pattern Recognition*, 1991, pp. 552–558.
[10] C. H. Lee and A. Rosenfeld, "Improved methods of estimating shape from shading using the light source coordinate system," *Artif. Intell.*, vol. 26, pp. 125–143, 1985.
[11] K. M. Lee and C. C. J. Kuo, "Shape from shading with a linear triangular element surface model," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, pp. 815–822, Aug. 1993.
[12] S. K. Nayar, K. Ikeuchi, and T. Kanade, "Shape from interreflections," in *Proc. Int. Conf. Computer Vision*, 1990, pp. 1–11.
[13] ____, "Surface reflection: Physical and geometrical perspectives," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 611–663, July 1991.
[14] J. Oliensis, "Shape from shading as a partially well-constrained problem," *Comput. Vision, Graph., Image Process.*, vol. 54, pp. 163–183, 1991.
[15] J. Oliensis and P. Dupuis, "A global algorithm for shape from shading," in *Proc. Int. Conf. Comput. Vision*, 1993, pp. 692–701.
[16] A. Pentland, "Shape information from shading: A theory about human perception," in *Proc. Int. Conf. Computer Vision*, 1988, pp. 404–413.
[17] A. P. Pentland, "Local shading analysis," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. PAMI-6, pp. 170–187, 1984.
[18] W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes in C.* Cambridge, U.K.: Cambridge Univ. Press, 1990.
[19] E. Rouy and A. Tourin, "A viscosity solutions approach to shape from shading," *SIAM J. Numer. Anal.*, vol. 29, no. 3, pp. 867–884, 1992.
[20] R. Szeliski, "Fast shape from shading," *Comput. Vision, Graph., Image Process.*, vol. 53, pp. 129–153, 1991.
[21] D. Terzopoulos, "Multilevel computational processes for visual surface reconstruction," *Comput. Vision, Graphics, Image Process.*, vol. 24, pp. 52–96, 1988.
[22] P. S. Tsai and M. Shah, "Shape from shading using linear approximation," *Image Vision Comput. J.*, vol. 12, no. 8, pp. 487–498, 1994.
[23] Q. Zheng and R. Chellappa, "Estimation of illuminate direction, albedo, and shape from shading," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 680–702, July 1991.

Fast Implementation of Forward Robot Kinematics of Position with Distributed Arithmetic Architecture

B. G. Mertzios and G. K. Grigoriadis

***Abstract*—This paper refers to the fast implementation of the forward kinematic equations of position of robotics manipulators, using a distributed arithmetic-based pipeline architecture. The building blocks of this pipeline architecture are the distributed arithmetic-based circuits that implement the matrix-vector multiplications involved in the calculation of the forward kinematics of position. The matrix-vector multiplications are implemented in the distributed arithmetic technique by using auxiliary binary functions, which are stored in look-up tables. The digit-serial configuration of the proposed implementation is described. The serial and the parallel configurations may result as special extreme cases of the digit-serial configuration.**

***Index Terms*— Digit-serial configuration, distributed arithmetic, forward kinematics of position, matrix-matrix-multiplication (MMM), matrix-vector-multiplication (MVM), pipelining.**

I. Introduction

Robotics manipulators are articulated chains of rigid bodies (links), which are connected serially by joints. One end of the chain is attached to a supporting base where a inertial frame (the base) is established, while the other end (the end-effector) is free, in order to accomplish the manipulation tasks. Forward (direct) kinematics deals with the problem of determining the position of the end-effector from a given set of joint coordinates. Inverse kinematics addresses the

Manuscript received June 26, 1994; revised April 14, 1996.

The authors are with the Department of Electrical and Computer Engineering, Automatic Control Systems Laboratory, Democritus University of Thrace, GR-67 100 Xanthi, Greece (e-mail: mertzios@demokritos.cc.duth.gr).

Publisher Item Identifier S 1083-4427(99)01456-3.