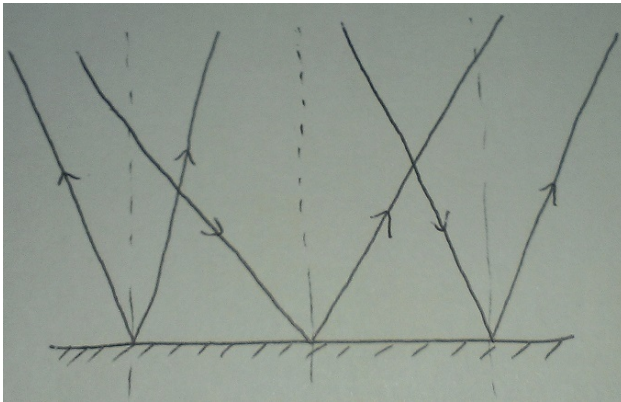


## Homework 3

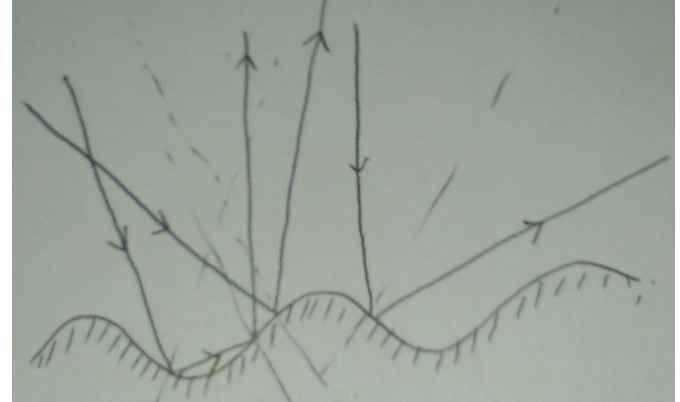
### 1 Theoretical Problems

#### 1.1

(a)



(a)



(b)

Figure 1: (a) **Specular reflection** on a smooth even surface; **Diffuse reflection** on an uneven surface.

Fig. 1 shows the scenarios for specular and diffuse reflection. For **specular reflection**, all of the light rays are reflected along same direction. For **diffuse reflection**, the direction of light rays is random and also reflect from the uneven surface. Since the concentration of light intensity is greater for specular reflection, the spot on the surface where it reflects from looks brighter.

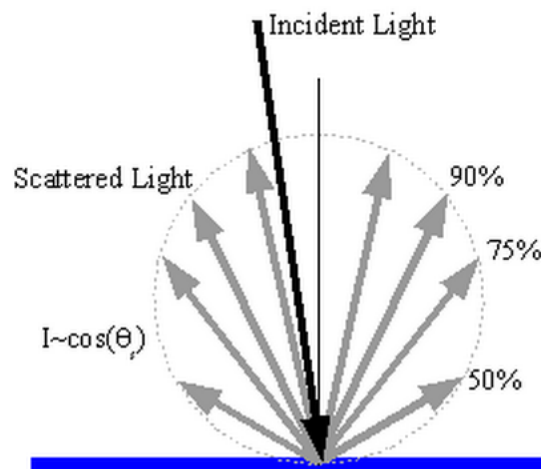


Figure 2: Lambertian, or diffuse, scatter of light is governed by a cosine intensity relationship

(b) For Lambertian surfaces, the directional hemispheric reflectance is independent of illumination direction. A formal model is given by *bidirectional reflectance distribution function*, or **BRDF** is

independent of outgoing direction. Therefore, the radiance( $\rho$ ) is independent of viewing angle. Fig. 2 shows an example of this.

## 1.2

(a) As derived in class, the radiosity( $B(x, y)$ ) can be written in terms of albedo( $\rho$ ) and the angles( $\theta_e, \theta_i$ ) as:

$$B(x, y) = \rho \frac{\cos \theta_e}{\cos \theta_i} \quad (1)$$

$$= \rho \frac{\langle \widehat{N}_L, \widehat{N} \rangle}{\langle \widehat{N}_l, \widehat{N} \rangle} \quad (2)$$

where,  $\widehat{N}_L$  denotes the normalized normal of light source direction,  $\widehat{N}_l$  denotes normalized normal of the lens direction and  $\widehat{N}$  denotes the surface normal at  $(x, y)$ . Assuming viewer centered geometry, eq. 2 reduces to :

$$\begin{aligned} B(x, y) &= \rho \frac{\langle (-p_L, -q_L, 1), (-p, -q, 1) \rangle}{\langle (-p_l, -q_l, 1), (-p, -q, 1) \rangle} \\ \Rightarrow B(x, y) &= \rho \cdot \frac{p_L p + q_L q + 1}{p_l p + q_l q + 1} \sqrt{\frac{p_L^2 + q_L^2 + 1}{p_l^2 + q_l^2 + 1}} \end{aligned} \quad (3)$$

Eq. 3 shows the reflectance map  $R(p, q)$ .

(b) To generate isodensity curves of constant brightness in  $R(p, q)$ , we set  $B(x, y) = c$ , where  $c$  is a constant. This reduces eq. 3 to:

$$c_1 \cdot (1 + p_l p + q_l q) = c_2 \cdot (1 + p_L p + q_L q) \quad (4)$$

where  $c_1 = c \cdot \sqrt{p_l^2 + q_l^2 + 1}$  and  $c_2 = \rho \cdot \sqrt{p_L^2 + q_L^2 + 1}$ . Eq. 2 shows that the isodensity curves of constant brightness are lines.

(c) The contours of constant brightness of an image of sphere are straight lines in the  $(p, q)$  frame.

(d) Fig. 6 shows the isocontours from multiple images.

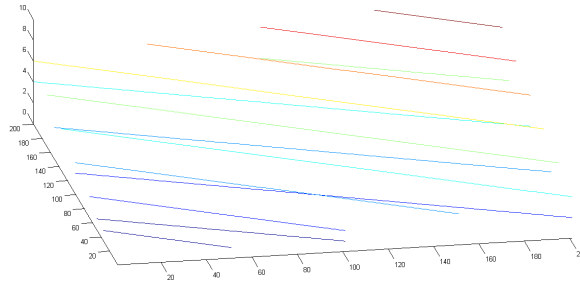


Figure 3: Lines from two different images

## 1.3

a) Given  $Z = -\frac{Bf}{d}$ , we can write the 1st derivative as  $\frac{\partial Z}{\partial d} = \frac{Bf}{d^2}$ . Change in depth( $\Delta Z$ ) with disparity( $\Delta d$ ) can be approximated as :  $\frac{\partial Z}{\partial d} = \lim_{\Delta d \rightarrow 0} \frac{\Delta Z}{\Delta d}$ , which says that  $\frac{\partial Z}{\partial d}$  approximates to  $\frac{\Delta Z}{\Delta d}$

under  $\lim_{\Delta d \rightarrow 0}$ . Fig. 4 shows the plot of the function  $\frac{\partial Z}{\partial d}$ . As seen change in depth decreases with change in disparity.

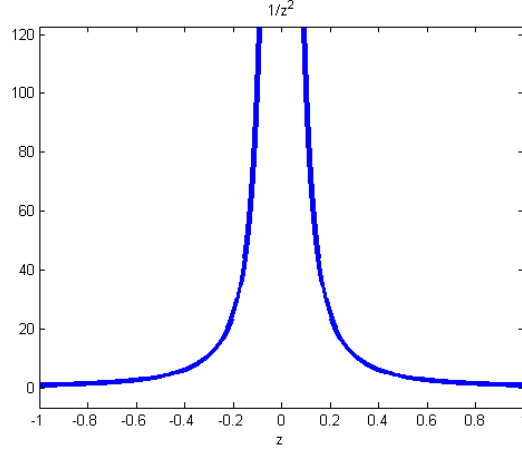


Figure 4: Plot of  $\frac{\partial Z}{\partial d}$  vs  $d$ .

b) The partial derivatives of  $Z$  wrt  $B, f$  and  $d$  look like:

$$\frac{\partial Z}{\partial d} = \frac{Bf}{d^2}, \quad \frac{\partial Z}{\partial B} = -\frac{f}{d}, \quad \frac{\partial Z}{\partial f} = -\frac{B}{d} \quad (5)$$

Plugging terms in eq. 5 into the error estimate for  $Z$ , we get:

$$\delta Z = \sqrt{\left(\frac{Bf}{d^2} \delta d\right)^2 + \left(\frac{f}{d} \delta B\right)^2 + \left(\frac{B}{d} \delta f\right)^2} \quad (6)$$

Eq. 6 shows that  $\delta Z$  has a linear relation with baseline width( $B$ ) and focal length( $f$ ).

(c) We can compute the relative error  $\delta Z$  as follows:

$$\begin{aligned} \frac{\delta Z}{Z} &= -\frac{d}{Bf} \sqrt{\left(\frac{Bf}{d^2} \delta d\right)^2 + \left(\frac{f}{d} \delta B\right)^2 + \left(\frac{B}{d} \delta f\right)^2} \\ &= -\sqrt{\left(\frac{\delta d}{d}\right)^2 + \left(\frac{\delta B}{B}\right)^2 + \left(\frac{\delta f}{f}\right)^2} \\ \Rightarrow \left\| \frac{\delta Z}{Z} \right\| &= \sqrt{\left(\frac{\delta d}{d}\right)^2 + \left(\frac{\delta B}{B}\right)^2 + \left(\frac{\delta f}{f}\right)^2} \end{aligned} \quad (7)$$

From eq. 7, if we want measure accuracy of features we assume  $\delta B = 0$  and  $\delta f = 0$ . This gives us  $\frac{\delta Z}{Z} = \frac{\delta d}{d}$ . So, the features should be measured with same accuracy as depth map in order to get a relative error  $< 1\%$ .

## 2 Practical Problems

The problem of generating shape from shading is solved in two stages. First, we have to generate the albedo map and the normals to the surface. Next, we generate a depth map from the above normals. I describe the approach to both these in order.

## 2.1 Generating Normals

For lambertian surfaces, intensity at any point on a surface is given as:

$$I = \rho \cdot \langle \hat{N}, \hat{N}_L \rangle \quad (8)$$

where  $\hat{N}$  is the normalized normal at the surface point,  $\hat{N}_L$  is the reflected light direction normalized. In order to determine  $\hat{N}$ , we use multiple ( $\geq 3$ ) light sources which are not in the same plane. Thus, eq. 8 can be written for multiple light sources as follows:

$$\begin{aligned} I_1 &= \rho \cdot (N_x \cdot L_{1x} + N_y \cdot L_{1y} + N_z \cdot L_{1z}) \\ I_2 &= \rho \cdot (N_x \cdot L_{2x} + N_y \cdot L_{2y} + N_z \cdot L_{2z}) \\ I_3 &= \rho \cdot (N_x \cdot L_{3x} + N_y \cdot L_{3y} + N_z \cdot L_{3z}) \\ &\vdots \\ I_n &= \rho \cdot (N_x \cdot L_{nx} + N_y \cdot L_{ny} + N_z \cdot L_{nz}) \end{aligned}$$

We fold the normal( $N$ ) and the albedo( $\rho$ ) into one vector ( $\mathbf{G}$ ). Assume  $\mathcal{V}$  denotes a vector of light source directions.

$$\mathcal{V} = \begin{pmatrix} L_1^T \\ L_2^T \\ \vdots \\ L_n^T \end{pmatrix}$$

For  $n > 3$ , we can solve the equations with least squares as follows:

$$\begin{aligned} \mathcal{V}\mathbf{G} &= \mathbf{I} \\ \implies \mathcal{V}\mathcal{V}^T\mathbf{G} &= \mathcal{V}^T\mathbf{I} \\ \implies \mathbf{G} &= (\mathcal{V}\mathcal{V}^T)^{-1}\mathcal{V}^T\mathbf{I} \end{aligned}$$

Therefore, the albedo( $\rho(x, y)$ ) and the normals to the surface( $N$ ) can be written as:

$$\rho(x, y) = \|\mathbf{G}\| \quad (9)$$

$$N(x, y) = \mathbf{G}/\rho(x, y) \quad (10)$$

## 2.2 Generating Depth Map

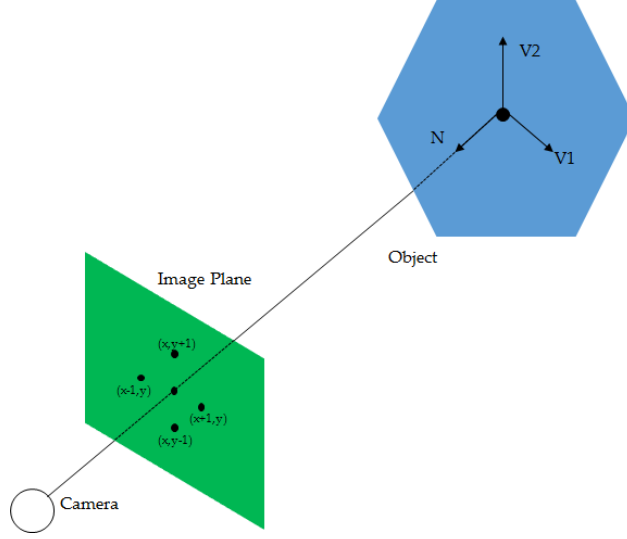


Figure 5: Schematic diagram showing the normals for the image plane, camera and 3D world object.

From Fig. 6, the normal at surface( $N$ ) must be orthogonal to  $V1$  which gives:

$$\begin{aligned}
 V1 &= (x+1, y, z_{(x+1,y)}) - (x, y, z_{(x,y)}) \\
 &= (1, 0, z_{(x+1,y)} - z_{(x,y)}) \\
 \implies N \cdot V1 &= 0 \\
 \implies (N_x, N_y, N_z) \cdot (1, 0, z_{(x+1,y)} - z_{(x,y)}) &= 0 \\
 \implies N_x + N_z(z_{(x+1,y)} - z_{(x,y)}) &= 0
 \end{aligned} \tag{11}$$

Again, the normal at surface( $N$ ) must be perpendicular to  $V2$  which gives:

$$\begin{aligned}
 V2 &= (x, y+1, z_{(x,y+1)}) - (x, y, z_{(x,y)}) \\
 &= (0, 1, z_{(x,y+1)} - z_{(x,y)}) \\
 \implies N \cdot V2 &= 0 \\
 \implies (N_x, N_y, N_z) \cdot (0, 1, z_{(x,y+1)} - z_{(x,y)}) &= 0 \\
 \implies N_y + N_z(z_{(x,y+1)} - z_{(x,y)}) &= 0
 \end{aligned} \tag{12}$$

Taking boundary conditions at pixels where  $N$ -values are not available, we take the forward direction for depth calculation, which gives:

$$-N_x + N_z(z_{(x-1,y)} - z_{(x,y)}) = 0 \tag{13}$$

$$-N_y + N_z(z_{(x,y-1)} - z_{(x,y)}) = 0 \tag{14}$$

Putting together Eqs. (11, 12, 13, 14) into a sparse matrix( $M$ ), we can get the following system of equations:

$$M \cdot z = v \tag{15}$$

Here  $M \in \mathbb{R}^{2 \times S, S}$ , where  $S$  is the size of the image,  $z$  is a vector( $|z| = S$ ) of pixels of the image, and  $v$  is a vector of length  $2 \times S$  of  $x$  and  $y$  components of normals( $N_x, N_y$ ) repeated for every pixel. It is easily seen that  $M$  is a sparse matrix. To speed up our computation, I use only those pixels in

the image which had moderately high intensity. These pixels are labeled together in a binary image, **mask**.

In order to solve Eq. 15, we invert matrix  $M$  using QR decomposition and solve for  $z$ . Figures (6,7,8) shows the results for synthetic-image, dog and sphere images respectively.

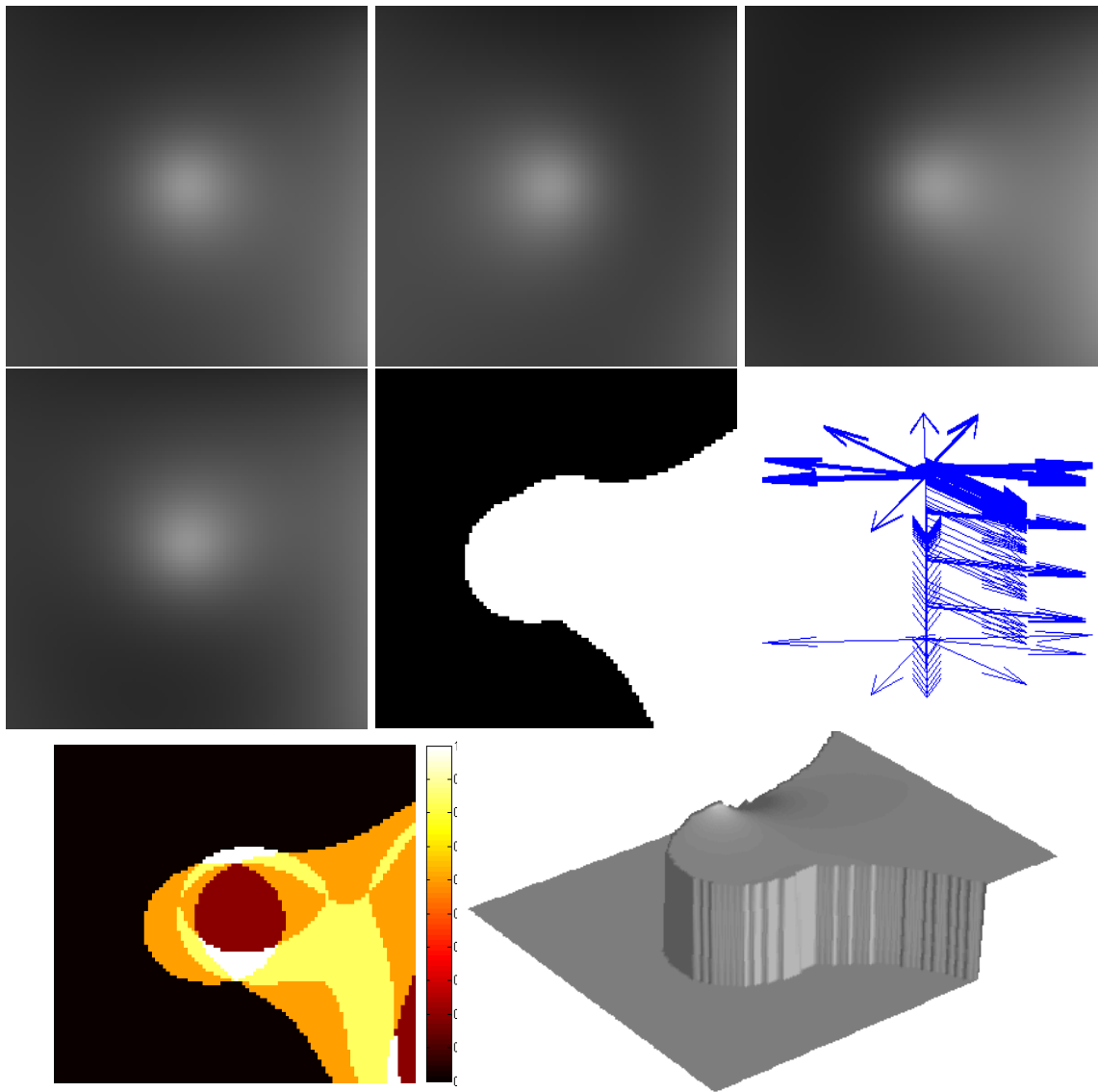


Figure 6: **Synth-images** dataset. Images shown in order are 4 original images, mask image, *quiver-plot* of surface normals, albedo map and the reconstructed depth.

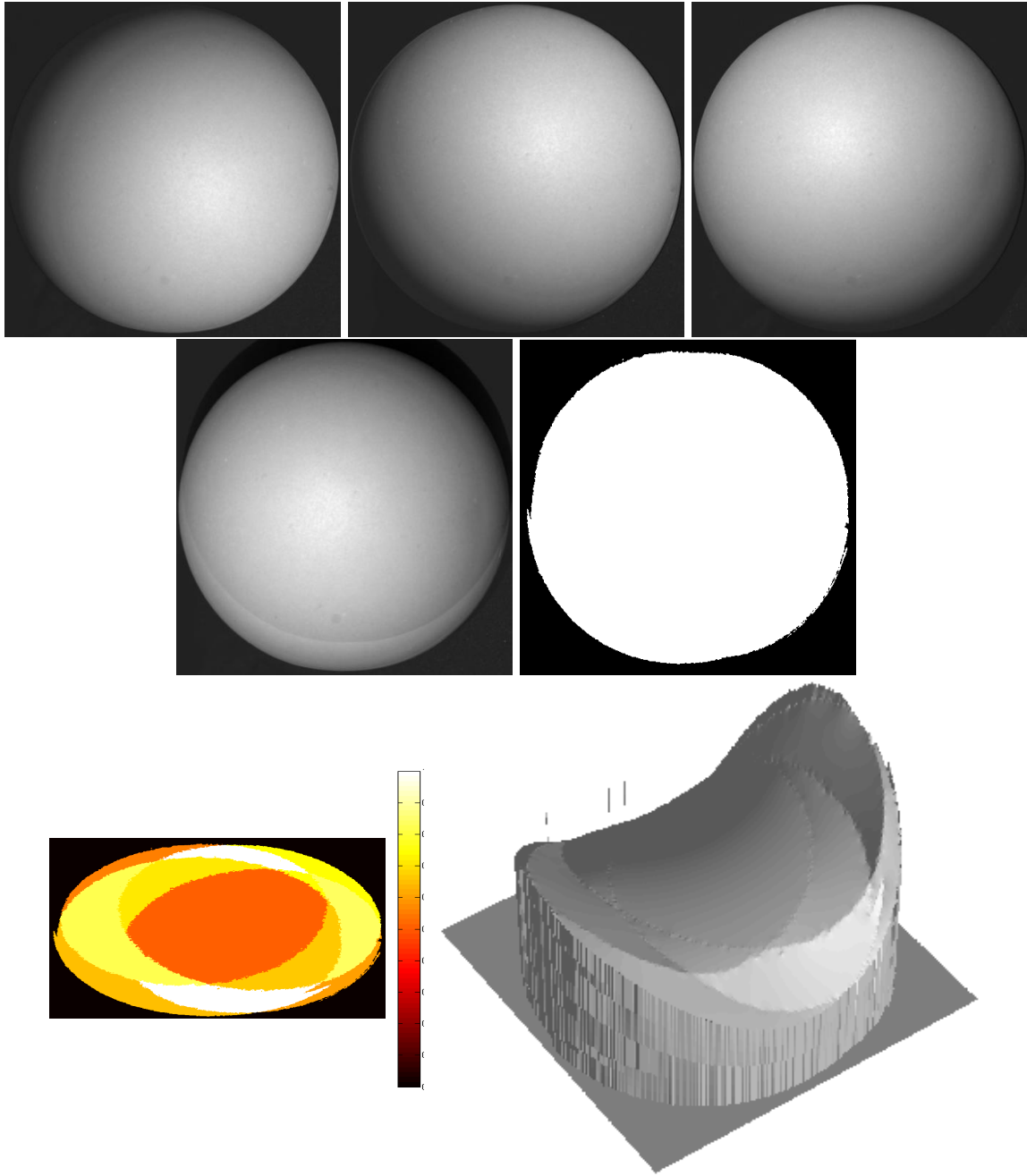


Figure 7: **Sphere-images** dataset. Images shown in order are 4 original images, mask image, *quiver-plot* of surface normals, albedo map and the reconstructed depth.

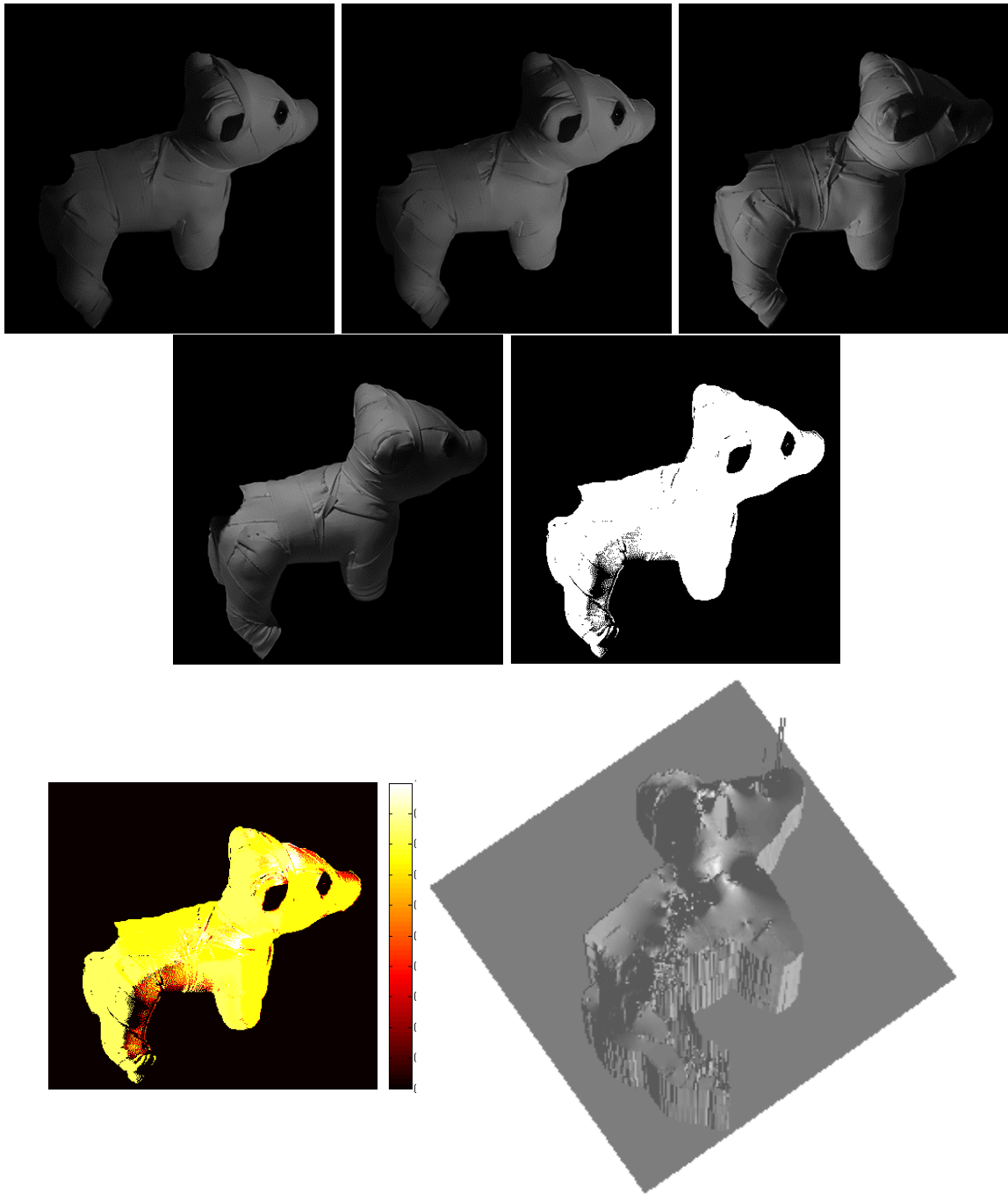


Figure 8: **Dog-png** dataset. Images shown in order are 4 original images, mask image, *quiver-plot* of surface normals, albedo map and the reconstructed depth.