Textured 3D Reconstruction Using Photometric Stereo for Projection-based Display Systems

Isao Miyagawa (member)†, Hiroyuki Arai (member)†, Yukinobu Taniguchi (member)†

Abstract We propose a new textured 3D reconstruction method that uses image sequences output by projection-based display systems. So that image sequences can recreate convex objects on a planar display, we extend the photometric model used in photometric stereo to handle full color images. According to the photometric model, we generate multiple shading images from surface normals on the original 3D object with varying light directions. We show that an image data matrix created from the multiple images can be decomposed into a surface normal matrix scaled up to the color intensity of three channels and a light-source directional matrix, and that its rank is at most three. In order to ensure robust and unique matrix decomposition, we introduce new metric constraints by using pairs of the directional lights that correspond to odd and even frames. Based on matrix decomposition, we describe a method that reconstructs the surface normals for 3D shape and surface texture. We examine the proposed method by challenging it with both computer-simulated data and real images. These results demonstrate that our method yields good 3D reconstruction and surface texture.

Key words: projection-based display system, photometric stereo, radiometric compensation, 3D reconstruction

1. Introduction

Smartphones and tablet devices are now popular with the general public and hold the position of an essential tool in life. To allow such mobile information devices to furnish attractive entertaining services, our aim is a display system that, in association with the user’s camera, provides an interactive service that references locations in the real world. For instance, the display system will enhance viewers’ appreciation of 3D works of art, pictures, and provide virtual relief at galleries and museums. Your friends can enjoy 3D versions of event site and theme park characters you captured as 2D images, and shoppers can peruse 3D products when window shopping. Utilizing “the object you see today at this place” is an important concept in these applications. Once the display system has the 3D object, it will arrange the appearance of the objects with surface texture day after day or freely alter the ambient lighting under which the 3D object is displayed. Moreover, AR techniques allow the user to fuse the 3D object with actual objects or other 3D objects acquired from other users or central repositories. Even if object combinations do not exist in the database, the system will be able to create attractive 3D object combinations to satisfy the user’s preference. If the display system virtually exhibits the objects in conjunction with a scene, it realizes the goal of 3D models that are inexpensive to make.

Provided that the user’s camera captures the multiple images output from the display systems, we offer 3D reconstruction in the framework of Structure from X. We use a display system that outputs the image sequence created from a 3D shape and its surface texture. In order to realize 3D reconstruction without requiring pixel correspondence in the image sequence or installing special equipment or functions in the camera, we employ image shading computed from surface normals on the modeled 3D object with varying light directions based on the principle of photometric stereo. Photometric stereo is well-known to yield 3D reconstruction by its use of Structure from Shading; it estimates 3D shape from multiple images captured under different lighting conditions. The most popular methods assume an orthographic model with Lambertian reflectance and decompose an image data matrix by using singular value decomposition in the same factorization manner; they simultaneously recover lighting direction vectors and surface normals on 3D shape. By employing the basic algorithms, we design a display system that reconstructs virtually the 3D object with color texture from its image sequence. In addition, we focus on projection-based display systems because projectors are extremely flexible output devices that can display images...
anywhere. Projection-based displays can also realize projection-mapped shows by using huge public viewing spaces or building sides. If the projector is connected to a camera, the projector and the camera can be structured as a closed-loop system, i.e. it feeds back the RGB value observed by the camera to the projector. By using feedback and a compensation technique\(^\text{12}-\text{14}\), the color brightness output from the projector can be adjusted to obtain the desired image sequence.

In this paper, we propose a textured 3D reconstruction method that obtains the 3D object and its surface texture from an image sequence; this method is a key player in realizing the display system desired. Existing photometric stereo methods\(^\text{3}\) cannot easily handle 3D objects with color texture under ambient and environment illumination, because they basically extract 3D geometric surfaces by varying directional light under dark room conditions. Our solution extends the photometric model used in photometric stereo to handle full color images for projection-based display systems. According to the extended photometric model, we generate multiple shading images from surface normals on the original 3D object with varying light directions, and capture the desired image sequence by employing a compensation technique even if unknown ambient light is present. We show that an image data matrix created from the multiple images can be decomposed into a surface normal matrix scaled up to the color intensity of three channels and a light-source directional matrix, and that its rank is at most three. Generally, 3D reconstruction using photometric stereo demands strong metric constraints to robustly decompose a light-source directional matrix and a surface normal matrix from the image data matrix given the presence of image noise. In order to ensure robust and unique matrix decomposition, we introduce new metric constraints by using pairs of the directional lights that correspond to odd and even frames. Based on matrix decomposition, we obtain surface normals on 3D shape and the texture from the image data matrix.

This paper is organized as follows: Section 2 describes the projection-based display system based on the orthographic projection model. Section 3 describes our proposed method; it simultaneously reconstructs a convex 3D object and its surface texture from multiple images. Section 4 details the performance of our method on both computer-simulated data and real images and discusses the results of our experiments. Section 5 concludes this study and mentions future work.

### 2. Projection-based Display System

**Fig. 1(a)** illustrates the relationship between a surface normal and a directional vector of a distant light source. We assume that a convex object with Lambertian surface is illuminated by distant light sources, each of which has a different orientation, and that a camera observes the irradiance emanating from the object’s surface. We denote pixel coordinates on the image by suffix \(j \in \{1,2,\ldots,P\}\), and the surface normal, which corresponds to the \(j\)-th pixel, on the object by vector \(\mathbf{n}_j\). When we denote the surface geometry by

\[
Z_j = f(X_j, Y_j),
\]  

the surface normal \(\mathbf{n}_j\) at the \(j\)-th point is given by

\[
\mathbf{n}_j = \alpha_j \begin{bmatrix} -\partial f/\partial X_j, & -\partial f/\partial Y_j, & 1 \end{bmatrix}^T,
\]  

\[
\alpha_j = \frac{1}{\sqrt{\left(\partial f/\partial X_j\right)^2 + \left(\partial f/\partial Y_j\right)^2 + 1}}.
\]  

according to Eq.\(^{15}\); see Appendix A. Consider the directional vector \(\mathbf{l}_i\) of a distant light source, where we identify different lights by suffix \(i \in \{1,2,\ldots,M\}\). According to the orthographic projection adopted in photometric stereo\(^{4,16,17}\), we describe the observed intensity as

\[
C_{\mu ij} = \left( \int S(\lambda)\rho_j(\lambda)\mu(\lambda)\,d\lambda \right) \mathbf{l}_i^T \mathbf{n}_j = \mu_j \mathbf{l}_i^T \mathbf{n}_j,
\]  

where \(\rho_j(\lambda)\) is the reflectance of the object’s surface for wavelength \(\lambda\) at the \(j\)-th point. \(S(\lambda)\) is a spectral of incidence light, and \(\mu(\lambda)\) is the \(\mu\)-channel spectral sensitivity of the camera. Since the integral term is independent of both surface normal and directional vector of the light source in Eq. (4), we replace it with \(\mu_j\), which is called the albedo. The existing photometric stereo methods based on Eq. (4) yield surface normals, light direction vectors, and a constant albedo \(\mu_j = \mu\) from multiple images.
Substituting the projection-based display system shown in Fig. 1(b), we aim to acquire the same multiple images that are produced by the photometric model under orthographic projection. Only the projector displays the desired images, which may be observed by the virtual orthographic camera shown in Fig. 1(a), as if the object on the screen is illuminated from each direction \( \mathbf{l}_i \) by different distant light sources. In order to represent a 3D object with texture on the planar screen, we use \( \mu_j, \mu \in \{r, g, b\} \) in Eq. (4) as each pixel value on the texture image. Namely, the shading image computed from the inner product between surface normal \( \mathbf{n}_j \) and directional vector \( \mathbf{l}_i \) is superimposed on the object’s texture image. Note that surface normal \( \mathbf{n}_j \) does not mean the surface normal on the actual screen, but means the surface normal on a convex “virtual object”, which does not exist on the screen.

We adjust the three color lights that illuminate the planar screen by a projection-based display system, so that the camera observes the desired color brightness \((C_{r,j}, C_{g,j}, C_{b,j})\). We support the case wherein a pixel on the image captured by the camera is matched to both one of the pixels output from the projector and one of the points illuminated on the screen, by plane homographies among the projector, camera, and screen. In this paper, we assume that the display surface is Lambertian and that the measured pixel value for a color channel is linear in terms of irradiance. According to (12)~(14), given \( P_{k,j} \) as projector’s brightness output from the \( k \)-channel, \( k \in \{r, g, b\} \), we have

\[
\begin{bmatrix}
C_{r,j} \\
C_{g,j} \\
C_{b,j}
\end{bmatrix} =
\begin{bmatrix}
V_{r,j} \\
V_{g,j} \\
V_{b,j}
\end{bmatrix}
\begin{bmatrix}
P_{r,j} \\
P_{g,j} \\
P_{b,j}
\end{bmatrix}
+ \begin{bmatrix}
F_{r,j} \\
F_{g,j} \\
F_{b,j}
\end{bmatrix},
\]

We denote an element of the color mixing matrix, which directly connects projector brightness and camera brightness by \( V_{\mu,k,j} \), and each component of the ambient light by \( F_{\mu,j} \), which includes black offset from the projector. We compensate color brightness \( P_{k,j} \) by the following equation:

\[
\begin{bmatrix}
P_{r,j} \\
P_{g,j} \\
P_{b,j}
\end{bmatrix} =
\begin{bmatrix}
V_{r,j} \\
V_{g,j} \\
V_{b,j}
\end{bmatrix}
\begin{bmatrix}
C_{r,j} \\
C_{g,j} \\
C_{b,j}
\end{bmatrix}
- \begin{bmatrix}
F_{r,j} \\
F_{g,j} \\
F_{b,j}
\end{bmatrix},
\]

where we assume that the projector responses are linearized by using a conventional color calibration technique in advance. Once we generate the desired color intensity \( C_{\mu,j} \) according to Eq. (4) at each pixel, we control color brightness \( P_{k,j} \) output from the projector by using Eq. (6). By applying the above compensation technique to the generated image with light directional vector \( \mathbf{l}_i \), we can obtain the desired image sequence based on the principle of photometric stereo.

3. Our Method

3.1 Matrix Decomposition by using Rank-3 Approximation

According to the format of image data matrix used in 3), we arrange all observed pixels \((C_{r,j}, C_{g,j}, C_{b,j}), j \in \{1, 2, \cdots, P\}\), under all illumination lights \( \mathbf{l}_i, i \in \{1, 2, \cdots, M\}\). Then when we reformat all of the pixels as a \( 1 \times 3P \) row-vector \([C_{r,1}, C_{g,1}, C_{b,1}, \cdots, C_{r,P}, C_{g,P}, C_{b,P}]\) and rearrange all of the row-vectors in sequential order, we obtain the following extended image data matrix

\[
\mathbf{I} = \begin{bmatrix}
C_{r,1} & C_{g,1} & \cdots & C_{r,P} & C_{g,P} & C_{b,P} \\
C_{r,2} & C_{g,2} & \cdots & C_{r,P} & C_{g,P} & C_{b,P} \\
\vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\
C_{r,M} & C_{g,M} & \cdots & C_{r,P} & C_{g,P} & C_{b,P}
\end{bmatrix},
\]

by matrix representation. Since elements \( C_{\mu,j} \) are given by Eq. (4), we can describe matrix \( \mathbf{I} \) defined in Eq. (7) as the product of two matrices as follows,

\[
\mathbf{I} = \mathbf{L} \mathbf{N},
\]

\[
\mathbf{L} = \begin{bmatrix}
\mathbf{l}_1 & \mathbf{l}_2 & \cdots & \mathbf{l}_i & \cdots & \mathbf{l}_M
\end{bmatrix}^T,
\]

\[
\mathbf{N} = \begin{bmatrix}
r_1 \mathbf{n}_1 & g_1 \mathbf{n}_1 & b_1 \mathbf{n}_1 & \cdots & r_P \mathbf{n}_P & g_P \mathbf{n}_P & b_P \mathbf{n}_P
\end{bmatrix}.
\]

Matrix \( \mathbf{L} \) is the light-source directional matrix, and matrix \( \mathbf{N} \) is the surface normal matrix scaled up to RGB pixel values \((r_j, g_j, b_j)\) in the surface texture image. Eqs. (8) ~ (10) show that \( \mathbf{M} \times 3 \) matrix \( \mathbf{I} \) is decomposed into the \( \mathbf{M} \times 3 \) matrix \( \mathbf{L} \) and the \( 3 \times 3 \) matrix \( \mathbf{N} \), and that its rank is at most three.

Based on Eqs. (8) ~ (10), we describe here a method that reconstructs the textured 3D surface from image data matrix \( \mathbf{I} \). We decompose the image data matrix \( \mathbf{I} \) into the following matrices:

\[
\mathbf{I} = \mathbf{U} \mathbf{\Lambda} \mathbf{V}^T,
\]

\[
\mathbf{U} = \begin{bmatrix}
\mathbf{U}_{(M \times 3)} & \mathbf{U}_{(M \times M')} \end{bmatrix},
\]

\[
\mathbf{\Lambda} = \begin{bmatrix}
\mathbf{\Lambda}_{(3 \times 3)} & \mathbf{0}_{(3 \times M')} \\
\mathbf{0}^T_{(3 \times M')} & \mathbf{\Lambda}_{(M' \times M')}
\end{bmatrix},
\]

\[
\mathbf{V} = \begin{bmatrix}
\mathbf{V}_{(3 \times 3P)} \\
\mathbf{V}_{(M' \times 3P)}
\end{bmatrix},
\]

where \( \mathbf{U} \) is an orthogonal matrix, \( \mathbf{\Lambda} \) is a diagonal matrix, and \( \mathbf{V} \) is an orthogonal matrix. Each \( \mathbf{U} \) and \( \mathbf{\Lambda} \) are extracted by singular value decomposition (SVD) of the image matrix \( \mathbf{I} \). In this case, the maximum rank of \( \mathbf{\Lambda} \) is \( 3 \), and because \( \mathbf{N} \) is scaled up to RGB, \( \mathbf{\Lambda} \) is also scaled up to RGB. By means of this characteristic, we can reconstruct the surface matrix \( \mathbf{N} \) by using \( \mathbf{\Lambda} \) and \( \mathbf{V} \). We can also reconstruct the light-source matrix \( \mathbf{L} \) by using \( \mathbf{U} \) and \( \mathbf{\Lambda} \).
by applying singular value decomposition (SVD), where we replace $M - 3$ with $M'$. In Eqs. (12) $\sim$ (14), each matrix appended sign $(a \times b)$ indicates that it is $a \times b$ matrix. As Eq. (8) shows that the rank of matrix $I$ is at most three, we obtain

$$I \approx U_{(M \times 3)} A_{(3 \times 3)} V_{(3 \times 3P)},$$

(15)

as the rank-3 approximation for matrix $I$. From Eq. (15), we have the temporary light-source directional matrix $\hat{L}$ and the temporary surface normal matrix $\hat{N}$ by

$$\hat{L} = U_{(M \times 3)},$$

(16)

$$\hat{N} = A_{(3 \times 3)} V_{(3 \times 3P)}.$$  

(17)

However, the matrix decomposition given by Eqs. (16) and (17) is not unique. In fact, if $Q$ is any invertible $3 \times 3$ matrix, matrices $\hat{L}Q$ and $Q^{-1}\hat{N}$ are also a valid decomposition of image data matrix $I$, since we have

$$\hat{L}Q(Q^{-1}\hat{N}) = \hat{L}(QQ^{-1})\hat{N} \approx \hat{L}\hat{N} \approx I.$$  

(18)

Thus, matrix $\hat{L}$ and matrix $\hat{N}$ are, in general, different from matrix $L$ and matrix $N$. Note that matrix $L$ and matrix $\hat{L}$ span the column space of the image data matrix $I$. Since the column space is three-dimensional because of the rank theorem, matrix $L$ and matrix $\hat{L}$ are different bases for the same space, and there must be a linear transformation between them.

As mentioned above, matrix $\hat{L}$ is a linear transformation of the true light-source directional matrix $L$. Likewise, matrix $\hat{N}$ is a linear transformation of the true surface normal matrix $N$. Consequently, the matrix decomposition given by Eqs. (16) and (17) recovers the light-source directional matrix and the surface normal matrix of up to $3 \times 3$ linear transformation matrix $Q$, namely,

$$L = \hat{L}Q,$$

(19)

$$N = Q^{-1}\hat{N}.$$  

(20)

3.2 Lights for Metric Constraints

We use metric constraints in determining matrix $Q$ to ensure that the two matrices computed by Eqs. (19) and (20) are valid decompositions of matrix $I$. In order to robustly obtain matrix $Q$ even in the presence of image noise, we need some strong surface scene or lighting constraints. As shown in Fig. 1(b), according to our concept that we can flexibly control the illumination lights by the projector in view of any camera; thus we compute matrix $Q$ by using metric constraints with the light-source directional matrix $L$.

In this paper, under the assumption that the number of observed images, $M$, is even, we employ the metric constraints associated with each pair of the directional vector of odd and even frames. Generally, directional vector

$$l_i = \begin{bmatrix} \sin(\theta_i) \cos(\phi_i) \\ \sin(\theta_i) \sin(\phi_i) \\ \cos(\theta_i) \end{bmatrix},$$

(21)

is described in terms of semi-spherical polar coordinates. In Eq. (21), we denote $i = 2k - 1$ and $i = 2k$ for odd and even frames, respectively. For each pair of directional vectors that correspond to odd and even frames, we set the following phase relation:

$$\theta_{2k} = \theta_{2k-1} = \theta,$$

(22)

$$\phi_{2k} = \phi_{2k-1} + \frac{\pi}{2}, \quad 0 \leq \phi_i < 2\pi,$$

(23)

where $k = 1, 2, \ldots, K (= M/2)$. We assume that angle $\theta$ is restricted to $0 < \theta < \pi/2$ and is constant in the system. As directional angle $\phi_{2k-1}$ may be arbitrarily selected in $0 \leq \phi_i < 2\pi$, we here vary angle $\phi_{2k-1}$ in the directional light vector $l_{2k-1}$ at a constant angle as if the vector projected on $XY$-plane circulates, as shown in Fig. 2.

Next, we give metric constraints for the above directional vectors of illumination. We assume that the temporary light-source directional matrix is given by

$$\hat{L} = \begin{bmatrix} I_1 & I_2 & \cdots & I_{2k-1} & I_{2k} & \cdots & I_{M-1} & I_M \end{bmatrix}^T,$$

(24)

in Eq. (16). Directional pair $\hat{I}_{2k-1}^T Q$ and $\hat{I}_{2k}^T Q$ determined by Eq. (19) should be identified with odd vector $l_{2k-1}$ and even vector $l_{2k}$ set by Eqs. (21) $\sim$ (23). That is, we get the following metric constraints

$$\hat{I}_{2k-1}^T QQ^T \hat{I}_{2k-1} = 1,$$

(25)

$$\hat{I}_{2k}^T QQ^T \hat{I}_{2k} = 1,$$

(26)

$$\hat{I}_{2k-1}^T QQ^T \hat{I}_{2k} = \cos^2(\theta),$$

(27)
from the norm of each directional vector and inner product between the directional pair.

The above metric constraints hold for directional vectors in the sequence \( k = 1, 2, \ldots, K \). When we use \( M \) frames observed by the camera, we have linear systems of \( 3K (= 3M/2) \) equations based on Eqs. (25) \(~\sim (27)\). By solving the simultaneous equations, we uniquely obtain transformation matrix \( Q \), see Appendix B for details.

3.3 Textured 3D Reconstruction

Given transformation matrix \( Q \), as we obtain valid surface normal matrix \( N \) from Eq. (20), we recover both surface normal \( \mathbf{n}_j \) at each pixel and the pixel value \((r_j, g_j, b_j)\) from matrix \( N \). Note that the metric constraints determined by Eqs. (25) \(~\sim (27)\) have no direct connection with angle \( \phi_j \). In other words, even if we use transformation matrix \( Q \) satisfying the metric constraints, each directional vector in the matrix \( L \) given by Eq. (19), might rotate around the \( Z \)-axis due to an offset angle. To avoid this problem, we use the following matrix:

\[
Q_0 = \begin{bmatrix}
1^T e_1 & 1^T e_2 & 0 \\
1^T e_2 & 1^T e_2 & 0 \\
0 & 0 & 1
\end{bmatrix}^{-1} \begin{bmatrix}
\sin(\theta) & 0 & 0 \\
0 & \sin(\theta) & 0 \\
0 & 0 & 1
\end{bmatrix},
\]

with \( e_1 = [1, 0, 0]^T \) and \( e_2 = [0, 1, 0]^T \), so that the initial directional pair \( I_1 \) and \( I_2 \) in the adjusted matrix \( LQ_0 \) equal \([\sin(\theta), 0, \cos(\theta)]^T\) and \([0, \sin(\theta), \cos(\theta)]^T\), respectively.*

Suppose that \( Q_0^{-1}N \) yields the surface normal matrix

\[
\hat{N} = \begin{bmatrix}
\hat{n}_i, \hat{n}_g, \hat{n}_b, \ldots, \hat{n}_r, \hat{n}_g, \hat{n}_p
\end{bmatrix},
\]

after calculating Eq. (20). The new normal matrix \( \hat{N} \) should be identical to Eq. (10). Note that the \( j \)-th pixel vectors \( \hat{n}_{\mu_j}, \mu \in \{r, g, b\} \), in the surface normal matrix are scaled up to the color channel’s brightness.

Since each surface normal \( \hat{n}_{\mu_j} \) in matrix \( \hat{N} \) is a unit vector as shown in Eq. (2), we get each pixel value \( \mu_j \) by calculating the following inner product:

\[
\mu_j = \hat{n}_{\mu_j}^T \frac{\hat{n}_{\mu_j}}{||\hat{n}_{\mu_j}||}, \quad \mu \in \{r, g, b\},
\]

where \( ||\hat{n}_{\mu_j}|| \) is the norm of vector \( \hat{n}_{\mu_j} \). We then obtain surface normal \( \mathbf{n}_j \) at the \( j \)-th pixel, by computing the average of

\[
\mathbf{n}_j = \frac{1}{3} \left( \frac{\hat{n}_r}{r_j} + \frac{\hat{n}_g}{g_j} + \frac{\hat{n}_b}{b_j} \right),
\]

from Eqs. (29) and (30).

After we obtain all RGB pixel values and surface normals by using Eqs. (30) and (31), we reconstruct the surface geometry \( Z_j \) defined in Eq. (1) by applying the path integral to the sets of surface normals, which is equivalent to solving Poisson equations\(^{10}\).

3.4 Algorithm

We summarize a textured 3D reconstruction algorithm based on the above derived equations as follows:

(1) Capture an image sequence and get image data matrix \( I \) defined by Eq. (7).

(2) Decompose matrix \( I \) by using SVD, and obtain the rank-3 approximation by Eq. (15).

(3) Set the temporary light-source directional matrix \( \hat{L} \) and the temporary surface normal matrix \( \hat{N} \) according to Eqs. (16) and (17).

(4) Compute 3 x 3 transformation matrix \( Q \) by solving simultaneous equations based on Eqs. (25) \(~\sim (27)\) for the image sequence.

(5) Obtain the valid surface normal matrix \( Q_0^{-1}N \) computed by Eqs. (17), (20), and (28), and then compute all RGB pixel values \((r_j, g_j, b_j)\) and surface normals \( \mathbf{n}_j \), by using Eqs. (30) and (31).

(6) Reconstruct the surface geometry \( Z_j \) in Eq. (1) by applying the path integral to the sets of surface normals.

4. Experiments and Results

4.1 Computer Simulation

We implemented the textured 3D reconstruction method described in Section 3 on a personal computer, OS: Windows XP (32-bit), CPU: Intel Xeon X5365 (3 GHz), Memory: 3.25 GB. We conducted computer simulations to examine the performance of our proposed method. Given 3D unimodal surface \( Z_j \) and the surface texture shown in Fig. 3, we obtained each surface normal \( \mathbf{n}_j \) in Eq. (2) by calculating \( \partial f/\partial X_j \) and \( \partial f/\partial Y_j \), which yielded the difference between neighboring pixels on the depth map \( Z_j = f(X_j, Y_j) \). We generated 20 images with various light directions according to Eq. (4), and then added Gaussian image noise with 0 mean and \( \sigma \) standard deviation to each pixel in the image sequence. After we performed 100 independent trials, we measured RMS errors of the 3D surface and the surface texture yielded by our method.

First, as we give inclination angle \( \theta \) a constant value in the systems, we checked the performance of our method by using computer-simulated data with various
values of angle $\theta$ and a constant $A = 5$ in the 3D surface. Fig. 4 shows examples of the generated images, where a gray-scale image of $100 \times 100$ pixels is used as the surface texture. Fig. 5 shows the RMS errors estimated under noise level $\sigma = 1.0 \sim 4.0$. The RMS errors of depth gradually decreased in $10^\circ \leq \theta \leq 50^\circ$ and suddenly increased when $\theta > 80^\circ$. In Fig. 5(a), we found that the inclination angle is limited to $30^\circ \leq \theta \leq 80^\circ$ if the RMS error is to be less than 0.1 under each noise level. When $\theta < 10^\circ$ or $\theta > 80^\circ$ in directional vectors, our method failed to offer high quality 3D reconstruction. Note that the 3D object is almost featureless in the sample images with $\theta = 10^\circ$; see Fig. 4. It is hard to confirm any variation in the generated images because they did not include shading that would yield sufficient information according to illumination direction. Since random image noise is present, the method probably did not work for the image sequence. As inclination angle increases, the shading of the 3D object becomes visible in Fig. 4. However, sample images with $\theta = 80^\circ, 85^\circ$, which are greatly different from the original surface texture, have vivid dark shadings for each illumination direction. Although our method did not attain good 3D reconstruction with $\theta = 85^\circ$, it was successful with $\theta = 80^\circ$. It is remarkable that our method could offer good quality 3D reconstruction right up to the point at which the depth error suddenly increased. On the contrary, in Fig. 5(b), our method maintained almost constant and small color brightness errors with $40^\circ \leq \theta \leq 80^\circ$. The errors gradually grow with $40^\circ \leq \theta \leq 30^\circ$ and suddenly increase when $\theta > 80^\circ$. This simulation demonstrated that the quality of texture yielded by our method deteriorates when the shading determined by illumination direction is dark. From these results, we confirmed that we should set inclination angle $\theta$ to $30^\circ \leq \theta \leq 60^\circ$ in order to accurately reconstruct both surface depth and the texture. Given this experience, we used $\theta = 45^\circ$ in
the directional lights are parallel to the Z-axis. That is, the variation of shading in the image sequence does not support Structure from Shading. If $\theta \to 90^\circ$, then the right side of Eq. (27) equals 0. As the directional vectors of the pair approach orthogonal vectors, the geometric constraints work effectively and ensure valid matrix decomposition. However, the orthogonality means that illumination lights are perpendicular to the Z-axis direction on the object. Note that application of photometric stereo is limited to convex objects that satisfy $l^T n_i > 0$. If the inner product between the illumination light and normal on flat surface becomes to be zero, the system cannot display the pixel value by using the photometric model described in Section 2.

Second, we conducted a computer simulation with $\theta = 45^\circ$ while varying parameter $A$ in the 3D surface model. Fig. 6 shows examples of the image sequence used in the simulation. Fig. 7 plots RMS error versus parameter $A$. As shown in Fig. 7(a), the RMS curve is convex downward, where the minimum is shifted to the right for each noise level. As the height of surface increased, the surface shape gradually appears in the image sequence; see Fig. 6. Note that the 3D object with $A = 1$ is almost featureless in the sample images. As these smooth images were randomly corrupted by image noise, our method could not yield good 3D reconstruction. When $A \geq 3$, our method worked well for image sequences with sharp shading. Fig. 7(b), on the other hand, shows a constant characteristic for each noise level. The RMS error tends to be constant, as parameter $A$ increased. These simulations demonstrated that if our method processes an image sequence generated from a sufficient convex 3D object, it can robustly reconstruct the textured 3D object regardless of the noise level.

4.2 Real Images

(1) Our Experimental System

Fig. 8 shows our projection-based display system; it consisted of an LCD projector (SONY VPL-F-41), an IEEE1394b camera (Point Grey GRAS-50S5C-C), and a screen. The system captures one full color image per second, with a resolution of $1,600 \times 1,200$ pixels. We estimated plane-homographies among the projector, the camera, and the screen in the system configuration shown in Fig. 8(b), and used these plane-homographies to handle the pixels on the images output from the projector and the images observed by the camera. For the experimental system, we also measured the color channel’s luminance from the projector light.
by installing a spectroradiometer (PR655 SpectraScan) in order to linearize the projector responses. We compensated the image sequence by applying the compensation method using a color-mixing matrix \( (2)^{13} \), which uses the inverse response functions of the projector. 

Fig. 9(a) and Fig. 9(b) show examples of the desired image for textured 3D reconstruction and compensated image captured by the camera, respectively. By employing plane-homographies in the system, we extracted the pixels in a rectangular frame, i.e. the screen in Fig. 9(b) and obtained the geometrically-warped image shown in Fig. 9(c). We used 20 corrected frame images as the image sequence in subsequent experiments. Actually, we obtained image data matrix \( I \) by adequately sub-sampling each corrected image; the memory limits of the computer that implemented our algorithm prevented it from handling the entire corrected image.

(2) Artificial Data

First, we confirmed whether our textured 3D reconstruction method worked well in the above experimental system by using simple artificial data. Our method uses radiometric compensation to obtain the desired image sequence from the projection-based display system, and it yields the 3D surface normal and the texture by using metric constraints with the light-source directional matrix \( L \). Thus we implemented several methods in which these functions were modified and carried out experimental comparisons of textured 3D reconstruction. Table 1 shows the functions used. Method-A and Method-B handle observed images that are not compensated. Method-B and Method-C use light-source directional matrix \( L \), from which the desired image sequence is also generated, in unique matrix decomposition. Af-

<table>
<thead>
<tr>
<th>Table 1</th>
<th>Methods compared.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Items</td>
<td>Compensation</td>
</tr>
<tr>
<td>Method-A</td>
<td>unused</td>
</tr>
<tr>
<td>Method-B</td>
<td>unused</td>
</tr>
<tr>
<td>Method-C</td>
<td>used</td>
</tr>
<tr>
<td>Our Method</td>
<td>used</td>
</tr>
</tbody>
</table>
Fig. 13  Textured 3D objects reconstructed by each method.

For reference, we plot the singular values of matrix $I$ used in each method in descending order in Fig. 11. For reference, we plot the singular values of the ideal image data matrix composed of the desired images with no image noise. Both the uncompensated and compensated image data matrices showed that the singular value of rank-4 is very small compared to that of rank-3. As shown in Eqs. (8) ~ (10), color pixels in the surface texture do not impact rank-3 decomposition. This advanced test confirms that the rank-3 approximation of both uncompensated and compensated image data matrices is valid for matrix decomposition.

Next, we compared each directional vector $l_i$ in the light-source directional matrix $L$ yielded by Method-A and our method. In Fig. 12, we show the estimated directional angles $\phi_i$, $i \in \{1, 3, \cdots, 19\}$ for each method. Both methods yielded incremental angles that were linear in terms of frame response and obtained almost the same directional angles for all frames as the ground truth. The metric constraints determined by Eqs. (25) ~ (27) do not use color brightness in surface texture. Hence, Method-A also accurately recovered the directional angles even though it used the uncompensated image data matrix. The calculated RMS error between ground truth and the estimated angles was 0.32° for Method-A and 0.15° for our method. Eq. (19) robustly and accurately yielded the directional angles of projec-
Fig. 15 Examples of image sequences generated from real 3D objects.

In Fig. 13, we show the textured 3D reconstruction results yielded by each method. At a glance, each method seems to have generated five unimodal shapes whose surfaces are similar to those of the ground truth. However, Method-A and Method-B reconstructed different mountain heights and surface distortion in flat zones. In contrast, both Method-C and our method yielded superior 3D reconstruction results. As the given light-source directional matrix $L$ allowed Method-C to achieve valid matrix decomposition of the compensated image data matrix, Method-C probably worked well in reconstructing 3D objects. In our method, the metric constraints associated with each pair of the directional lights offer good matrix decomposition and robustly reconstruct the surface normal of each pixel. Furthermore, as shown in Fig. 12, our method estimated the light-source directional matrix with high accuracy.
These results demonstrate that the proposed method can simultaneously and accurately reconstruct light-source directional matrix $L$ and surface normal matrix $N$ from the image data matrix. Fig. 14 shows the surface texture yielded by each method. As Method-A and Method-B used uncompensated image data matrix, their recovered texture images differed from the desired image. On the other hand, Method-C and our method well matched the ground truth; they are supported by the radiometric compensation of projection-based display system.

(3) Real Objects

We used a 3D laser scanner (KONICA MINOLTA VIVID910) to measure 3D coordinates on the surface of real objects and the surface texture. First, we interpolated dense 3D points from the measured 3D point cloud and obtained depth map $Z_j = f(X_j, Y_j)$ on the $XY$-plane, which is defined as the reference plane in the object’s coordinate system. Next, we selected some feature points on the depth map and extracted the corresponding points on the texture image. By using the geometric transformation calculated from the reference points, we pasted the surface texture onto the depth map. After completing the processing from 3D point measurement to image registration, we obtained a textured 3D object as the ground truth. We examined three textured 3D objects: miniature’s face, doll of giraffe, and human face. We show examples of each image sequence generated from the depth map and the surface texture, “Desired”, in Fig. 15. We used the experimental system in Fig. 8 to observe uncompensated and compensated image sequences; the former is used in Method-A and Method-B, and the latter is used in Method-C and the proposed method. In Fig. 16, we show the singular values of real image data matrix $I$ used in each method in descending order, as compared to those of the ideal image data matrix with no image noise. We confirmed that the rank-3 approximation of both uncompensated and compensated image data matrices is basically valid for matrix decomposition.

We show the textured 3D reconstruction results yielded by each method in Fig. 17, Fig. 18, and Fig. 19. Method-A and Method-B have slight distorted surfaces at the hollows around the eyes in Fig. 17, while Method-C and our method basically match the ground truth. As shown in Fig. 18, Method-A and Method-B yielded distorted shapes at the legs and head of the giraffe, and placed a convexo-concave surface in the background. In Fig. 19, each method output remarkably different shapes at the nose and mouth. For instance, Method-A and Method-B created an inflated nose. In contrast, Method-C and our method yielded the same nose and mouth as the ground truth. Table 2 shows RMS errors calculated between ground truth and each reconstructed object. For all samples, our method exhibits the same performance as Method-C, which recovers surface normal matrix from compensated image data matrix by using known light-source directional matrix. Namely, these results show that metric constraints based on directional light pairs in odd and even frames work robustly and effectively in the matrix decomposition of the image data matrix.

Fig. 20 shows the surface textures yielded by each method, and Table 3 summarizes the RMS errors between the image recovered by each method and the ground truth. Method-A and Method-B recovered texture images that differed from the ground truth. Method-C and our method output good surface texture images, which were close to the ground truth. In Fig. 21, we show the directional angles $\phi_i$ computed by our method for only odd frames. The method estimated the directional angles with high accuracy for all image sequences. Since our proposed method could reconstruct each textured 3D object in seconds, these experiments on real objects ensure that it can robustly and promptly yield not only good textured 3D reconstruction but also directional lights with high accuracy for projection-based display systems.

5. Conclusion

We aim to realize a projection-based display system that can provide interactive services in which the user’s camera is linked to locations in the real world. In order for the image sequence displayed by the system to yield convex 3D objects with surface texture, we extended
Fig. 17  Textured 3D objects reconstructed by each method.

Fig. 18  Textured 3D objects reconstructed by each method.
the photometric model used in photometric stereo to handle full color images. According to the photometric model, we generate multiple shading images from 3D surface with color texture. The system uses a radiometric compensation technique so that the camera captures the desired image sequence. We showed that an image data matrix created from multiple images can be decomposed into a surface normal matrix scaled up to the color intensity of three channels and a light-source directional matrix, and that its rank is at most three.

We introduced new and robust metric constraints based on the directional light pairs to ensure unique matrix decomposition against image noise. Based on matrix decomposition, we described a new textured 3D reconstruction method that obtains surface normals on 3D shape and the texture from the image data matrix.

We provided the results of experiments on both computer-simulated data and real images. First, we conducted computer simulations to examine the per-
formance of our textured 3D reconstruction method. The results confirmed that the proposed method needs an adequate inclination angle $\theta$, e.g., $30^\circ \leq \theta \leq 60^\circ$, in order to accurately reconstruct both 3D surface and texture. By using an image sequence with depth parameter $A$ for convex 3D objects, we found that the method can robustly reconstruct the textured 3D object regardless of each noise level. Next, we implemented a projection-based display system in an experimental environment, and conducted experiments to test our method and several comparative methods when challenged with real images observed by the system. This experiment demonstrated that our proposed method can robustly and effectively yield not only 3D surface with texture but also the directional angles of illumination light. Needless to say, our concept also suits the image data matrix, which is likely to degrade not only the recovery of light direction but also textured 3D reconstruction. As the captured images are also geometrically-warped owing to the camera’s or projector’s orientation in the system, we need to study an appropriate configuration for ensuring desired 3D reconstruction in projection-based display systems. Through field tests, we intend to enhance our method to handle image sequences captured by hand-held cameras. Future work includes flexible color calibration to indirectly realize radiometric compensation so that the user’s camera can attain the desired image sequence. Considering that the image sequences will be captured with arbitrary timing, our proposed system also needs to automatically identify odd and even frames.

Acknowledgments

The authors would like to thank H. Yabushita, who was a colleague of our laboratories, for her help in measuring 3D points on the surface of real objects by using a 3D laser scanner. In addition, the constructive comments and advice from the editor and reviewers are gratefully acknowledged.

Appendix A: Surface Normal

This section provides an additional explanation of the surface normal given by Eqs. (2) and (3). Given the 3D coordinate of the $j$-th point on the object by

$$\mathbf{P}_j = \begin{bmatrix} X_j, & Y_j, & Z_j \end{bmatrix}^T,$$

we get vector $\mathbf{n}_{X_j}$ in the direction of the X-axis by

$$\mathbf{n}_{X_j} = \frac{\partial \mathbf{P}_j}{\partial X_j} = \begin{bmatrix} \frac{\partial X_j}{\partial X_j}, & \frac{\partial Y_j}{\partial X_j}, & \frac{\partial Z_j}{\partial X_j} \end{bmatrix}^T$$

$$= \begin{bmatrix} 1, & 0, & \frac{\partial f}{\partial X_j} \end{bmatrix}^T,$$

from Eq. (1). In the same way, we get vector $\mathbf{n}_{Y_j}$ in the direction of the Y-axis by

$$\mathbf{n}_{Y_j} = \frac{\partial \mathbf{P}_j}{\partial Y_j} = \begin{bmatrix} \frac{\partial X_j}{\partial Y_j}, & \frac{\partial Y_j}{\partial Y_j}, & \frac{\partial Z_j}{\partial Y_j} \end{bmatrix}^T$$

$$= \begin{bmatrix} 0, & 1, & \frac{\partial f}{\partial Y_j} \end{bmatrix}^T.$$ 

Thus we have vector $\mathbf{n}_{Z_j}$ in the direction of the Z-axis by using the following outer product:

$$\mathbf{n}_{Z_j} = \mathbf{n}_{X_j} \times \mathbf{n}_{Y_j} = \begin{bmatrix} -\frac{\partial f}{\partial X_j}, & -\frac{\partial f}{\partial Y_j}, & 1 \end{bmatrix}^T.$$

By computing the normalization of the vector $\mathbf{n}_{Z_j}$, we obtain the surface normal

$$\mathbf{n}_j = \frac{\mathbf{n}_{Z_j}}{||\mathbf{n}_{Z_j}||},$$

at the $j$-th point on the object, where $||\mathbf{n}_{Z_j}||$ is the norm of the vector $\mathbf{n}_{Z_j}$.

Appendix B: Computing matrix $Q$ based on metric constraints

For convenience of explanation, we give the vectors $\mathbf{I}_{2k-1}$ and $\mathbf{I}_{2k}$ in Eq. (24) as follows,

$$\mathbf{I}^T_{2k-1} = [l_{2k-1,1}, l_{2k-1,2}, l_{2k-1,3}],$$

$$\mathbf{I}^T_{2k} = [l_{2k,1}, l_{2k,2}, l_{2k,3}].$$

Since matrix $QQ^T$ is a $3 \times 3$ symmetric matrix, we denote $Q' = QQ^T$ by

$$Q' = \begin{bmatrix} q_1, & q_2, & q_3 \\ q_2, & q_4, & q_5 \\ q_3, & q_5, & q_6 \end{bmatrix}.$$

Given $M (= 2K)$ frames by the camera, Eqs. (25) ~ (27) yield the following simultaneous equations:
with the replacements of

\[ a_{k1} = l_{2k-1,1}, \quad a_{k2} = 2l_{2k-1,1}l_{2k-1,2}, \]
\[ a_{k3} = 2l_{2k-1,1}l_{2k-1,3}, \quad a_{k4} = l_{2k-1,2}, \]
\[ a_{k5} = 2l_{2k-1,2}l_{2k-1,3}, \quad a_{k6} = l_{2k-1,3}, \]
\[ b_{k1} = l_{2k,1}, \quad b_{k2} = 2l_{2k,1}l_{2k,2}, \quad b_{k3} = 2l_{2k,1}l_{2k,3}, \]
\[ b_{k4} = l_{2k,2}, \quad b_{k5} = 2l_{2k,2}l_{2k,3}, \quad b_{k6} = l_{2k,3}, \]
\[ c_{k1} = l_{2k-1,1}l_{2k,1}, \quad c_{k2} = l_{2k-1,1}l_{2k,2} + l_{2k-1,2}l_{2k,1}, \]
\[ c_{k3} = l_{2k-1,1}l_{2k,3} + l_{2k-1,3}l_{2k,1}, \quad c_{k4} = l_{2k-1,2}l_{2k,2}, \]
\[ c_{k5} = l_{2k-1,2}l_{2k,3} + l_{2k-1,3}l_{2k,2}, \quad c_{k6} = l_{2k-1,3}l_{2k,3}. \]

Solving Eq. (40) yields the least-square solution. After substituting \( q_1, q_2, \ldots, q_6 \) for Eq. (39), we can uniquely obtain the desired transformation matrix \( Q \) by carrying out Cholesky decomposition.

References