Modeling and Rendering of Walkthrough Environments with Panoramic Images

Angus M.K. Siu
Department of Computer Science
City University of Hong Kong, Hong Kong

Ada S.K. Wan
Department of Computer Science
City University of Hong Kong, Hong Kong

Rynson W.H. Lau
Department of CEIT
City University of Hong Kong, Hong Kong

ABSTRACT

An important, potential application of image-based techniques is to create photo-realistic image-based environments for interactive walkthrough. However, existing image-based studies are based on different assumptions with different focuses. There is a lack of a general framework or architecture for evaluation and development of a practical image-based system. In this paper, we propose an architecture to unify different image-based methods. Based on the architecture, we propose an image-based system to support interactive walkthrough of scalable environments. In particular, we introduce the concept of angular range, which is useful for designing a scalable configuration, recovering geometric proxy as well as rendering. We also propose a new method to recover geometry information even from outdoor scenes and a new rendering method to address the problem of abrupt visual changes in a scalable environment.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation – *display algorithms*, *viewing algorithms*; I.4.5 [**Image Processing and Computer Vision**]: Reconstruction – *transformation methods*; I.4.10 [**Image Processing and Computer Vision**]: Image Representation – *Morphological*, *multidimensional*.

General Terms: Algorithms, Experimentation, Theory.

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

1. INTRODUCTION

Image-based rendering (IBR) is a technique to generate novel views from a set of reference images. It produces photo-realistic output without using a complex lighting model. A potential usage of IBR is interactive walkthrough of image-based environments, which may have many applications including 3D games and virtual tourism. However, IBR generally requires a large number of reference images or assumes that the geometry information is available. The problem of requiring a large number of images is that it will have high laboring cost and consume a significant amount of time in the image capturing process, while the problem of requiring the geometry information is that manual interaction or the use of expensive hardware is needed.

Meanwhile, computer vision research attempts to automatically create explicit object models by deriving the geometry and photogrammetry from images. Due to the correspondence matching problem and the segmentation problem, the application of these methods is usually limited to a single continuous object or structure. Thus, it cannot be generally applied to outdoor scene with multiple individual objects.

Our focus in this research is to combine these two streams of studies into a single image-based modeling and rendering (IBMR) problem. In this paper, we present an architecture for modeling and rendering of image-based environments. The system is designed to achieve real-time performance and support interactive walkthrough. The main contributions of this paper include:

- **Image-based Architecture.** We propose an architecture to unify different image-based methods. It provides a generic framework to categorize different processes and facilitate future development.
- **Scalable Configuration for Image Sampling.** We propose a scalable configuration with a novel concept called *angular range* to obtain image samples. We carry quantitative analysis and derive equations for angular range, ray direction, baseline distance and depth resolution. The equations help design the sampling configuration as well as determine the baseline distance and possible distortion in the output images.
- **Geometric Proxy Recovery.** We propose a novel method to recover geometric proxy from a panoramic image network. The method is designed for complex outdoor scenes consisting of non-rigid regions and multiple individual objects, which are difficult to handle by existing 3D reconstruction methods. It supports a large search range, i.e., lower sampling rate, allowing the source images to be widely captured.
- **Rendering.** An important feature of an image-based environment for walkthrough is scalability. To allow the environment to be scalable, the rendering system should be able to synthesize a novel view from only a local subset of images. This may lead to the problem of abrupt visual changes when switching between subsets of images in rendering. We present a rendering method to address this problem. Our method also makes use of angular range to select appropriate pixel samples as well as to determine the blending weights.

The rest of the paper is organized as follows. Section 2 discusses different image-based processes and briefly reviews existing methods. Section 3 correlates different image-based processes with a consistent image-based architecture. Section 4 presents the sampling configuration and the concept of angular range. Section 5 describes our method to estimate the camera parameters. Sections 6 and 7 describe our novel geometric proxy recovery method and our rendering method, respectively. Section 8 presents some experimental results of the proposed methods. Finally, section 9 gives a brief conclusion.

2. PREVIOUS WORK

Here, we briefly summarize the major image-based concepts and review existing works on different image-based processes.

VRST'04, November 10–12, 2004, Hong Kong.

Plenoptic function and parameterization. [1] introduces the 7D *Plenoptic function*, which describes the light intensity passing through every viewpoint, in every direction, at each time moment and for every wavelength. Plenoptic function is generally adopted by IBR for evaluating models of low-level vision. Unlike the traditional geometry approach in computer graphics, the radiant energy is described from the point of view of the observer (i.e., viewpoint oriented) rather than the point of view of the source (i.e., object oriented). Based on this approach, different dimensions of Plenoptic functions are developed with different restrictions, while the 7D Plenoptic function can be considered as the most general form of the function.

Parameterization may be understood as a re-sampling process to map pixel samples from the captured source images to the Plenoptic sample representation. Parameterization generally serves two major purposes. The first one is to remove redundant or unnecessary samples and the second is for effective rendering. In *Lightfield Rendering* [13] and *Lumigraph* [11], pixels are re-sampled to lie on a regular grid in a two-plane parameterization to form a 4D Plenoptic function. However, the viewpoint is roughly constrained to a box. *Concentric Mosaic* [18] constrains the camera motion to planar concentric circles when capturing the images, and creates concentric mosaics images by composing slit images taken at different locations along each circle. This method effectively reduces the number of reference images and forms a 3D Plenoptic function by constraining the viewer's line of sight to a 2D horizontal plane. However, this method is not scalable to large environments. *Panorama* [6] is a cylindrical projection on a fixed viewpoint to form a 2D Plenoptic function. As the viewpoint is fixed, it does not support translational motion. *Plenoptic Modeling* [15] uses a number a cylindrical projection at different viewpoints as the Plenoptic sample representation. This method is scalable and more suitable for creating image-based environments.

Geometric Proxy Recovery. The geometry information, which is used to assist in novel view synthesis, is called *geometric proxy*. It is vital for reducing the spatial sampling rate [5]. Without sufficient geometry information (number of depth levels), the manual image capturing effort and memory consumption will be too high to be practical. Geometric proxy can be derived from matching correspondences of the source images, which involves *stereo matching* and *structure and motion recovery (SMR)*. These areas have been intensively studied in computer vision.

Stereo matching [17, 22] aims at recovering a disparity / depth map from a rectified pair of planar images by minimizing a cost function. The image pair is first rectified according to the epipolar constraint, so that the corresponding points will lie on the same image scanlines. The dissimilarities of the points between two images along same scanlines are measured and encoded as a matching sequence. A cost function can then be constructed from these matching sequences. By minimizing the cost function, the correspondences and the disparity map can be computed. However, the search range of existing stereo matching algorithms is rather small, usually around 10 to 30 pixels. Further increase in the search range would cause much higher matching error.

SMR [10, 12] (or 3D reconstruction) aims at reconstructing an explicit 3D surface model. Feature points are first extracted from an image and matched with those of the other image. 3D positions of the matched points can be calculated with camera information by a triangulation procedure [9]. After that, 3D surface model may be reconstructed from these 3D points. However, there are several major problems with SMR. First, the object corners cannot always be extracted as feature points to construct an explicit surface model. Second, it is not possible to find correspondences in the occluded regions. Apparent edges and T-junctions [7] may cause mismatches in the correspondences. Third, the extracted feature points may not provide enough information about object continuity to allow complete segmentation. If there are several object structures in a scene, there may be insufficient information to completely segment them into individual surface models. Hence, although SMR may work well for recovering a single object or structure, it may not provide satisfactory results when there are several individual objects in the reference images. In addition, SMR generally requires a high image sampling rate in order to reduce the search range and hence, the errors.

Rendering. Rendering refers to the process of generating a novel view from a set of reference images, geometric proxy and camera parameters. It directly or indirectly involves the *warping* of pixels from the reference images to the novel view as well as the *blending* process to determine appropriate blending coefficients for combining pixels from different images to the pixels in the novel view. [4] generalizes existing IBR methods and suggests a set of goals for effective rendering.

There is very limited amount of work on IBR to address the scalability problem. [2] shows a scalable approach to capture images and a way to render novel views with a local subset of images. However, the problem of odd visual changes due to the change of image subsets for rendering has not been addressed.

3. AN IMAGE-BASED ARCHITECTURE

In [15], the complete flow of light in each direction of space $\mathbf{\Phi} = (\theta, \phi)$ from any point of a view $\mathbf{C} = (x, y, z)$ is represented by a 5D *Plenoptic function* $\wp(\theta, \phi, x, y, z)$. The image-based problem can be treated as *representation* and *reconstruction* of $\wp$.

We would like to consolidate this abstract mathematical description to a more concrete IBMR architecture. Our architecture as shown in Figure 1 is composed of two parts. The first part involves the offline process which aims at modeling $\wp$ from the captured images. An image $\boldsymbol{I} = (\mathbf{U}, f)$ can be described by the shape, $\mathbf{U} \subset \Re^2$ and the attribute function, f: $\mathbf{U} \rightarrow \Re^n$. The modeling processes involved include *camera parameters estimation*, *geometric proxy recovery* and *parameterization*. Camera parameter **P** defines where we capture the image samples, and can be estimated from matching corresponding image samples, which is the so-called camera calibration problem. The geometric proxy **G** is vital for reducing the sampling rate as mentioned in section 2. It may be viewed as a function g: $p \in \Re^2 \rightarrow d$, mapping each image space point p to depth information d. The geometric proxy recovery process derives the associated depth information from matching point correspondences. Parameterization may be considered as a function M: $\boldsymbol{I}_{cap} \rightarrow \boldsymbol{I}_{rep}$, which maps the pixel samples of the images $\boldsymbol{I}_{cap}$ to a specific Plenoptic sample representation $\boldsymbol{I}_{rep}$. It facilitates the rendering process and removes redundant data. If there are too many images used, compression may be included to reduce the data size.

The second part aims at reconstructing $\wp$ from the discrete data model. If the model has been compressed, it has to be de-compressed before rendering novel views. This may involve user interaction to change the view point and direction. Thus, this process is often expected to run in real-time.

Based on this architecture, we may fit in existing image-based modeling and/or rendering modules and give a global, conceptual view on the involved processes and techniques. This provides a general IBMR framework with the following properties:

- The image sampling rate depends on the geometric proxy recovery method used. [5] shows that the sampling rate is a function of depth uncertainty (or the number of depth levels), which in turn depends on the allowable search range of the geometric proxy recovery method. Thus, the allowable search range may determine the sampling rate.
- 3D reconstruction in computer vision may be considered as an approach to recover the geometric proxy. However, geometric proxy is not confined to explicit 3D models. We can simply recover as much geometry information as possible, instead of recovering explicit individual 3D models. With this relaxed requirement, only partial, instead of complete, segmentation may be needed [21]. Moreover, it is not necessary to match regions that are occluded or without sufficient texture. Thus, it becomes more practically feasible to use it to construct outdoor scenes for virtual walkthrough.

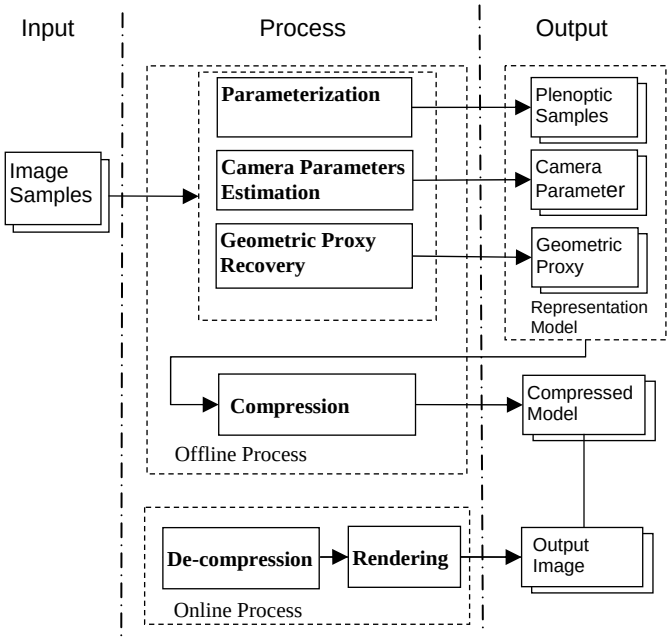


Figure 1. An image-based modeling & rendering architecture.

4. SCALABLE CONFIGURATION

As omni-directional images inherit the advantage of wide field-of-view (FOV), we use a set of panoramic images to model the image-based environment. We use single-optical center panoramic images instead of multi-optical center omni-vergent images [18] because they are more scalable and easier to obtain.

4.1 Sampling and Depth Resolution

Prior to capturing the reference images, we need to determine the sampling interval and camera locations (or configurations). [5] has derived the minimum sampling curve for Lightfiled rendering. It describes the relationship of the number of depth levels, sampling interval along baseline direction and output resolution. A problem on the recovered depth resolution is that a single pair of binocular panoramic images yields very biased reconstruction as mentioned in [18]. Thus, we need to investigate an image configuration with multiple images, which guarantees the depth resolution in all directions. We approach the problem by introducing a novel concept called *angular range*. With this concept, we show how we guarantee the depth resolution with multiple images.

4.2 Angular Range

The term search range is used to describe the region for searching a corresponding pixel in an image. The search range depends on the image resolution. Here, we introduce a more generic term, *angular range*, to describe the search range for correspondences, especially for omni-direction images.

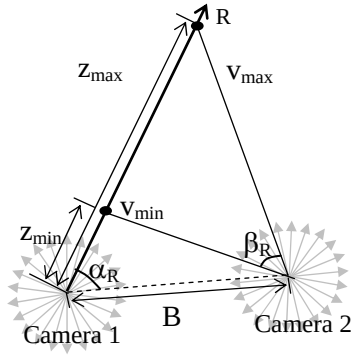


Figure 2. The angular range, β_R, for ray R.

Figure 2 shows two panoramic images captured by cameras 1 and 2. Consider ray *R* projected from image 1 with given minimum depth z_{min} and maximum depth z_{max}. z_{min} is the distance of the closest object from the camera. The point correspondence in image 2 should be located within an *angular range* β_R. This angular range is related to *B*, the baseline of the two cameras, and α_R, the angle between ray *R* and the baseline:

$$\tan\beta_R = \frac{B\sin\alpha_R\left(\frac{1}{z_{min}} - \frac{1}{z_{max}}\right)}{1 - B\cos\alpha_R\left(\frac{1}{z_{min}} + \frac{1}{z_{max}}\right) + \frac{B^2}{z_{min}z_{max}}} \quad (1)$$

If $z_{max} \to \infty$, Equation (1) can be simplified to:

$$\beta_R = \tan^{-1}\left[\frac{\sin\alpha_R}{\frac{z_{min}}{B} - \cos\alpha_R}\right] \quad (2)$$

For a panoramic image with *s* pixels wide, the search range, or the number of depth levels n, can be calculated as:

$$n = s * \beta_R / 2\pi \quad (3)$$

Based on the above quantitative analysis, we have derived the equation to compute the angular range, which have the following practical applications:

- **Determining the baseline distance:** Given the allowable search range of a matching algorithm and the depth range of the objects, the maximum baseline distance *B* (sampling interval) for capturing images can be determined.
- **Design for the required depth resolution with N images:**

  From Equations (1) and (2), we observe that β_R drops to zero when α_R approaches to zero (near the epipolar direction). Equation (3) shows that the number of depth levels n tends to

zero when β_R is zero. This means that a single image pair cannot provide sufficient depth information in some particular directions. We have to make use of multiple images instead.

Figure 3 plots Equation (2), with β_R against α_R. The term z_{min}/B generally skews the sinusoidal function. When z_{min} is much larger than B (i.e., objects in the environment are far from the capturing position), β_R is close to a sine function. We would observe that $|\beta_R|$ is comparatively large when ($\pi/3 < \alpha_R < 2\pi/3$) and ($4\pi/3 < \alpha_R < 5\pi/3$), which is the grey region in Figure 3. It corresponds to the viewing range as shown in Figure 4(a) in a single pair of images. By using a group of three images as shown in Figure 4(b), the rays in all directions can achieve a specific depth resolution. In our system, we use one redundant set of images as shown in Figure 4(c) to deal with the case of mismatches and occlusion. It results in a general and scalable image network as in Figure 5.

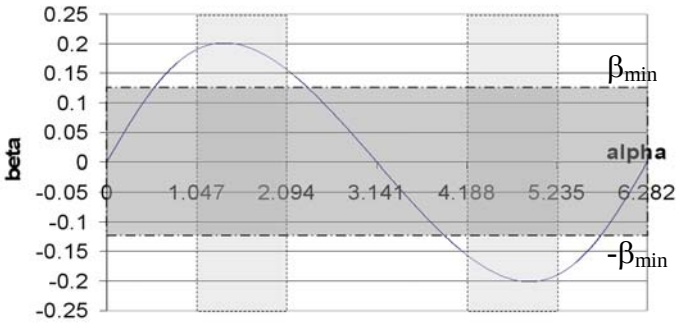


Figure 3. Effect of alpha, α_R, on the angular range, β_R.

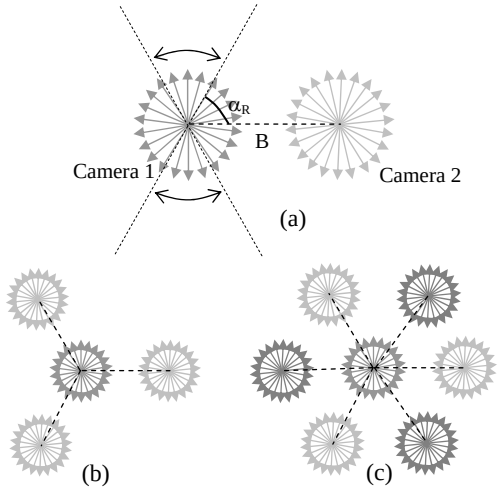


Figure 4. Camera configurations.

5. CAMERA PARAMETERS ESTIMATION

Without using specific equipment, multiple groups of planar images are first captured at different optical centers (*nodes*). Images obtained from the same node (group) are stitched together to form a panoramic image. Referring to Figure 5, we need to estimate a consistent set of camera parameters for the panoramic image network. The nodes are first triangulated and the *adjacency* between each pair of nodes is established. To save computational cost, we only match point correspondences between each pair of adjacent images. Three adjacent nodes form a triangle, called *self-loop*, which is used for consistency checking. With this configuration, we may obtain a consistent set of camera parameters using a method proposed in [24].

An alternative approach in IBMR is to interpolate novel views without camera pose information (i.e., with implicit geometry), such as Joint View Triangulation [14]. However, the trade-off is that it will be difficult to integrate other 3D objects into the environment. Thus, we estimate the camera parameters in our system to enhance the capability for developing applications that require 3D object integrations, such as 3D games.

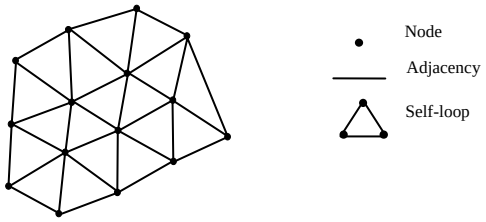


Figure 5. An image network.

6. GEOMETRIC PROXY RECOVERY

6.1 Overview

Geometric proxy recovery is one of the difficult processes in IBMR. With 3D reconstruction methods, feature points are first extracted and then matched by comparing the pixel intensity. Based on the correspondences, 3D points can be estimated and explicit 3D surface model can be reconstructed. However, this approach only works well for a single continuous object / structure, but not for a scene with multiple objects.

Here, we address the problem of geometric proxy recovery of a scene from a new perspective. First, we extract regions with sufficient frequency changes for matching, instead of object corners / features. After that, we carry out multi-resolution phase-based matching, instead of traditional intensity-based matching. To improve the robustness, we perform the matching across multiple images and formulate the array of cost-functions into a 3D maximum flow problem. By solving the maximum flow problem, a disparity map can be obtained, which is then transformed into an image-based representation, called *Relief Occlusion-Adaptive Mesh* (ROAM), instead of explicit 3D surface model. Details on each part of this method are discussed next.

6.2 Non-uniform Region Extraction

A major problem of traditional *feature-based* matching methods [16, 23] is that it is difficult to extract all corner points of every object robustly. If the among of feature / corner points are insufficient, we cannot reconstruct the object surface correctly. It is also difficult to completely segment different objects based on the sparse set of object features. On the other hand, stereo matching methods [17, 22], which are *area-based*, attempt to match all pixels. However, their results are usually very noisy and the search range is usually limited to 10 to 30 pixels only.

To deal with these problems, we have adopted a quasi-dense matching approach. We aim at obtaining the highest number of points from the non-uniform regions for matching. These non-uniform regions refer to regions with sufficient frequency changes and can be identified by the methods such as [14]. This approach has several advantages. First, the corner extraction problem is bypassed. Second, the completeness of this quasi-dense matching approach is much higher than sparse point matching. Third, this

method skips matching uniform regions. On the other hand, uniform regions are difficult to match and tend to increase the matching error. However, they usually do not create visual artifacts in the output images. Hence, ignoring these regions in the matching process will improve the matching accuracy and efficiency but would not create much artifact in the output.

6.3 Correspondence Matching

To increase the number of matching points and to improve the matching accuracy, we use multi-resolution matching with wavelet transform. We perform phase-based, instead of intensity-based, matching. This method works well in frequency domain, providing a generic way to match in a coarse-to-fine manner. First, we decompose image $\boldsymbol{I}$ into a pyramid representation. Let $\boldsymbol{\Phi}^j$ and $\boldsymbol{\Psi}^j$ be the matrices of scaling functions and wavelet functions, respectively. Second, we decompose the scaling function at level j into wavelets by the synthesis filters $[\mathbf{P}^j \mid \mathbf{Q}^j]$ as follows:

$$[\boldsymbol{\Phi}^{j-1}(x) \mid \boldsymbol{\Psi}^{j-1}(x)\ ] = \boldsymbol{\Phi}^j\,[\mathbf{P}^j \mid \mathbf{Q}^j] \qquad (4)$$

By setting $\boldsymbol{I} = \boldsymbol{\Phi}^N$, a sequence of wavelets can be obtained by applying the synthesis filters as shown in Equation (4).

Another problem in matching correspondences is occlusion. If only a single image pair is used, some regions may only appear in one image but not the other. In addition, as mentioned in section 4, the recovered depth resolution from a single image pair would be very low near the epipolar direction. To deal with this problem, we use multiple images. Except for the uniform regions, we establish a disparity function with N images on every pixel of image $\boldsymbol{I_0}$ to estimate the corresponding disparity value, $\delta = 1/z$, where the depth z is the distance between the corresponding 3D point $\mathbf{X}$ and the camera center. Given a point p in $\boldsymbol{I_0}$, we may construct an attribute function $f^k(p, \delta)$ with the k^{th} image $\boldsymbol{I}^k$. $f^k(p, \delta)$ represents the intensity values along the epipolar curve as a function of δ. A correct δ represents the location on the surface of an object. Thus the luminance value from $f^k(p, \delta)$ of different views should be very similar. Then, we can estimate the correct δ by comparing the similarity of $f^k(p, \delta)$ for a specific δ. As we use phase-based matching with multi-resolution analysis, we use the wavelet coefficients, instead of intensity values, to construct the attribute function. We define the cost function $SM(p, \delta)$ as the sum of the variances of the wavelet coefficients $w(p_k, \delta)$ as follows:

$$SM(p,\delta) = \frac{1}{m+1}\left[\sigma_v(p,\delta) + \sum \alpha_l \sigma_w(p,\delta)\right] \qquad (5)$$

where σ_v and σ_w are the standard deviation of the scaling and wavelet coefficients, respectively. The cost function considers m levels of coefficients. α_l is the weight for the coefficients at level l. Lower levels correspond to larger regions and thus with a larger weight α_l.

6.4 Global 2D Optimization

Although guided matching with multiple images can determine depth information with high precision, local mismatches and noise can cause poor output. To deal with this, we impose a coherence constraint to optimize the computation of disparity. To perform global optimization, we formulate the depth recovery problem as a maximum flow problem, which determines the minimum-cut 2D surface for a cubic data set. Referring to Equation (5), the matching cost $SM(p, \delta)$ is a function of point p and disparity δ. As $p = (u, v)$, where u, v are the coordinates of $\boldsymbol{I_0}$, the matching cost $SM(u, v, \delta)$ can be represented as a 3D array as shown in Figure 6. We may use the algorithms in [17] to solve the maximum-flow problem to obtain a 2D set of disparity values. The coherence of the surface can be controlled by appropriate smoothness factor t_s.

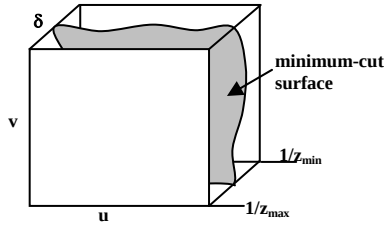


Figure 6. Formulating the depth recovery problem as a maximum flow problem.

6.5 ROAM Formation

After recovering the disparity map, we need to provide an image-based representation for the depth information among multiple views. The image-based representation should be able to separate the matched regions from the unmatched regions while preserving the topological correctness in geometric space. Although the 2D continuous mesh used in image morphing may be an effective image-based representation scheme, it enforces a continuous motion for the whole image [3]. Thus, it is not suitable to represent a scene with different individual objects as they are likely to have different motions. Alternatively, the image-based representation can be in the form of multiple textures on an explicit geometry model, like view-dependent texture mapping (VDTM) [8]. However, this method requires complete identification and segmentation of individual objects, which is very difficult to be done automatically.

To deal with the representation problem, we apply the *Relief Occlusion-Adaptive Mesh* (ROAM) [19] here. ROAM can be considered as a set of hybrid 3D-2D meshes. This method partitions an image into matched and unmatched patches. A matched patch corresponds to a globally consistent object surface at geometric space, while an unmatched patch corresponds to a 2D Graphical Object (GO). They are defined separately for each image. This representation allows objects to have different motions while keeping unmatched regions for rendering.

Here, we create a ROAM from the recovered disparity map. Initially, we subdivide the first image into regular patches of 8x8 pixels each. The 3D positions of the patches are estimated from the recovered disparity map. Based on the estimated 3D positions, we project the patches onto multiple views. The consistency of each patch can be validated by the topological constraints in a way similar to [20]. This validation process considers whether a patch is flipped or intersected with other patches. It removes the flipped or intersected patches and outputs a set of matched patches, which correspond to a consistence set of object surfaces, in 3D geometric space, among different views. For the unmatched regions in all the other views, they are broken into another set of unmatched patches by edge-constrained Delaunay triangulation. The unmatched patches may correspond to the occluded regions which are also saved for occlusion handling in rendering time.

7. RENDERING

7.1 Scalability

For our method to support scalable environments, it must be able to generate each output image with only a local subset of images.

To determine which subset of images to use, we define a circular viewing scope S_v of radius r_v ,which is centered at the viewer's position, as shown in Figure 7. All the nodes within S_v will be used for rendering. The selection of r_v is generally based on the average baseline distance, viewer's moving speed, and the amount of available memory. In the example shown in Figure 7, we assume a client machine to have only a small amount of memory and set r_v to cover only around seven nodes.

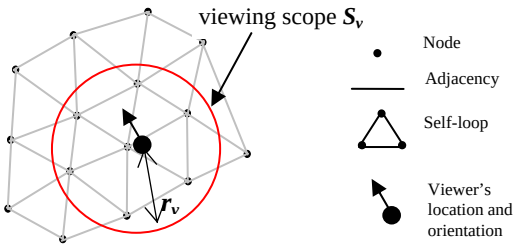


Figure 7. An image network.

7.2 Rendering Overview

In our rendering method, we first determine which reference images would be selected to compute each pixel of the output image. A blending map is also computed to specify the corresponding blending fields of a reference image. Based on this map, we warp pixels from the appropriate nodes to construct the output image. Details of the rendering method are discussed below.

7.3 Selecting Reference Nodes

The simplest way to render the output is to warp pixels from all images and then compose all of them. However, this may introduce excessive image blur if the pixel depth is uncertain. Thus, for each pixel in the output image, we only warp pixels from two images for interpolation. To select two appropriate images for warping, the angular range gives a good guideline.

In Figure 8, a ray **R** is projected from a pixel in the desired view **D**. We need to compute the pixel intensity from a subset of views $\mathbf{C}_k$. Given the minimum and maximum object depths z_{min} and z_{max}, respectively, the corresponding pixels in views $\mathbf{C}_k$ should occur within the angular range β_k. We can see that $\mathbf{C}_1$ has a smaller angular range than $\mathbf{C}_3$ (i.e., $\beta_1 < \beta_3$) while $\mathbf{C}_2$ has a smaller angular range than $\mathbf{C}_4$ (i.e., $\beta_2 < \beta_4$). Assuming that the depth information is uncertain, the maximum distortion by warping a pixel from a view with a smaller angular range should be smaller than that from a view with a larger angular range. This means that warping from $\mathbf{C}_1$ or $\mathbf{C}_2$ has smaller potential distortion than from $\mathbf{C}_3$ or $\mathbf{C}_4$, respectively. Since **R** divides the set of nodes into two groups, one on the left of **R** and the other on the right, we simply choose the view with the minimum angular range on each side. Hence, in Figure 8, $\mathbf{C}_1$ and $\mathbf{C}_2$ would be chosen.

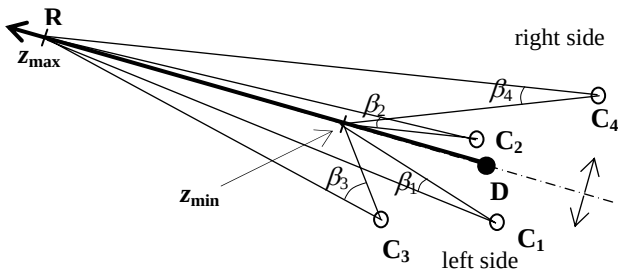


Figure 8. Angular range of different nodes to a ray.

7.4 Computation of Blending Fields

After selecting the appropriate nodes, we then determine the corresponding *blending field* for each pixel. This blending field describes how the corresponding nodes are weighted to compute a given pixel. In [4], *angular difference* between the desired ray and the reference ray on the surface proxy is used to determine the blending field. Although this provides a good estimation of the desired ray, it relies on the accuracy of the depth information. If the depth information is incorrect or absent, the blending weight would be incorrect.

We use *angular range* instead of *angular difference* to calculate the blending field. We assign a higher weight to smaller angular range. As angular range is independent of depth information, the blending field can be calculated correctly even through the depth information is incorrect. The blending weight $w_{ang}(i)$ for a pixel in image i is computed as follows:

$$w_{ang}(i) = \frac{K_{ang}}{\beta_i}, \quad \text{where} \quad K_{ang} = \frac{1}{\sum_j 1/\beta_j} \tag{6}$$

7.5 Smooth Transition and Image Composition

As the user navigates in the image-based environment, different subsets of images will be used for rendering. Normally, some regions of an image may not be matched. These regions may change suddenly, resulting in temporal aliasing. This aliasing occurs in two situations: in geometric space and in image space.

Temporal aliasing under geometric space occurs when a reference image located near the boundary of the viewing scope. In this case, when a user moves forward, the reference image may fall out of the viewing scope and the blending weight for that image would drop to zero suddenly. Then, the output image no longer reference to that image and a portion of the output image changes suddenly.

Temporal aliasing under image space means that across the output image, the blending weight changes suddenly. This problem commonly appears in images obtained from mosaicing. When multiple images are composed to form a single image, sudden change may appear at image boundaries, as a result of different intensity gradients of different images and the difficulties in matching all textured regions.

To address the aliasing problem in geometric space, we add a location coefficient w_{loc} in the blending weight calculation as:

$$w_{loc}(i) = 1 - \left(\frac{d_i}{r_v}\right)^2 \tag{7}$$

where d_i is the distance between the i^{th} reference node and the viewer's position. The coefficient makes sure that when an image moves towards the boundary of the viewing scope, the blending coefficient is gradually reduced to zero.

To address the aliasing problem in the image space, we apply the *Gaussian filter* $\boldsymbol{G}$ to the blending coefficients to smooth the blending of texture. Finally, the blending field B can be obtain as:

$$B = \boldsymbol{G} \otimes w \quad \textbf{where} \quad w(i) = w_{ang}(i) \cdot w_{loc}(i) \tag{8}$$

8. RESULTS AND DISCUSSIONS

We have implemented the proposed image-based system and conducted some experiments to test its performance. The experiments were carried out on a PC with a P4 2.4GHz CPU, 1G RAM, and a GeForce FX5900 graphics card. The image-based

environment is composed of 15 panoramic images, captured around 300mm from one another with 256 depth levels used.

First, we decompose image I_0 into different resolution levels. At each level, we extract non-uniform regions for matching. Figure 9 shows the extracted feature map at different levels. It also indicates the area percentages covered by the extracted non-uniform regions, which we refer to as the *covering percentage*. The union of the features at the three levels covers a total of around 85% of the area for matching, with 53.3%, 23.1% and 8.6% from levels 2, 1 and 0, respectively. Although the extracted features for matching at higher levels can produce higher depth resolutions, the area covered by the features is only 53.3% at level 2. With the multi-resolution approach, the covering percentage of non-uniform regions can be increased to 85%.

Using the method mentioned in section 6.3, we formulate the cost function into a maximum flow problem to compute the disparity map. Figure 10 shows the resulting disparity map after the global optimization with different smoothness factors. If the smoothness factor is set to zero, it is equivalent to the case that every pixel is considered independent of its nearby pixels and the disparity map becomes very noisy. After considering the nearby pixels by the global optimization (smoothness factor = 15), much of the noise is removed successfully. We then construct the ROAM based on the recovered disparity map. Two examples are shown in Figure 11.

Table I shows the computational cost of the modeling process. It takes less than half of an hour to complete the whole process. Although the resolution for each of the 15 images is as high as 9000x950, it takes only around ten minutes for the cost function formation, which is much more efficient than existing dense matching methods. For the mesh formation, it uses around 15 minutes to check for the topological constraints. As a comparison, other IBR methods, such as Light-field rendering or Lumigraph, usually require a large amount of reference images and may take hours of pre-processing time.

Table I. Computational Costs.

Geometric Proxy Recovery	(minutes)
Feature Extraction	0.02
Cost Function Formation	9.99
Global Optimization	1.12
Mesh Formation	14.36
Total	**25.49**

Based on the ROAM, we may render novel views in real-time and allow the user to walkthrough interactively. Figure 12 compares the frame rates with different numbers of polygons and at different output resolutions. The renderer can achieve a real-time performance of 30 frames per second even though 227,125 triangles are involved, where the patches are formed at a very fine scale. When doubling the scale of the patches, the triangle number is decreased to 55,950 and the performance is increased to over 50 frames per second. On the other hand, if the output resolution is reduced from 512x512 to 256x256, the rendering performance is more or less the same. The results indicate that the rendering performance mainly depends on the triangle number. Comparing with other IBR methods, such as *Plenoptic stitching*, which performance is only around 5 to 10 frames per second, our method is much faster. This is because our method only uses a small amount of images for rendering.

Table II shows the data size. The whole environment is only about 12MB of data which is much smaller than other existing methods, such as Lightfield rendering and Plenoptic stitching, which require several hundreds of MBs. The significant reduction in model size is due to the high matching performance of our image registration method, which recovers the geometric proxy with high depth resolution. This high depth resolution significantly reduces the sampling rate and hence the data size.

A video demo of the method proposed here can be found in: www.cs.cityu.edu.hk/~rynson/projects/ibmr/ibmr.html. We can see significant relative object motion from the demo. This shows that the recovered geometric proxy contain sufficiently rich depth information to allow significant relative motion to be observed.

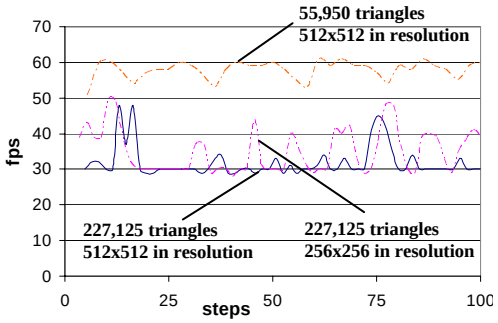


Figure 12. Rendering Performance.

Table II. Data Size of the image-based environment.

Node Number	15
Compressed Image Size (MB)	9
Geometric Proxy Size (MB)	3
Total Data Size (MB)	**12**

9. CONCLUSIONS

In this paper, we have presented an image-based architecture for modeling and rendering of image-based environments for interactive walkthrough. First, we propose a generic architecture to consolidate the mathematical model of IBR. Second, the concept of angular range is introduced and quantitative analysis is performed. Based on the results from the quantitative analysis on angular range, we may determine the sampling configuration and sampling rate of an environment. Third, based on the proposed image-based architecture, we have developed the corresponding modeling and rendering methods for creating the walkthrough environment solely from the captured images.

For the modeling method, we present the integrated use of wavelets as well as global optimization to recover geometric proxy with high depth resolution. For the rendering method, we show how to minimize the distortion and image blur by using the angular range for reference image selection and composition. Experimental results show that our modeling method is able to match most regions in the images and successfully remove the noise. The resulting data size is also much smaller than that of existing methods. Another advantage of our rendering method is that it is scalable and supports 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

[1] 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.

[2] D. Aliaga and I. Carlbom, "Plenoptic Stitching: A Scalable Method for Reconstructing 3D Interactive Walkthroughs," *Proc. ACM SIGGRAPH*, pp. 443-450, 2001.

[3] Y. Altunbasak and A. Tekalp, "Occlusion-Adaptive, Content-Based Mesh Design and Forward Tracking," *IEEE Trans. on Image Processing*, **6**(9), 1997.

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

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

[6] E. Chen, "QuickTime VR: An Image-based Approach to Virtual Environment Navigation," *Proc. ACM SIGGRAPH*, 1995.

[7] K. Cornelis, M. Pollefeys, and L. van Gool, "Tracking Based Structure and Motion Recovery for Augmented Video Productions," *Proc. of ACM VRST*, pp. 17-24, Nov. 2001.

[8] P. Debevec, G. Borshukov, and Y. Yu. "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping," *Proc. of EG Rendering Workshop*, 1998.

[9] O. Faugeras, *Three-Dimensional Computer Vision: A Geometric Viewpoint*, MIT Press, 1993.

[10] O. Faugeras, L. Robert, S. Laveau, G. Csurka, C. Zeller, C. Gauclin, and I. Zoghlami, "3-D Reconstruction of Urban Scenes from Image Sequences," *CVIU*, **69**(3):292-309, 1998.

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

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

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

[14] M. Lhuillier and L. Quan, "Image Interpolation by Joint View Triangulation," Proc. of CVPR, 2, pp. 139-145, 1999.

[15] L. McMillan and G. Bishop, "Plenoptic Modelling: An Image-based Rendering System," *Proc. ACM SIGGRAPH*, pp. 39-46, 1995.

[16] W. Niem and R. Bushmann, "Automatic Modelling of 3D Natural Objects from Multiple Views," *Image Processing for Broadcast and Video Production*, Springer-Verlag, 1994.

[17] S. Roy and I. Cox, "A Maximum-flow Formulation of the N-Camera Stereo Correspondence Problem," *Proc. ICCV*, 1998.

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

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

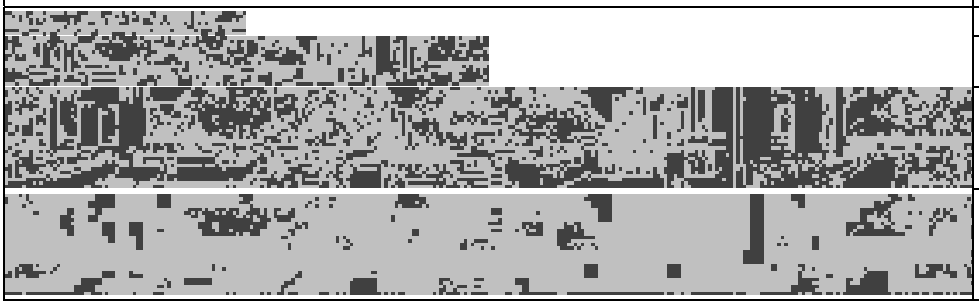
[20] A. Siu and R.W.H. Lau, "Image Registration for Image-based Rendering," *IEEE Trans. on Image Processing* (to appear).

[21] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision*, PWS, 1998.

[22] J. Sun, H. Shum, and N. Zheng, "Stereo Matching using Belief Propagation," *IEEE Trans. on PAMI*, 2003.

[23] P. Torr, A Fitzgibbon, and A. Zisserman, "Maintaining Multiple Motion Model Hypotheses over Many Views to Recover Matching and Structure," *Proc. of ICCV*, p.727-732, 1998.

[24] A. Wan, A. Siu, and R.W.H. Lau, "Recovering Camera Pose from Omni-directional Images," *Proc. BMVC* (to appear), 2004.



Level	Covering Percentage (%)
Level 0	**71.6%**
Level 1	**58.9%**
Level 2	**53.3%**
Union of three levels	**85.0%** **(8.6% - level 0** **23.1% - level 1** **53.3% - level 2)**

Figure 9. Extracted feature map at different resolution levels.

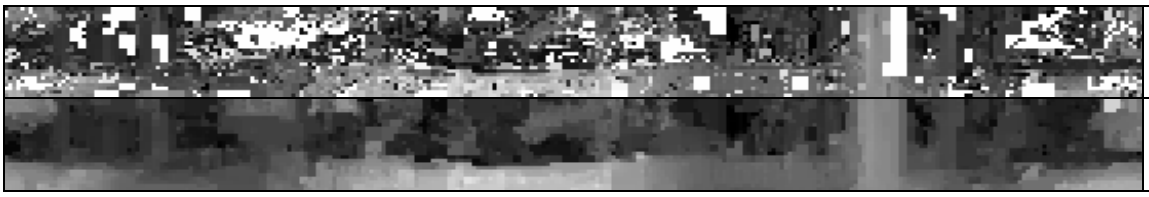
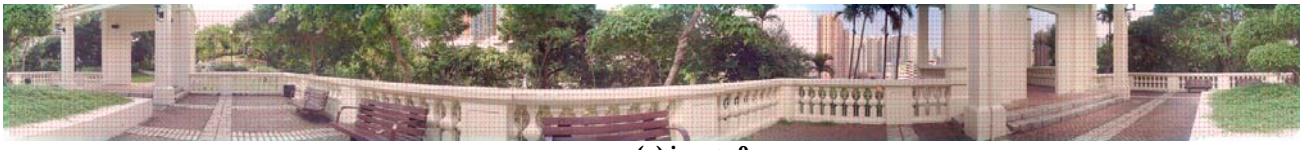
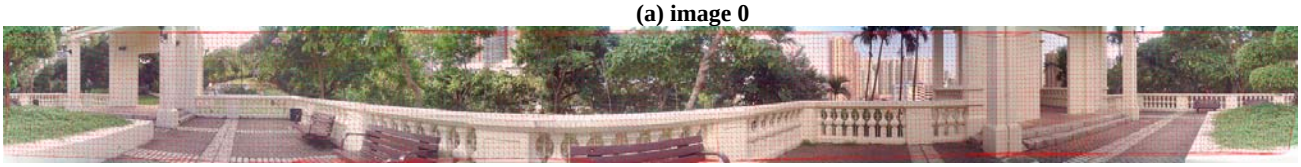


Figure 10. Disparity map at different smoothness.



(a) image 0



(b) image 5

Figure 11. ROAMs produced for some of the reference images.