Photometric Reconstruction in Real Time without SVD using GPU

Tomoaki Teshima
Hideo Saito
3-14-1 Kohoku Yokohama 223-8522,
Keio University, Japan
Email: {tomoaki, saito}@ozawa.ics.keio.ac.jp

Vincent Nozick
Venceslas Biri
Institute Gaspard Monge UMR 8049,
University Paris-Est Marne-la-Vallee, France
Email: {vnozick, biri}@univ-mlv.fr

***Abstract*—The proposed system reconstruct the 3D shape of the object in real time. The computation speed is accelarated compared to the conventional method since we process each pixel simultaneously on GPU.**

I. Introduction

We show a real time shape reconstruction method based on photometric reconstruction. In the proposed method, we reconstruct the target object by using the combination of laptop computer, webcam and GPU. We use the display as the lightning device and webcam to capture the images. Compared to the previous works[1],[2], we compute the normal vector without using SVD but based on the position of the lights

II. Theory

The photometric reconstruction was proposed[3]. Photometric reconstruction is based on the reflecton of the object. When the position of the light source and the camera is known and multiple images are obtained under different light conditions, the depth map is reconstructed as following. The reflectance on the surface of the object is expressed as Eq.(1).

$$\mathbf{L} = \rho \mathbf{S} \cdot \mathsf{n} \tag{1}$$

In Eq.(1), $\mathbf{L}$ denotes the intensity of the one pixel of the input images, ρ denotes the albedo of the object, $\mathbf{S}$ denotes the geometric relative position of the light sources and the camera, and n denotes the normal vector of the surface. The vector $\mathbf{L}$ is a single column vector and the length of the vector is the number of the input images, which must be 3 or more. The $\mathbf{S}$ denotes the geometric relative position of the light sources and the camera. The number of the row of the $\mathbf{S}$ is equal to the number of the light conditions, and the number of the column is 3.

When $\mathbf{L}$ and $\mathbf{S}$ are known, n and ρ are obtained from Eq. (1) as shown in Eq.(2) and (3).

$$n = \frac{\mathbf{S}^{-1}\mathbf{L}}{\rho} \tag{2}$$

$$\rho = |\mathbf{S}^{-1}\mathbf{L}| \tag{3}$$

III. Method

We propose a real time shape reconstruction system using a laptop computer. The object needs to exist in front of the display screen and stay still for at least 4 frames to reconstruct the image.

The last 4 images are used to reconstruct the shape. When ever the new image is captured, the last 4 images is updated by replacing the oldest image with the captured image.

A. Light Source Matrix

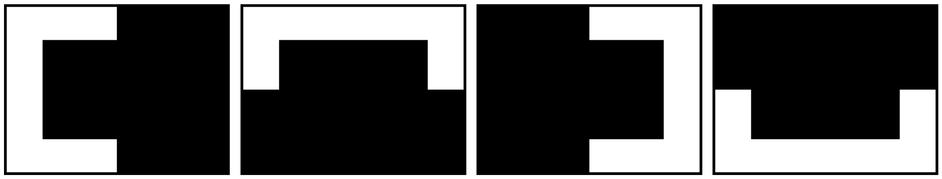
First, we display 4 different types of images on the display to give a different light conditions to the object. To make the direction of the light orthogonal, we used 4 images instead of the minimum requirement, 3. The 4 light source images and the input images in each case are shown in Fig.1 and 2. Each images are displayed sequentially. The image of the object is captured via the webcam placed on the top of the display. The capture of the webcam is synchronized with the change of the light conditions. The geometric relation between the display and the webcam is measured previously. which provides us the information of the light source matrix $\mathbf{S}$ decribed in Eq.(1). The matrix is pre-calculated once, and reused for every frame.

The previous methods[1], [2] were using SVD to obtain the light source matrix $\mathbf{S}$. They state the SVD approach sometimes confound the X component and the Y component of the normal map due to the big difference between the Z component and the others. In the proposed method, we use the geometric relationship between the camera and the light source. Each component of the normal map are obtained correctly, in every frame.

Since we use 4 images, the light source matrix $\mathbf{S}$ becomes 4 by 3 matrix. Instead of calculating the inverse matrix $\mathbf{S}^{-1}$, we calculate the pseudo-inverse matrix $\mathbf{S}^{+}$.

B. Reconstruction of the Normal Map

To obtaiin the normal map, the input image intensities are multiplied to matrix $\mathbf{S}^{+}$. This operation is done on every pixel. The intensity value of the corresponding pixels are chosen among the obtained 4 images and grouped to construct $\mathbf{L}$ which is vector of 4. From $\mathbf{S}^{+}$ and $\mathbf{L}$, we obtain the normal vector and the albedo of the surface based on Eq.(2) and (3).



(A) LEFT (B) TOP (C) RIGHT (D) BOTTOM

Fig. 1. Light Source Used to Compute the normal map

Fig. 2. Example of input images under different light conditions

(a) X component (b) Y component (c) Z component

Fig. 3. Each component of the reconstructed normal map

Fig. 4. Example of the computed depth map

The example of the reconstructed normal map is shown in Fig.3.

C. Convert Normal map to the Depth Map

Three dimensional shape is defined by the depth map. This corresponds the depth value to each pixel $(x,\ y)$. The convert from normal map to the depth map is done by iterative approach. The depth of each pixel are update based on the following equation,

$$\begin{aligned} Z_{i,j}^{k+1} &= \tfrac{1}{4}[Z_{i+1,j}^{k} + Z_{i-1,j}^{k} + Z_{i,j+1}^{k} + Z_{i,j-1}^{k} \\ &\quad -p_{i,j} + p_{i-,j} - q_{i,j} + q_{i,j-1}] \end{aligned} \tag{4}$$

where $Z_{i,j}^{k}$ expresses the depth of $(i,\ j)$ on the k-th iteration and p and q express the normal map in x and y direction. For the first frame, we start the computation on $Z = 0$ at every pixel. After the first frame, we will use the current frame depth map as the initial value. This will give us a better computed result of depth map.

IV. Experiments

The experiment is done on a laptop computer with Intel(R) Core(TM)2 Duo CPU T9300(2.50GHz) and nVidia GeForce 8600M GT. The image was obtained by the integrated webcam on the laptop and the resolution was 640x480 by 10 frame per second. The result is shown in FigIV

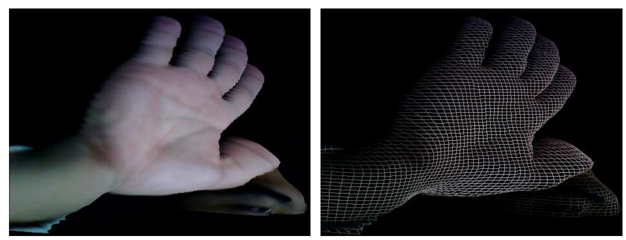


Fig. 5. Reconstructed shape of the hand with the texture and the wired version

On our implementation, the computation of the normal vector, albedo and the depth map is done by GPU. Each process has been done by shader programming since they are very suitable to compute them simultaneously. The number of the iteration for the computation of the depth map was up to 100 times. During our tests, the fps limitation is only due to the webcam hardware specification which was 10 fps.

V. Conclusion

We proposed a real time 3D shape reconstruction method based on photometric stereo. The reconstrucion was done 10 time per second and the limitation is due to the capture frame rate of the webcam. The computation speed was achieved based on the simultaneous computation of the GPU.

References

[1] V. Nozick, I. Daribo, H. Saito. “GPU-based photometric reconstruction from screen light” In Proc. 18th International Conference on Artificial Reality and Telexistence, 2008

[2] C. Hernandez, G. Vogiatzis, G. Brostow, B. Stenger, and R. Cipolla. “Non-rigid photometric stereo with colored lights.” In Proc. IEEE 11th International Conference on Computer Vision ICCV 2007, pages 18, 2007

[3] R. J. Woodham. “Photometric method for determining surface orientation from multiple images”. pages 513531, 1989