HAL
open science

GPU-based photometric reconstruction from screen light

Vincent Nozick, Daribo Ismael, Hideo Saito

To cite this version:

Vincent Nozick, Daribo Ismael, Hideo Saito. GPU-based photometric reconstruction from screen light. 8th Annual International Conference on Artificial reality and Telexistence (ICAT2008), Dec 2008, Japan. pp.242-245. hal-00733816

HAL Id: hal-00733816

https://hal.science/hal-00733816

Submitted on 19 Sep 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

GPU-based photometric reconstruction from screen light

Vincent Nozick
Institut Gaspard Monge UMR 8049, Université Paris-Est Marne-la-Vallée, France
vnozick@univ-mlv.fr

Ismaël Daribo
Signal and Image Processing Department, Telecom ParisTech, France
daribo@telecom-paristech.fr

Hideo Saito
Graduate School of Science and Technology, Keio University, Japan
saito@ozawa.ics.keio.ac.jp

Abstract

This paper address the problem of obtaining from one web camera and a computer display, a facial reconstruction of the user within online messaging applications.

In this paper we present a 3D shape recovery in real time based on the photometric information of a set of 4 images under varying illumination conditions. Our GPU implementation provides online realistic 3D face recontructions.

1. Introduction

Enjoy three-dimensional entertainment at home is intend to be one of the new promising communication services. The development of digital TV and autostereoscopic display allow to easily insert stereoscopic technologies at home. We expect in a near future that every household will be equipped with such equipments. In that way we designed a 3D reconstruction application based on cheap and accessible devices that allows users to communicate via online applications with a 3D perception of their interlocutor, as shown is Figure 1 by only means of a computer screen and a web camera.

1.1. 3D reconstruction

There exist various techniques to perform 3D reconstruction from videos. Some of them can work in real-time and most of them require several calibrated cameras. Depth from stereo methods [10] compute a disparity map from point correspondences. The visual hulls [8] method extracts the silhouette of the main object of the scene on the images from every camera. The 3D shape of this object is then approximated by the intersection of the projected silhouettes. The plane-sweep algorithm can compute a 3D reconstruction of the scene in real-time [9] using a discretisation of the scene with parallel planes.

Figure 1. 3D facial reconstruction.

Some other methods can perform a 3D reconstruction from a single camera. Optical flow methods [7] analyze the motion of the object of the scene to recover the 3D information. Finally, 3D reconstruction can be performed by Radiometric techniques. These methods require several images of the same scene under different lighting condition to extract the 3D shape of the scene. Since our method belongs to the latter family, we expose the related works on the next section.

1.2. Previous Work

In the past decades, intense interest in photometric stereo problem has produced many excellent works for establishing the theoretical part. The idea of photometric stereo, first introduced by Woodham [12], is to determine the surface

orientation at each point by varying the direction of incident illumination while holding the same point of view.

The main and difficult part is to find a way to map the RGB intensity to a normal map. To overcome this issue, experimental methods have been investigated by Christensen and Shapiro [1] and Hertzmann and Seitz [6]. They build for each material surface a look-up table with general reflectance properties. The main drawback of these methods concern the assumption that objects have homogeneous surfaces, which is not workable for complex objects. However they proposed a full segmentation into different materials.

More recently, Hernandez and al. [5] have presented work on using spatially separated red, green and blue light sources to estimate a dense depth map from a untextured non-rigid surface. By using a calibration tool, a mapping RGB intensity to normal map is provided, and thus, the depth map is obtained by integration of the normal map.

From the small baseline multi-flash camera used in [2] is possible to compute first the depth edge. In fact the flash illumination allows to measure the cast shadow width. Based on this measurement a gradient map field is provided which is integrated by using a Poisson equation.

As previously seen, the knowledge of the lighting conditions are commonly necessary. Hallinan [4] overcomes the issue of not knowing the lighting conditions by proposing a low-dimensional illumination representation of human face under arbitrary light conditions. Given a set face image, the lighting conditions are estimated by using principal components in an image basis.

Finally, for biochemistry purpose a simple system has been proposed by Filippini and al. [3] consisting in using a computer screen as a programmable light source, working with a web camera which captures the visible absorption features of samples as chemical image.

The rest of the paper is organized as follows: in Section 2, each steps of the 3D geometry recovery from input images are discussed. Section 3 describe the real time implementation on GPU. Finally the results are presented in Section 4, and conclusion are discussed in Section 5.

2. Our Approach

In this section we present in details the recovery of the 3D structure of an human face by photometric means. By using four light sources via the computer screen, as show the input image in Figure 2, the normal surface map are estimated by using an image basis issue from the lighting conditions. And then, the depth map are computed from the normal surface map integration.

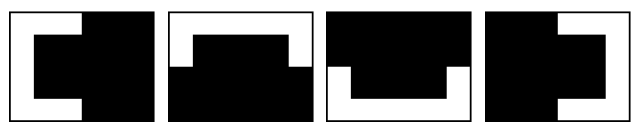
2.1. Lighting from the screen

The computer screen is used as a large programmable light source area and provides various lighting conditions

(a) Left (b) Top (c) Right (d) Bottom

Figure 2. Input images under varying illumination.

for the photometric reconstruction. The illuminated scene is captured by a camera attached on the screen and a reconstruction is performed for every new captured frame. Indeed, the new image is transfered to the system and added to the latest captured images set. To optimize the quality of the reconstruction, every image of the set should be taken under different lighting condition and the number of images should be as big as possible. However, using too much images with a dynamic scene will lead to a latency on the reconstruction process since every image should correspond to the same scene geometry. Moreover, using too much images may increase the computation time and prevent from real-time rendering. Considering these constraints, using four input images and hence four lighting conditions seems to be a good compromise. To ensure enough lighting, the light source should not be punctual and since the light direction should be roughly known, we choose to illuminate alternatively every top, right, bottom and left half part of the screen, as shown on Figure 3.



(a) Left (b) Top (c) Right (d) Bottom

Figure 3. Different lighting conditions on the computer screen.

As mentioned above, the system do not have to wait for four new views between two consecutive reconstructions. The latest captured image will update the image set by replacing the existing image under the same lighting condition. Naturally, this approach involves synchronization between the screen and the camera.

Finally, the ambient light of the scene should be reduced to the minimum to maximize the screen light contribution.

2.2. Surface normal map

Given an image of a scene, a surface normal map associates a surface normal vectors to every pixel of the image. As presented by [13], the Singular Value Decomposition (SVD) can be used to compute a surface normal map from a set of N images I_j ($j = 1...N$) of dimension $width \times height$. The N input images are converted to grey scale images and considered as one dimensional arrays. A matrix A with N rows and $width \times height$ columns is defined

such every row of A corresponds to an input image. The SVD of $AA^\top$ provides a set of eigen vectors that determines for every pixel the contribution coefficient of the input images to create the normal map. According to the big size of $AA^\top$, the computation time required for the SVD may prevent from real-time rendering. An alternative to this approach is to compute the SVD of $A^\top A$ which is a $N \times N$ matrix. This approach is much faster however the eigen vector information is common for every pixel of the image.

Since every input image is different, the SVD of $A^\top A$ will provide N orthogonal eigen vectors. The eigen vector associated to the biggest eigen value corresponds to the z-axis. The two next biggest eigen values correspond to two others orthogonal directions. Since we arranged our light system to be oriented only in vertical and horizontal directions, these two eigen values will correspond to the x and y image axis. We can identify which of the two eigen vectors corresponds to the x and y axis by comparing the provided coefficient for every input image. The x axis eigen vector will provide a high coefficient for the images associated with the left and right lighting while the y axis eigen vector will give high values for the top and bottom lighting. To check the identification, the ith component of the x, y and z eigen vectors should correspond to the light orientation of the image of the ith line of A.

(a) X component (b) Y component

Figure 4. Example of normal surface map.

Finally, for every pixel of the camera image, we compute a normal vector. The x component (respectively the y and z component) is given by the dot product of the x axis eigen vector (respectively the y and z vector) with the column of A corresponding to the current pixel as shown in Figure 4. For the depth map recovery, the normal vectors should be normalized however since the eigen vectors of $A^\top A$ are common for all the pixels, some resulting normal vectors may be null. These vectors should be detected to avoid mistakes during the depth map recovery process.

2.3. Depth map computation

The depth value $z = z_{i,j}$ of a object at the pixel position (i, j) can be determined using an iterative scheme under K iterations $k = 1..K$ using the following equation :

$$z_{i,j}^{k+1} = \frac{1}{4}[z_{i+1,j}^k + z_{i-1,j}^k + z_{i,j+1}^k + z_{i,j-1}^k - p_{i,j} + p_{i-1,j} - q_{i,j} + q_{i,j-1}] \quad (1)$$

Where p and q correspond to the normal map in x and y direction.

As suggested by [11], we choose to solve this equation using Gauss-Seidel method. Indeed, even if an iterative method does not guarantee the best accuracy, it presents the advantage to accept as an initial solution the depth map of the previous frame, which leads to a fast convergence. Moreover, Gauss-Seidel relaxation is very well suited to be implemented on GPU since every pixel can be processed simultaneously. Finally, to ensure a constant reference depth for all the reconstructions, the relaxation process is not applied on the corners of the depth map (Figure 5).

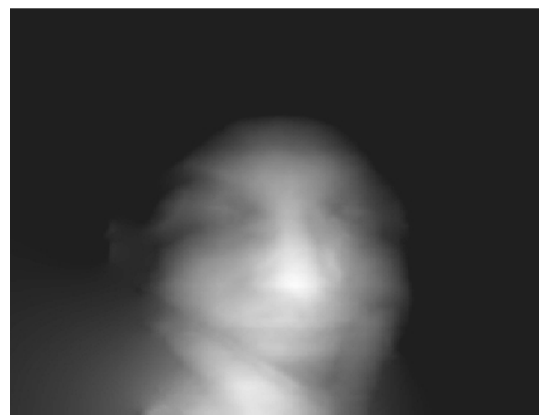


Figure 5. Depth map.

3. Implementation

For each new reconstruction, a new black and white pattern is displayed on the screen. Then, the scene is captured by the camera. The camera-screen synchronization is a critical issue to solve, especially with webcams using streaming mode. In our system, we used Video for Linux and used a read method that waits until it receives the queried image. This approach is not slower than the streaming method and contributes to an accurate synchronization.

The latest grabbed image is converted in grey scale and inserted in the matrix A. Since the camera and the screen are synchronized, we know the image lighting conditions and thus can insert this image on the corresponding line of A. Hence, this method prevents from inserting in A two images with the same light conditions. The $A^\top A$ matrix does not require a full computation to be updated. Actually, only one row and one column should be updated and since $A^\top A$ is symmetric, these row and column can be updated simultaneously.

The eigen vectors and surface normal maps are computed on the CPU and transfered to the GPU for the relaxation step. The Gauss-Seidel iterations are performed off-screen by the GPU using frame buffer object (FBO). Two textures are used alternatively to contain the depth map of the previous iteration or to be attached to the FBO for the current iteration rendering. The relaxation process uses the equation presented on section 2.3.

Finally, a dense flat mesh is projected on the screen. Every vertex depth is modified according to the depth map using a vertex program. Then the meshes are multi-textured with the four input images with a fragment program.

This method does not require any transfer of the depth map between the GPU and the main memory.

4. Results

We have implemented our system on a PC Intel core duo 1.86 GHz with an nVidia GeForce 7900 GTX. The video acquisition is performed by one USB Logitech fusion web camera connected to the computer. With a 320×240 resolution, the acquisition frame rate reaches 15 frames per second. All the computations are make within the image size of 320×240 pixels.

The GPU parallelism computation allows for the depth map recovery over 100 iterations to obtain a convergent solution with Gauss-Seidel. During our tests, the fps limitations is only due the web camera hardware limitations.

For more accuracy, the subject should be far from the camera to produce an orthographic projection. A distance trade-off is made such the subject can receive enough light from the screen. Figure 6 depicts some 3D reconstruction performed in real-time.

Figure 6. 3D geometry surface recovery: (left) input image, (right) 3D surface.

5. Conclusion

In this paper we present a real time implementation on GPU of a 3D facial reconstruction destined to home use public application by simply using a standard web camera and the computer screen. By knowing the light conditions, it's possible to reconstruct the 3D geometrical surface with only one webcam.

Thanks to our GPU implementation of the relaxation step and the multi-texturing blending, this method can reach real-time rendering.

However a limitation of this method concerns the screen light contribution that must be predominant over the ambient light.

References

[1] P. H. Christensen and L. G. Shapiro. Three-dimensional shape from color photometric stereo. *International Journal of Computer Vision*, 13(2):213–227, 1994. 2

[2] R. Feris, R. Raskar, L. Chen, K. Tan, and M. Turk. Discontinuity preserving stereo with small baseline multi-flash illumination. In *IEEE International Conference in Computer Vision (ICCV'05)*, Beijing, China, 2005. 2

[3] Filippini, D. Svensson, and I. S. P. S. Lundstrom. Computer screen as a programmable light source for visible absorption characterization of (bio)chemical assays. *Chemical Communications - Royal Society Of Chemistry*, 2:240–241, 2003. 2

[4] P. Hallinan. A low-dimensional representation of human faces for arbitrary lighting conditions. *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '94)*, pages 995–999, June 1994. 2

[5] 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 1–8, 2007. 2

[6] A. Hertzmann and S. Seitz. Shape and materials by example: a photometric stereo approach. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages I–533–I–540 vol.1, 2003. 2

[7] S. Kang, M. Uyttendaele, S. Winder, and R. Szeliski. High dynamic range video. In *In Proc. SIGGRAPH*, pages 319–325, 2003. 1

[8] M. A. Magnor. *Video-based Rendering*. A K Peters Ltd, 2005. 1

[9] V. Nozick and H. Saito. On-line free-viewpoint video : From single to multiple view rendering. *Journal of Automation and Computing (IJAC)*, 5(3):257–267, 2008. 1

[10] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proc. IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001. 1

[11] G. Schindler. Photometric stereo via computer screen lighting for real-time surface reconstruction. In *Proceedings of 3DPVT - the 4th Internation Symposium on 3D Data Processing, Visualisation and Transmission*, 2008. 3

[12] R. J. Woodham. Photometric method for determining surface orientation from multiple images. pages 513–531, 1989. 1

[13] A. Yuille, D. Snow, R. Epstein, and P. Belhumeur. Determining generative models of objects under varying illumination: Shape and albedo from multiple images using SVD and integrability. *Intl. J. of Computer Vision*, 35(3):203–222, 1999. 2