

3D Surface Reconstruction Using Polynomial Texture Mapping

Mohammed Elfaragy, Amr Rizq and Marwa Rashwan

Bibliotheca Alexandrina, P.O.Box 138, Chatby, Alexandria 21526, Egypt

Abstract. A lot of research has been conducted on 3D surface reconstruction using multiple images of fixed view point and varying lighting conditions. In this paper, a new image-based modeling technique based on Polynomial Texture Mapping (PTM) is presented. This technique allows automated reconstruction of highly detailed 3D texture-mapped models. The new technique addresses some of the shortcomings of the previous techniques, mainly in output quality and texture mapping. Moreover, it greatly enhances the perception of existing PTMs. A single PTM is used to generate a uniformly lit diffuse map, a normal map and a height map. The three maps are used to reconstruct a 3D texture-mapped surface. Given the huge archives of PTMs maintained by many institutions and museums, this technique will allow building equally large libraries of 3D models that can improve artifact perception and will aid cultural heritage research.

1 Introduction

Polynomial Texture Maps (PTM) were first introduced by Malzbender et al[1]. PTMs are generated using multiple photographs of an object, taken from the same view point, under varying lighting directions. The resulting texture maps can be viewed by means of a viewer software which enables users to dynamically change the lighting angle and intensity. This greatly enhanced the perception of archeological artifacts, especially items of very low surface relief. PTM proved to be a very useful tool for archeologists and it is being used to study and archive whole collections in various museums and institutions.

Assuming a Lambertian surface is being captured, a particular pixel in an input image set attains constant chromaticity. Luminance value changes largely for the same pixel depending on the lighting angle. The final color of the pixel $(R(u,v), G(u,v), B(u,v))$ is computed by modulating an unscaled color value $(R_n(u,v), G_n(u,v), B_n(u,v))$ by an angle-dependent luminance factor, $L(u,v)$ for each pixel:

$$\begin{aligned} R(u,v) &= L(u,v)R_n(u,v); \\ G(u,v) &= L(u,v)G_n(u,v); \\ B(u,v) &= L(u,v)B_n(u,v); \end{aligned} \tag{1}$$

Luminance dependence on light direction is modeled by the following biquadratic function per texel:

$$L(u,v;l_u,l_v) = a_0(u,v)l_u^2 + a_1(u,v)l_v^2 + a_2(u,v)l_u l_v + a_3(u,v)l_u + a_4(u,v)l_v + a_5(u,v) \quad (2)$$

Where (l_u, l_v) are projections of the normalized light vector into the local texture coordinate system (u, v) and L is the resultant surface luminance at that coordinate.

For $n+1$ input photos, the best fit at each pixel is computed using Singular Value Decomposition (SVD) to solve the system of equations for a_0 - a_5 :

$$\begin{pmatrix} l_{u0}^2 & l_{v0}^2 & l_{u0}l_{v0} & l_{u0} & l_{v0} & 1 \\ l_{u1}^2 & l_{v1}^2 & l_{u1}l_{v1} & l_{u1} & l_{v1} & 1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ l_{uN}^2 & l_{vN}^2 & l_{uN}l_{vN} & l_{uN} & l_{vN} & 1 \end{pmatrix} \begin{pmatrix} a_0 \\ a_1 \\ \vdots \\ a_5 \end{pmatrix} = \begin{pmatrix} L_0 \\ L_1 \\ \vdots \\ L_N \end{pmatrix} \quad (3)$$

The SVD is computed once for a specific arrangement of light sources and then it can be applied per pixel. A separate set of coefficients $(a_0$ - $a_5)$ is stored per pixel alongside the unscaled color value RGB. This PTM format is known as LRGB PTM and it explicitly separates and models luminance per pixel. Each pixel is stored as a nine-byte block, such that one byte is assigned for each coefficient $(a_0$ - $a_5)$ and three bytes are assigned for the RGB values. An alternate representation in which each color channel is stored directly as a biquadratic polynomial also exists. It is referred to as RGB PTM, and it is stored as eighteen bytes, six bytes for coefficients $(a_0$ - $a_5)$ for each of the three color channels. RGB PTM is more accurate than LRGB and is generally used when modeling variations of pixel color owing to other parameters besides incident light direction, such as highly reflective materials.

PTMs technique was further expanded by Mudge et al[2] by introducing PTM Object Movies (POMs). POMs are assembled from a number of individual PTM files used per inclination angle row. Single and multiple rows are possible. The POM viewer application permits examination of the object from various viewing angles while allowing user to dynamically change lighting direction.

2 Related Work

A lot of research has been conducted in the area of 3D surface reconstruction from images featuring a static camera angle with multiple lighting conditions. Most of this work is based on the Photometric Stereo (PS) method. PS was first introduced by Woodham[3]. The idea of PS is to vary the direction of the incident illumination between successive views while holding the viewing direction constant. This provides enough information to determine surface orientation at each picture element. Ikeuchi[4] expanded the technique to support specular surfaces by using a distributed light source obtained by uneven illumination of a diffusely reflecting planar surface. Later, Woodham[5] introduced a method to compute dense representations of the intrinsic curvature at each point on a visible surface, based on PS. [6] used the PS technique for 3D surface reconstruction. [7,8,9,10] further enhanced output quality by using Markov Random Fields, Jacobi's Iterative Method, Color Segmentation, Frankot-Chellappa Algorithm, respectively. While these approaches managed to significantly improve the output quality, the PS-based 3D reconstructed surfaces are still

not accurate enough to compare with other 3d photogrammetry methods that depend on images from multiple points of views. This is mainly due to the limited number of input photos and the presence of noise. Many approaches were taken to overcome difficulties caused by restrictive lighting conditions and shadowing problems. [11] proposed a method to separate the m-bounced light in the PS setup, thus removing the impact of inter-reflections for the shape recovery process. The drawback of this method is that it assumes a uniformly colored lambertian surface and requires using light sources of multiple colors. [12] formulated the PS problem to the Markov Random Field Problem and showed how to solve it by graph cut, which properly calculates the surface normal and automatically evades the interference of specular reflection. This gave accurate results but the algorithm used has a complexity of $O(N^3)$. Traditionally, PS uses a limited number of up to four images with different lighting conditions. PTMs, on the other hand, are generated from a much larger number of photos, usually, not less than 36 images per PTM. This large number of photometric data input allows for better separation of color and luminance, and nullifies shadowing problems introduced in traditional PS. This fact was used in the proposed technique to efficiently extract clean diffuse maps for reconstructed surfaces.

Finding a good estimation of the displacement map using an existing normal map is an essential part of the work presented in this paper. Depth-sensing cameras can be used to extract these maps; however, experimental results show that the random error of depth measurement ranges from a few millimeters up to about 4 cm at the maximum range of the sensor. This leads to great loss in data especially for items of very low surface relief[13]. Dmitriev and Makarov[14] introduced a technique for generating height maps from normal maps by integrating depth changes in a circular area surrounding each pixel in the normal map. MacDonald et al[15] discussed the extraction of normal maps from PTMs, which is one of the basic foundations of the work presented in this paper.

3 PTM Based 3D Modeling and Texture Mapping

The information in a PTM covers both color and form of the object and hence, it can be used to construct 3D geometry and texture maps for the object of interest. The technique described here generally works best for objects of flat nature such as coins, stone tablets and carved walls, which are the types of objects PTM is usually used to model. Expansion to arbitrary shaped objects is also possible through usage POMs. The reconstructed texture-mapped 3D model is based on three 2D maps; diffuse and normal maps for color mapping and surface details, respectively, and a displacement map which is used to shift vertices of a 3D grid to construct 3D geometry.

3.1 Diffuse Map

Diffuse maps define the main color of the surface. A good diffuse map should contain only color information without any directional light effects, inter-reflections, specular highlights or self-shadowing. The presence of any of these effects in a diffuse map will make the object respond in an incorrect way to virtual incident light, such as

casting shadows in the wrong directions or showing highlights where no direct incident light hits the object.

It is difficult to get uniformly lit diffuse maps using regular photography because this requires special lighting conditions that might not be possible for all objects of interest, especially objects in archeological sites which are lit by direct sunlight. To compensate for the absence of uniform lighting, designers usually spend hours using photo editing software to fix any part of the image that is affected by direct lighting. Usually, the results include many fake parts with color information that does not match reality. This leads to loss of information and so cannot be used for research purposes. Laser scanning and 3D photogrammetry based texture maps also suffer from these shadowing problems; this gives PTMs a clear edge here.

PTM allows extraction of super accurate diffuse maps because the luminance and chromaticity information is stored separately in PTM. In LRGB PTM format, information exists out of the box. RGB PTM format does not provide the same information directly. However, similar results can be obtained by setting:

$$(l_u, l_v) = (N_u, N_v) \quad (4)$$

Where (l_u, l_v) are projections of the normalized light vector into the local texture coordinate system (u, v) , and (N_u, N_v) is the surface normal projection into the same coordinate system (see 3.2). In this case, a different light direction is applied per texel to guarantee all texels will get the same amount of light. This gives a uniformly lit surface that is ideal for use as a diffuse map. Figure 1 shows an example of a diffuse map obtained from a PTM.



Fig. 1. (*left*) A snapshot of PTM under certain light condition (*right*) the extracted diffuse map showing uniformly lit pixels

3.2 Normal Map

A normal map is a texture map containing surface normal at each texel stored as RGB value. Normal maps are usually used to alter pixel normal to give the illusion of high-

resolution geometry details when they are mapped onto a low-resolution 3D mesh. It is a widely used technique for real time rendering, hence can be very useful when using 3D models generated from PTMs for real time applications. Additionally, normal information is useful for obtaining diffuse maps for RGB PTM format (see 3.1).

Directional lighting information for each pixel is already stored in a PTM, which makes it possible to get a very good estimate of the surface normal at that pixel. This is achieved by setting $\frac{\partial L}{\partial u} = \frac{\partial L}{\partial v} = 0$ to solve for the maximum of the biquadratic in Eq. (2). This yields lighting angle of maximum reflected luminance divided by surface normal N :

$$N = (I_u, I_v, \sqrt{1 - I_u^2 - I_v^2}) \quad (5)$$

Where

$$I_u = \frac{a_2 a_4 - 2a_1 a_3}{4a_0 a_1 - a_2^2} \quad I_v = \frac{a_2 a_3 - 2a_0 a_4}{4a_0 a_1 - a_2^2} \quad (6)$$

Normal components XYZ are represented by RGB image components, respectively. X and Y are in the range [-1, 1] while Z is in the range [0,1]. For convenient storing, all values are mapped to [0,255], the range usually used for RGB images. Figure 2 shows the surface normal extracted from a PTM file.



Fig. 2. A snapshot of PTM (*left*), and the extracted Normal map (*right*)

3.3 Height Map

Unlike normal maps, height maps (a.k.a displacement maps) are used to actually deform geometry and build a mesh of actual 3D details. Height maps are usually grayscale texture maps where pixel's white level corresponds to height (usually in Z direction) value of vertices of a 3D grid mesh. Height maps can be used either to

refine the surface of an existing 3D model or to build the model from scratch. The latter case is widely used for 3D terrain modeling. Modern GPUs provide real time surface tessellation which allows using height maps to generate models in real time.

The height maps used here are generated from normal maps obtained in the previous section. A normal at every surface point is perpendicular to height map gradient. Here, the inverse problem is what needs to be solved. Obtaining a height map from normal map requires integration. This operation is not always guaranteed to yield precise result since it is based on an estimated normal map. In addition, information about surface discontinuities is lost in normal maps. The 3D models generated are good approximations for the real model and can help to improve perception when used alongside a usual PTM. These models are also very useful for real time navigation of artifacts and architectural designs.

The algorithm used for height map generation is an iterative algorithm where each iteration improves contrast between low and high points. A low number of iterations can be used to generate a height map suitable for adding surface details to an already existing 3D model. A large number of iterations yields a map that can be applied to a grid mesh to generate the model from scratch.

The height map pixels are initiated to zero height. At each new iteration, each pixel's height is slightly modified according to the slopes of the surrounding pixels normals and their heights in the current iteration. For vertices on the same row, only X components will contribute, whereas vertices on the same column will affect height only using their Y components. For vertices on the four corners, both X and Y components will contribute to the height shifting. Contributions from all eight surrounding pixels will be averaged and added to the current height. Signs differ depending on the location of the surrounding pixel relative to the pixel being modified. Figure 3 shows a summary of surrounding pixels contributions and associated signs, where N_x and N_y are the X and Y components of the surface normal, respectively.

- N_x - N_y	+ N_y	+ N_x + N_y
- N_x	Current Pixel	+ N_x
- N_x - N_y	- N_y	+ N_x - N_y

Fig. 3. Contributions of surrounding pixels to the current pixel height when generating the height map from the normal map

Below is a pseudo code for the height map creation loop:

```

For(number of iterations set by user)
  For (all pixels of the PTM)
    CurrentPixelHeight +=
    (

```

$$\begin{aligned}
& (\text{UpperLeft.Height} - \text{UpperLeft.Nx} + \text{UpperLeft.Ny}) + \\
& (\text{Left.Height} - \text{Left.Nx}) + \\
& (\text{BottomLeft.Height} - \text{BottomLeft.Nx} - \text{BottomLeft.Ny}) + \\
& (\text{Upper.Height} + \text{Upper.Ny}) + \\
& (\text{Bottom.Height} - \text{Bottom.Ny}) + \\
& (\text{UpperRight.Height} + \text{UpperRight.Nx} + \text{UpperRight.Ny}) + \\
& (\text{Right.Height} + \text{Right.Nx}) + \\
& (\text{BottomRight.Height} + \text{BottomRight.Nx} - \text{BottomRight.Ny}) \\
&) / 8
\end{aligned}$$

The resulting pixel values are then mapped to values in the range [0,255]. Figure 4 shows the resulting height maps for different iterations numbers.

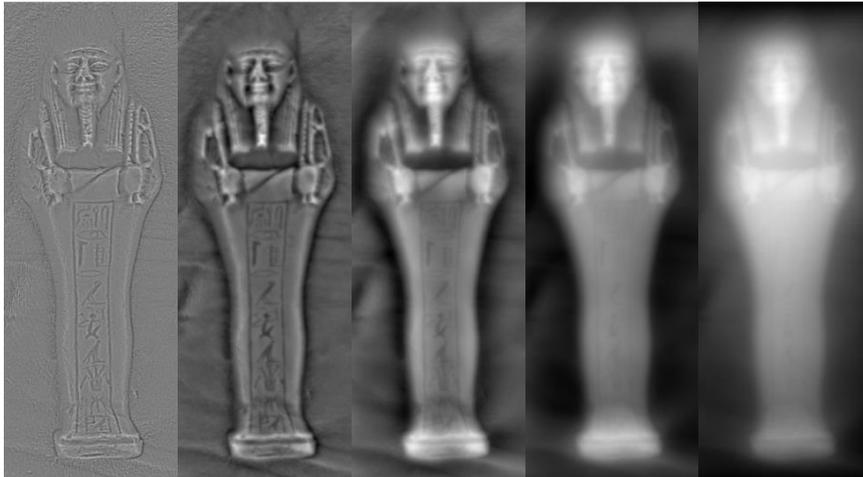


Fig. 4. Height map extracted from normal map. From left to right, results at 1, 100, 1000, 10000 and 100000 iterations

3.4 3D Surface Modeling

2D Delaunay triangulation was used to generate rectangular grids that were deformed using resulting height maps and texture-mapped using diffuse and normal maps. It was noticed that the quality of the generated models improved proportionally with number of iterations used to generate the height map. Iterations beyond 100,000 iterations had no noticeable effect. The same three maps were also used to generate models in real time using DirectX 11 tessellation features in “Unity 3D” [16] graphics engine. The models were viewed smoothly despite having a huge amount of surface details. Figures 5 and 6 show examples of the generated models.



Fig. 5. Wireframe, shaded and texture-mapped renderings of the reconstructed surface

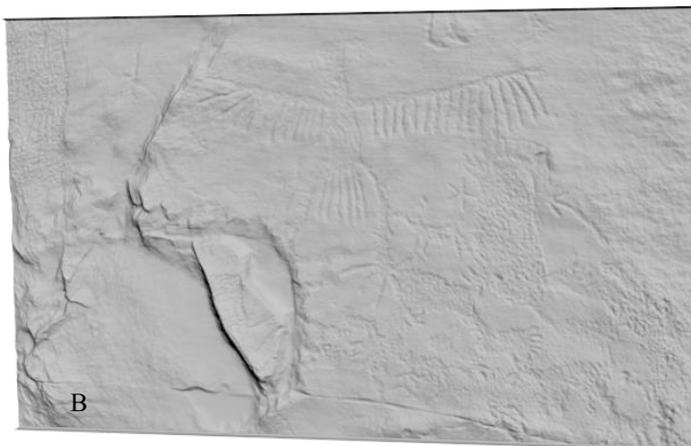




Fig. 6. (A) Original PTM, (B) Reconstructed model, (C) Model with bump and height maps

4 Conclusion and Future Work

In this paper, a new technique to reconstruct 3D models from PTMs has been introduced. The technique is fast, easy and capable of producing high quality texture mapped 3D models. Diffuse, normal and height maps are extracted from a PTM file and then used to generate a 3D texture mapped surface. User Interference is minimal and the whole process can be automated by initially setting a few parameters.

This work can be expanded and refined in many ways. In depth comparisons between models generated using this technique and laser scans of the same objects are required. These comparisons can help to determine the best settings, such as the optimal number of iterations to generate height maps, and the ideal number of photos to generate the PTM, in order to get the generated models as accurate and close to reality as possible. They would also reveal more of the strengths and weaknesses of the technique and which objects are suitable for modeling this way and which ones are not.

Another current limitation is that the generated models are only of planar nature. It is possible to generate the same set of texture maps from POMs and hence generate 3D models for non-planar objects such as cylindrical seals and other objects of interest.

5 Acknowledgements

The authors would like to thank their collaborators: Ropertos Georgiou, Bruce Zuckerman, Constantinos Sophocleous, Marilyn Lundberg and Nikos Bakirtzis. Thanks to Hewlett-Packard Labs and the Cultural Heritage Imaging Board of Directors, contributors and volunteers for supporting this work.

References

1. Malzbender, T., Gelb, D., Wolters, H.: Polynomial texture maps. SIGGRAPH '01 Proceedings of the 28th annual conference on Computer graphics and interactive techniques. New York (2001) 519-528
2. Mudge, M. , Malzbender, T., Schroer, C. , Lum, M.: New Reflection Transformation Imaging Methods for Rock Art and Multiple-Viewpoint Display. VAST'06 Proceedings of the 7th International conference on Virtual Reality, Archaeology and Intelligent Cultural Heritage. Aire-la-Ville, Switzerland (2006) 195-202
3. Woodham, R.: Photometric Stereo: A Reflectance Map Technique For Determining Surface Orientation From Image Intensity. Proceedings of SPIE 0155, Image Understanding Systems and Industrial Applications I, 136 (January 9, 1979)
4. Ikeuchi, Katsushi, "Determining Surface Orientations of Specular Surfaces by Using the Photometric Stereo Method," Pattern Analysis and Machine Intelligence, IEEE Transactions on , vol.PAMI-3, no.6, pp.661,669, Nov. 1981
5. Woodham, R.J., "Determining surface curvature with photometric stereo," Robotics and Automation, 1989. Proceedings., 1989 IEEE International Conference on , vol., no., pp.36,42 vol.1, 14-19 May 1989
6. Kyoung-Mu Lee; Kuo, C.-C.J., "Shape reconstruction from photometric stereo," Computer Vision and Pattern Recognition, 1992. Proceedings CVPR '92., 1992 IEEE Computer Society Conference on , vol., no., pp.479,484, 15-18 Jun 1992
7. Kam-Lun Tang; Chi-Keung Tang; Tien-Tsin Wong, "Dense photometric stereo using tensorial belief propagation," Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on , vol.1, no., pp.132,139 vol. 1, 20-25 June 2005
8. Ikeda, O., "Uniform converting matrix in photometric stereo shape reconstruction method," Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on , vol.1, no., pp.1,317-20 vol.1, 25-28 July 2004
9. Ikeda, O., "Shape reconstruction for color objects using segmentation and photometric stereo," Image Processing, 2004. ICIP '04. 2004 International Conference on , vol.2, no., pp.1365,1368 Vol.2, 24-27 Oct. 2004
10. Ikeda, O., "An accurate shape reconstruction from photometric stereo using four approximations of surface normal," Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on , vol.2, no., pp.220,223 Vol.2, 23-26 Aug 2004
11. Miao Liao; Xinyu Huang; Ruigang Yang, "Interreflection removal for photometric stereo by using spectrum-dependent albedo," Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on , vol., no., pp.689,696, 20-25 June 2011
12. Miyazaki, D.; Ikeuchi, K., "Photometric stereo using graph cut and M-estimation for a virtual tumulus in the presence of highlights and shadows," Computer Vision and Pattern Recognition Workshops (CVPRW), 2010 IEEE Computer Society Conference on , vol., no., pp.70,77, 13-18 June 2010
13. Khoshelham, K., & Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2), 1437-1454.
14. Dmitriev, K. & Makarov, E.: Generating displacement from normal map for use in 3D games. In ACM SIGGRAPH 2011 Talks (SIGGRAPH '11). ACM, New York, NY, USA, , Article 9 , 1 pages (2011)
15. MacDonald, L., & Robson, S.: Polynomial texture mapping and 3d representations. In Proc. ISPRS Commission V Symp. 'Close Range Image Measurement Techniques. (2010)
16. <http://unity3d.com/>