Image-based Modeling and Rendering with Geometric Proxy

Angus M.K. Siu
Department of Computer Science
City University of Hong Kong, Hong Kong
angus@cs.cityu.edu.hk

Rynson W.H. Lau
Department of CEIT
City University of Hong Kong, Hong Kong
Rynson.Lau@cityu.edu.hk

ABSTRACT
In this paper, we present an image-based method to recover a geometric proxy and generate novel views. We use an integrated modeling and rendering approach to deal with the difficulty of modeling, and reduce the sampling rate. Our system is based on two novel techniques. First, we propose the *Adaptive Mesh Segmentation* (AMS) technique for recovering geometric proxy of a scene environment. Second, we propose the *Trifocal Morphing* technique for efficient rendering with the geometric proxy, which can handle non-matched regions of the scene. Our method allows images to be sparsely captured and thus highly reduces the manual image acquisition effort as well as the data size.

Categories and Subject Descriptors: I.4.8 [**Image Processing and Computer Vision**]: Scene Analysis – *stereo, object recognition*, I.4.10 [**Image Processing and Computer Vision**]: Image Representation – *Morphological*.

General Terms: Algorithms, Experimentation, Theory.

Keywords: Image-based methods, image-based modeling, image-based rendering, 3D reconstruction, geometric proxy.

1. INTRODUCTION
Object modeling with traditional computer graphics techniques is usually expensive and difficult. It may require intensive manual work to create the geometric models. It may also be difficult to obtain the required parameters to model different material properties and lighting conditions for photo-realistic rendering.

3D reconstruction in *computer vision* tries to recovery explicit 3D geometry models from real images [Fitz98, Poll02]. Once 3D models of a scene are constructed, traditional rendering methods may be used to render novel views of the scene. In general, 3D reconstruction methods compute geometry information by matching corresponding feature points among multiple reference images. Since it is not possible to match occluded regions in the images, there may be insufficient information to reconstruct a complete explicit geometry model. In addition, most existing object recognition techniques are unable to identify and segment the exact boundaries of objects, and hence suffer from two limitations. First, they can only be used to recover individual objects or structures that contain sufficient surface texture. Second, to limit the error due to occlusion, they require very high spatial sampling rate and hence support only a small disparity range, typically around 10 ~ 30 pixels.

Image-based rendering (IBR) tries to generate photorealistic novel views through parameterizing the sampled images and re-constructing a viewer-oriented Plenoptic function [Adel91]. Plenoptic functions of various dimensions have been proposed with different assumptions and restrictions [McMi97, Gort96, Levo96, Shum99]. In general, to support translational motion, existing IBR methods require either some geometry information of the scene or a large number of reference images to be available. The problem of requiring the geometry information is that expensive hardware is needed, while the problem of requiring large number of reference images is that laboring and time consuming capturing process needs to be performed.

Although IBR research shows that explicit geometric models are not necessary for rendering image-based environments or objects, it does not mean that geometry information is no longer necessary. The geometry information, which is also known as geometric proxy [Bueh01] in IBR, is vital for reducing the spatial sampling rate [Chai00] and making IBR methods practical. Some works treat the geometric proxy recovery problem as the traditional 3D reconstruction problem or stereo matching problem. As a result, the need for geometric proxy inherits the difficulties / limitations of 3D reconstruction and stereo matching.

We note that the requirement for recovering geometric proxy is different from recovering explicit geometry models in traditional computer vision. In geometric proxy recovery, it may not be necessary to obtain the depth or 3D location of every image point. Partial, instead of complete, segmentation may be sufficient for novel view synthesis. In addition, new image-based representation schemes may be developed to handle the unmatched, occluded regions for rendering. These characteristics of geometric proxy have two major advantages. First, the problems in recovering geometry information from occluded and non-texture regions may be alleviated and geometric proxy recovery may become more practically feasible than 3D reconstruction. Second, geometric proxy may be recovered from images that are more widely sampled, significantly reducing the sampling rate and the manual image acquisition effort.

In this paper, we present an image-based method to recover the geometric proxy for rendering. Our method considers both the modeling process and the rendering process together. In the modeling process, we extend the image-based representation scheme called *Relief Occlusion-Adaptive Mesh (ROAM)*, which we developed earlier [Siu03], to store the recovered geometric proxy. We also propose an image registration technique called *Adaptive Mesh Segmentation (AMS)* for constructing the geometric proxy. This technique segments the input images into regions and acquires geometry information by matching correspondences with large disparity range. In the rendering process, we propose to use the *Trifocal Morphing* technique to synthesize arbitrary views and to handle occluded regions with smooth morphological transition.

Conference'04, Month 1–2, 2004, City, State, Country.

2. SYSTEM OVERVIEW

Figure 1 shows our image-based modeling and rendering system. We first recover the camera pose information from the sampled images using the method described in [Hart00]. We then recover the geometric proxy with our AMS technique to form ROAM. During the rendering time, we use the trifocal morphing technique to synthesize arbitrary novel views with ROAM.

Unlike 3D geometric model with texture mapping, ROAM segments an image $\boldsymbol{I}_j$ into a mixture of 3D and 2D meshes depending on whether individual regions of an image can be matched with those of the other image(s). Hence, ROAM is basically a set of hybrid 3D-2D meshes. For the matched patches $\mathbf{M}$, we estimate their depths and consider them as globally consistent 3D surfaces. For the unmatched patches $\mathbf{U}_j$, they may represent occluded regions and are treated as 2D Graphical Objects (GO) [Joan98]. They are defined separately and adaptively for each image. ROAM can be defined as follows:

$$ROAM = \{\boldsymbol{I}_j\,(\mathbf{M}, \mathbf{U}_j, f) \mid j = 1...\mathrm{N}\}, \quad f : (\mathbf{M} \cup \mathbf{U}_j) \rightarrow \mathsf{C}$$

where $\mathbf{M} \subset \Re^3$ and $\mathbf{U}_j \subset \Re^2$. f is the attribute function mapping the patches to the color space C.

3. GEOMETRIC PROXY RECOVERY

3.1 Feature Extraction and Guided Matching

To extract feature points from an input image, we first compute the interest points in each image with the Harris operator [Harr88]. By applying an appropriate threshold on the result, we may obtain a set of feature points for each image. We then attempt to establish correspondences among the images. Due to noise and occlusion, bad matches may occur. With the camera pose information, we use the epipolar constraint to guide the matching. The geometric constraint confines the search range from 2D (search window) to 1D (along an epipolar line). This significantly reduces the number of bad matches and computation time.

3.2 Triangulation and Consistency Checking

In this step, we construct matched triangular meshes from the unstructured point correspondences of the reference images. We first perform Delaunay triangulation on the unstructured, matched points in the first image to obtain a set of triangles. By comparing the topology and the pattern of a triangle in the first image with the corresponding triangles in the second and the third image, we may determine if the triangle is a matched triangle. If it is, we will attempt to match other triangles adjacent to it. The process repeats until all triangles in the set are checked. With this consistency checking, object segments are implicitly identified and some mismatched point correspondences can be removed.

3.3 Edge-Constrained Triangulation (ECT) and 3D Location Estimation

In the previous step, some of the triangles in one image may not match with those of the other images, due to errors in point matching. Here, we try to identify these regions and re-match them. By fixing the boundary vertices of the matched meshes, we connect the meshes together to form a new set of triangles. We again perform a consistency check on each of the newly formed triangles with the corresponding triangles in other images. The matched triangles are merged into the set of matched triangular meshes. The non-matched triangles are simply removed.

To extend the set of matched triangular meshes, we repeat the above mesh construction process. For successive iterations, we lower the threshold to select additional feature points that are outside the regions covered by the matched triangular meshes. We then attempt to construct matched triangular meshes on the additional feature points, provided that the newly formed meshes do not overlap with the existing matched triangular meshes. This iteration process continues until reaching the lowest threshold. In the final iteration, we keep the non-matched triangles produced and merge them to form the unmatched meshes.

To support arbitrary view synthesis, we estimate the location of each vertex of the triangular meshes in 3D space. We use the *linear triangulation* method to obtain the maximum likelihood estimate of a 3D point $\mathbf{X}$ from its 2D correspondences $\mathbf{x}^j$ in the j^{th} reference image. Let $\mathbf{P}_j$ be the 3x4 camera matrices of the j^{th} images. Then, $\mathbf{x}^j = \mathbf{P}_j\mathbf{X}$. The equations in each view can be combined into the form of $\mathbf{AX}=\mathbf{0}$, where

$$\mathbf{A} = \begin{bmatrix} x^0\mathbf{p}_0^{3\mathsf{T}} - \mathbf{p}_0^{1\mathsf{T}} \\ y^0\mathbf{p}_0^{3\mathsf{T}} - \mathbf{p}_0^{2\mathsf{T}} \\ \vdots \\ x^N\mathbf{p}_N^{3\mathsf{T}} - \mathbf{p}_N^{1\mathsf{T}} \\ y^N\mathbf{p}_N^{3\mathsf{T}} - \mathbf{p}_N^{2\mathsf{T}} \end{bmatrix} \qquad \mathbf{P} = \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} \qquad \mathbf{x}^i = (x^i, y^i, 1)^{\mathsf{T}}$$

This forms a redundant set of equations. We perform singular value decomposition on $\mathbf{A}$ and get the least square solution of $\mathbf{X}$ by finding the solution as a unit singular vector corresponding to the singular value of $\mathbf{A}$.

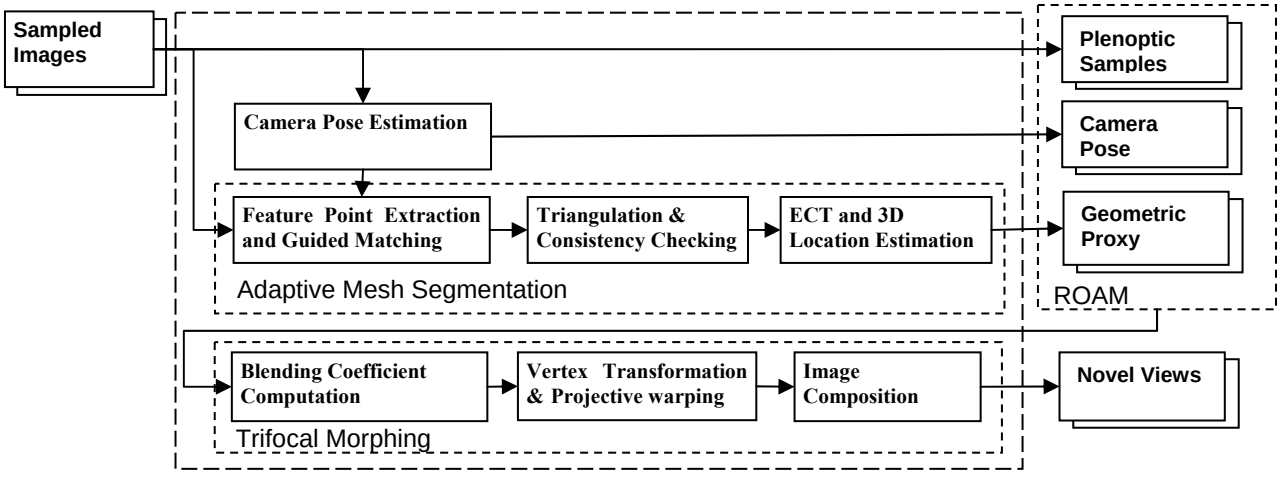

Figure 1. Our image-based modeling and rendering system.

4. RENDERING WITH ROAM

After constructing the ROAM, we may use it to generate novel views. A major problem is how to handle non-matched regions while allowing arbitrary view synthesis. If existing view interpolation methods are used to handle the non-matched regions, the viewpoint would be restricted to the line joining the camera centers of the two reference images. We propose a 3D trifocal morphing technique for rendering. Unlike traditional morphing techniques that use only 2 reference images to carry out linear interpolation, our method uses 3 images to achieve morphing in 3D space. It allows the user to have 6 degree-of-freedom of motion. Plane homography is used for texture warping to obtain perspective correct images.

4.1 Blending Coefficient Computation

To compute the pixel values of a novel view, we need to select appropriate images and texture blending coefficients to construct different parts of the view. For simplicity, we base our discussion on a 2D scenario, assuming the viewpoint and the centers of the reference cameras to be more or less located on a horizontal plane. Figure 2 shows the top view of a scene with $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ being the locations where reference images 1, 2 and 3, respectively, are taken. **d** is the camera location for generating the novel view. **p** is a feature point. We project a ray **dp** from **d** to **p**. We also project rays $\mathbf{c_1p}$, $\mathbf{c_2p}$ and $\mathbf{c_3p}$ from $\mathbf{c_1}$, $\mathbf{c_2}$ and $\mathbf{c_3}$ to **p**, respectively. We refer the angles formed between ray **dp** and each of the camera rays $\mathbf{c_1p}$, $\mathbf{c_2p}$ and $\mathbf{c_3p}$ as $\boldsymbol{\theta_1}$, $\boldsymbol{\theta_2}$ and $\boldsymbol{\theta_3}$. To minimize the *angular deviation* [Bueh01], the image with the smallest angle to one side of **dp** and that to the other side of **dp** will be selected for morphing. In the example, images 1 and 2 will be selected for **p**. The texture blending coefficients for **p** can be computed as:

$$\lambda_1 = \frac{\theta_2}{\theta_1 + \theta_2} \qquad \lambda_2 = 1 - \lambda_1 \qquad \lambda_3 = 0$$

where $\boldsymbol{\lambda_1}$, $\boldsymbol{\lambda_2}$, $\boldsymbol{\lambda_3}$ are the texture blending coefficients of images 1, 2 and 3, respectively, on **p**.

4.2 Projective Warping

To transform pixels from the references images to a novel view, we first project the 3D vertices to the output image. Based on the projected vertices, we warp each pixel from the triangular meshes of the input images. Although *Barycentric mapping* is a popular method to warp pixels of a triangular patch, it does not produce perspectively correct images. Instead, we perform perspective transformation using plane homography. Refer to Figure 3. Let $\mathbf{H_1}$ be a 4x3 matrix to transform a homogenous 2D point to the projective 3D space and $\mathbf{H_2}$ be a 3x4 matrix to transform a homogenous 3D point to 2D space. Homography **H** is given as:

$$\mathbf{x'} = \mathbf{Hx} = \mathbf{H_2 H_1 x} \tag{1}$$

$$\mathbf{H_1} = \begin{pmatrix} (\mathbf{I} - \frac{\widetilde{\mathbf{C}}\widetilde{\boldsymbol{\pi}}^{\mathbf{T}}}{\widetilde{\boldsymbol{\pi}}^{\mathbf{T}}\widetilde{\mathbf{C}} + 1})\mathbf{N} \\ -\frac{\widetilde{\boldsymbol{\pi}}^{\mathbf{T}}\mathbf{N}}{\widetilde{\boldsymbol{\pi}}^{\mathbf{T}}\widetilde{\mathbf{C}} + 1} \end{pmatrix} \tag{2}$$

where **N** is a 3x3 projective matrix from the 3x4 camera matrix of image 1, i.e., $\mathbf{P} = (\mathbf{N}^{\mathbf{T}}|\mathbf{p_4})$, and $\mathbf{p_4}$ is the 4th column of **P**. **C** is the centre of projection and $\mathbf{C}^{\mathsf{T}} = [\widetilde{\mathbf{C}}^{\mathsf{T}}, 1]$. $\boldsymbol{\pi}$ is the projective 3D plane formed and $\boldsymbol{\pi}^{\mathsf{T}} = [\widetilde{\boldsymbol{\pi}}^{\mathsf{T}}, \mathbf{1}]$. Since $\mathbf{H_2}$ is the camera matrix of image 2, we can determine **H** from Equations (1) and (2) with the estimated camera matrices. Then, we can perform projective warp for the images based on **H**.

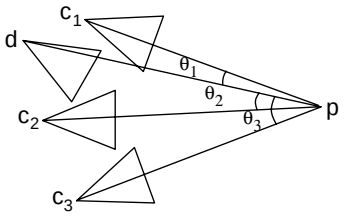

Figure 2. Selecting appropriate triangles for blending.

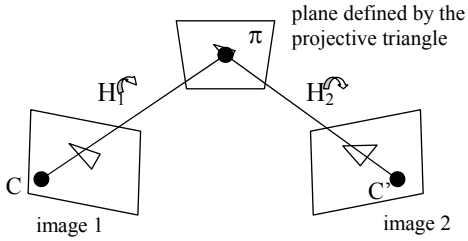

Figure 3. Implicit geometry of a triangular patch.

4.3 Image Composition

After computing the blending coefficients and warping the images, we may compose the output image. Let $l_j(x, y)$ be the pixel value of the j^{th} image at location (x, y) and λ_j be its blending coefficient. The final pixel value can be calculated as:

$$l(x,y) = \frac{\sum \lambda_j l_j(x,y)}{\sum \lambda_j} \tag{3}$$

When compositing the output image, we would draw the non-matched triangles first, as they represent occluded regions, followed by the matched triangles.

5. RESULTS AND CONCLUSION

We have conducted experiments with our IBMR system on a PC with a P4 2.2GHz CPU and 512MB RAM. We captured two sets of three reference images of two outdoor scenes at a resolution of 640x480 for the experiments. ROAM is constructed for the two sets of images with the AMS process and figure 4 shows the resulting ROAM on the source images for one of the experiments.

As shown in Table 1, the total processing time for three iterations is about one minute. As we reduce the feature point extraction threshold in later iterations, an increasing number of feature points will be extracted, leading to an increase in processing time. In general, the processing times for the two experiments are similar. As a comparison, other IBR methods, such as Lumigraph and Light-field rendering, take hours of pre-processing time for rebinning or compression.

Comparing the data size, Light-field rendering requires 2048 images to model a single lion. At a resolution of 256x256, the compressed data size is 3.4MB. [Bast99] requires 50MB of data to model a house. In our method, only three images are required, and the total data size (include geometric proxy) at a resolution of 640x480 is only 0.5MB.

Figure 5 shows the rendering performance of the trifocal morphing technique. It takes less than 0.5s to render an image of resolution 640x480, without 3D acceleration. To improve output quality, bilinear interpolation can be used in texture warping, but

the rendering time will increase to 0.7s. If the image resolution is reduced to 320x240, the rendering time drops significantly to 0.17s. Hence, the rendering performance mainly depends on image resolution. Figure 6 shows a synthetic arbitrary view.

Table 1. Performance of modeling operations.

Steps	Time (sec.)	
	Experiment 1	Experiment 2
Initial	0.7	0.6
First iteration	5.8	9.0
Second iteration	20.3	17.8
Third iteration	30.8	37.6
3D estimation	0.8	0.8
Total	58.4	65.8

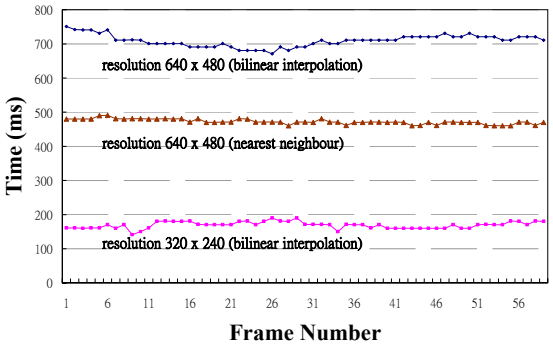

Figure 5. Rendering Performance.

Figure 6. Synthesis of an arbitrary novel view.

In conclusion, we have presented an image-based system, which integrates the modeling and rendering processes. The modeling process is based on the *Adaptive Mesh Segmentation* (AMS) technique to address the image segmentation and geometry recovery problems. The rendering process is based on the *trifocal morphing* technique to achieve smooth texture blending and handle occlusion. Our system has several advantages. First, it does not require expensive equipment or manual work to obtain geometric model. Second, the spatial sampling rate of reference images can be significantly reduced, compared with Light-field rendering. Third, it is flexible and easy for image capture because it does not require structured input or special capture gantry. In addition, unlike other image interpolation methods, the trifocal morphing technique can handle the occlusion problem while supporting arbitrary view synthesis.

ACKNOWLEDGEMENTS

The work described in this paper was partially supported by a CERG grant from the Research Grants Council of Hong Kong (RGC Reference Number: CityU 1308/03E).

REFERENCES

[Adel91] E. Adelson and J. Bergen, "Chapter 1: The Plenoptic Function and the Elements of Early Vision," *Computational Models of Visual Processing*, Landy and Movshon (Eds), MIT Press, 1991.

[Bast99] R. Bastos, K. Hoff et al., "Increased Photorealism for Interactive Architectual Walkthroughs," *Proc. ACM SI3D*, 1999.

[Bueh01] C. Buehler, M. Bosse, L. McMillan et al., "Unstructured Lumigraph Rendering," *Proc. ACM SIGGRAPH*, 2001.

[Chai00] J. Chai, X. Tong, S. Chan, and H. Shum. "Plenoptic Sampling," *Proc. ACM SIGGRAPH*, pp. 307-318, 2000.

[Joan98] G. Joans, V. Luiz, C. Bruno, and D. Lucia, *Warping and Morphing of Graphical Objects*, Morgan Kaufmann, 1998.

[Fitz98] A. Fitzgibbon, G. Cross, and A. Zisserman, "Automatic 3D Model Construction for Turn-table Sequences," *Proc. SMILE*, 1998.

[Gort96] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The Lumigraph," *Proc. ACM SIGGRAPH*, 1996.

[Harr88] C. Harris and M. Stephens, "A Combined Corner and Edge Detector," *Proc. Conf. on Alvey Vision*, pp. 147-151, 1988.

[Hart00] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, Cambridge University Press, 2000.

[Levo96] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proc. of ACM SIGGRAPH*, pp. 31-42, 1996.

[McMi97] L. McMillan, "An Image-Based Approach to Three-Dimensonal Computer Graphics," *Tech. Report 97-013*, UNC, 1997.

[Poll02] M. Pollefeys and L. van Gool, "From Images to 3D Models", *Communications of the ACM*, July 2002.

[Shum99] H. Shum, A. Kalai, and S. Seitz, "Omnivergent Stereo," *Proc. ICCV*, pp.22-29, 1999.

[Siu03] A. Siu and R. Lau, "Relief Occlusion-Adaptive Meshes for 3D Imaging," *Proc. ICME*, **2**, pp. 101-104, July 2003.

Figure 4. The relief occlusion-adaptive meshes (ROAMs) for an experiment.