Image Registration for Image-Based Rendering

Angus M. K. Siu and Rynson W. H. Lau

***Abstract*—Image-based rendering (IBR) has received much attention in recent years for its ability to synthesize photo-realistic novel views. To support translational motion, existing IBR methods either require a large amount of reference images or assume that some geometric information is available. However, rendering with a large amount of images is very expensive in terms of image acquisition, data storage, and memory costs. As IBR accepts various kinds of geometric proxy, we may use image registration techniques, such as stereo matching and structure and motion recognition, to obtain geometric information to help reduce the number of images required. Unfortunately, existing image registration techniques only support a small search range and require closely sampled reference images. This results in a high spatial sampling rate, making IBR impractical for use in scalable walkthrough environments.Our primary objective of this project is to develop an image registration technique that would recover the geometric proxy for IBR while, at the same time, reducing the number of reference images required. In this paper, we analyze the roles and requirements of an image registration technique for reducing the spatial sampling rate. Based on these requirements, we present a novel image registration technique to automatically recover the geometric proxy from reference images. With the distinguishing feature of supporting a large search range, the new method can accurately identify correspondences even though the reference images may only be sparsely sampled. This can significantly reduce the acquisition effort, the model size, and the memory cost.**

***Index Terms*—Image-based rendering (IBR), image matching, image registration, object recognition.**

I. Introduction

As IMAGE-based rendering (IBR) can generate photo-realistic novel views from reference images, it has received much attention in recent years. It is generally regarded as a powerful alternative to traditional geometry-based rendering in computer graphics. The focus of IBR studies has been extended from the earlier restricted view synthesis, as in *QuickTime VR* [9], to the recent scalable walkthrough environments, as in *Plenoptic Stitching* [3].

In order to support translational movement, most existing IBR methods may use a large number of reference images to capture all potential views. This approach, however, does not scale well because the number of images required, i.e., the spatial sampling rate, is a polynomial function to the size of the scene. For example, although feature tracking and optimization algorithms are used in [3], the sampling rate is as high as 1 600 images for a 49-m capturing path (i.e., one image for every 3 cm). Our objective of this project is to develop a method that can effectively reduce the spatial sampling rate to a state that it would be truly practical to model large scenes.

In [8], the minimum sampling curve shows that the spatial sampling rate can be effectively reduced if geometric information is available. A method that is often used to automatically extract intrinsic geometric information from reference images is *image registration*. It is basically an image processing operation to identify correspondences among a number of images. If the spatial sampling rate is be reduced, the image registration process will need to search a larger range for correspondences. This, however, introduces several challenges. First, the point matching error increases rapidly with the increase in the search range. It is necessary to control the error through some means. Second, when the reference images are taken further apart from each other, the change in image content due to perspective projection becomes significant and can no longer be approximated by simple translational movement as in block-based video compression methods such as MPEG. More accurate methods are needed to estimate the transformation of image contents. Third, a large search range will lead to a dramatic increase in computational cost. Depending on the matching algorithm, the matching time may be $O(n^2)$, where n is the search range. In video compression, the search range is only around 16 pixels. However, in IBR, it can be larger than 100 pixels. This means that existing image registration techniques are not suitable for IBR applications.

In this paper, we present a novel image registration method with a large search range. The outline of this paper is as follows. In Section II, we review existing IBR methods and the image registration techniques used. In Section III, we discuss the tasks involved in the image registration process in IBR and how these tasks are related to the spatial sampling rate. In Section IV, we present our image registration method in detail. In Section V, we show and evaluate some experimental results of the proposed method. Finally, in Section VI, we briefly conclude the paper and discuss possible future work.

II. Literature Review

Many IBR methods have been proposed to construct different types of Plenoptic functions [1], with image registration techniques being used in various stages of the method, such as image compression, disparity map computation, and feature tracking. In this section, we review representative IBR methods with an emphasis on the roles of the image registration process.

Manuscript received March 31, 2003; revised March 3, 2004. This work was supported in part by a CERG grant from the Research Grants of Hong Kong (RGC Reference Number: CityU 1308/03E) and in part by a SRG grant from City University of Hong Kong (Project Number: 7001465). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Reiner Eschbach.

A. M. K. Siu is with the Department of Computer Science, City University of Hong Kong, Hong Kong, SAR (e-mail: angus@cs.cityu.edu.hk).

R. W. H. Lau is with the Department of Computer Engineering and Information Technology, City University of Hong Kong, Hong Kong, SAR (e-mail: Rynson.Lau@cityu.edu.hk).

Digital Object Identifier 10.1109/TIP.2004.840690

Light field rendering [19] uses a large number of reference images to render novel views. It does not require any geometric information. The reference images are captured with cameras which centers are aligned regularly on a two-dimensional (2-D) plane. All light rays are indexed with an *st*-plane and a *uv*-plane, and, hence, the viewpoints are constrained to a bounding box. As compression is applied to reduce the size of the data model, an image registration process is needed to estimate the motion vectors of image blocks.

The parameterization of *Lumigraph* [12] is similar to light field rendering, but image registration is used for three-dimensional (3-D) shape approximation. A *rebinning* method is proposed to resample the reference images to the views of the virtual cameras located on a regular grid. This additional step allows the method to accept input images taken from arbitrary locations. Unlike light field rendering, geometric information is used in this method. A 3-D shape approximation process extracts the geometric information from the images, which is then used by the rebinning process to compute new views and by the compression process to compute the motion vectors.

Layered depth image (LDI) [33] and the LDI tree [7] make use of the warping equation [24] to transfer image points from the reference images to any arbitrary view taken at any viewpoint. However, they assume that the disparity map is already available. Unless expensive hardware equipment such as a laser rangefinder [23] is available, the disparity map would have to be derived from stereo matching, which is basically an image registration technique. Hence, the spatial sampling rate of this method needs to be high in order to minimize the matching error and computational cost.

By restricting the viewing orientation of the synthesized views to lie on a horizontal plane, *concentric mosaic* [31] constrains the camera motion to planar concentric circles when capturing the images. This reduces the number of reference images required. The method creates concentric mosaics images by composing slit images taken at different locations along each circle and forms a 3-D plenoptic function by constraining the viewing orientation to a 2-D horizontal plane. To reduce vertical distortion, depth correction is performed in concentric mosaic using a depth map. This depth map is computed by an image registration process [32].

QuickTime VR [9] is an IBR method that does not involve image registration. It combines a set of overlapping images taken from a fixed location but at different viewing orientations to form a cylindrical panoramic image. During the rendering stage, novel views of arbitrary orientations can be synthesized from the panoramic image. This method is simple and efficient, but the user's viewpoint is fixed.

Plenoptic stitching [3] uses a new sampling and reconstruction method particularly designed for interactive walkthrough applications. It captures the scene with an omnidirectional camera [25] and parameterizes light rays by their intersections with the observer plane and the ruled surface to create the plenoptic function. Image registration is involved in the feature tracking process. The tracked features are triangulated, serving as the references for warping line segments of the reconstructed image.

There are other popular IBR methods requiring different amount of inputs [11], [15], [17]. *Unstructured lumigraph* [4] attempts to generalize these methods by accepting varying amount of inputs (either a small number of reference images with geometric information or a large amount of reference images). It assumes the availability of the camera pose and geometric proxy. Although image registration is not directly used in this method, it is used to compute the camera pose and geometric proxy.

III. Image Registration in IBR

From the literature review presented in the last section, we may summarize that there are generally three applications of image registration in IBR: *compression, geometric proxy recovery*, and *camera pose estimation*. They are described as follows.

Compression: Both light field rendering and Lumigraph employ compression to reduce the model size. However, compression cannot reduce the spatial sampling rate. Hence, the acquisition effort of both methods is still high.

By considering the interframe redundancy, image registration may be used in motion prediction to estimate the motion vectors for compression. On the other hand, if the camera motion information and geometric proxy are already available or can be derived from other processes, we may make use of the information to compute the motion vectors more directly.

Geometric Proxy Recovery: Geometric proxy refers to the geometric information that can be used to reconstruct novel images. The disparity map of LDI, the depth map of concentric mosaic, the triangulated features of Plenoptic stitching and the 3-D surface model are different kinds of geometric proxy. As shown in [8], the use of geometric proxy can help reduce both the spatial sampling rate and the distortion problem. Different kinds of geometric proxy can be obtained using different image registration methods, such as stereo matching, and structure and motion recovery (SMR).

One type of geometric proxy, the disparity map, may be derived automatically by stereo matching [29], [35]. Images are first rectified according to the epipolar constraint so that corresponding points will lie on same image scanlines of a stereo pair. Points in one image are then matched along the same scanlines of another image. The dissimilarities of the points between the two images are measured. Correspondences are encoded in a matching sequence and a cost function can be constructed according to the matching sequences. By minimizing the cost function, the correspondences and disparity map can be obtained. However, the search range of existing stereo matching algorithms is rather small, usually less than 20 pixels. Further increases in the search range would cause a higher matching error.

The 3-D surface model is another kind of geometric proxy that may be obtained by SMR [14], [27]. By tracking feature points in different images, 3-D positions of these points can be calculated with camera pose information. The 3-D surface model may then be reconstructed from these 3-D points. However, there are several major problems with SMR. First, object corners are not always

included in the extracted 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 [10] 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. Because of these problems, SMR requires a high sampling rate of reference images to minimize the errors due to occlusion. This implies that the search range of SMR has to be small. In addition, SMR may work well with a single object or structure, but it may not provide satisfactory results when there are several individual objects in a single view.

Plenoptic stitching, on the other hand, triangulates the tracked features to form the geometric proxy. It extracts a set of features (e.g., points and corners) [30] and uses the Tomasi algorithm [37] to track them. The tracked features are then triangulated to produce the geometry information of the corresponding radial lines for warping. As this approach triangulates the features to form a standard mesh model, it assumes continuous motion across the whole image and fails to represent object discontinuity. As in stereo matching, this approach only supports a small search range in order to reduce the error rate and the effects of occlusion and object discontinuity.

Since existing image registration methods for geometric proxy recovery can only support a small search range, the spatial sampling rate of these methods becomes very high. In order to allow a larger search range, we need to develop new methods that would reduce matching errors.

Camera Pose Estimation: Camera pose information refers to a set of parameters for mapping 3-D points in the scene to pixel locations in a reference image. It includes two sets of parameters, *extrinsic parameters*, which define rotation and translation of camera pose, and *intrinsic parameters*, which define the mapping of 3-D points from the camera coordinate frame to the image coordinates.

Camera pose serves different purposes in IBR. In geometric proxy recovery, camera pose information provides the epipolar constraint in reducing the search range for matching correspondences from a 2-D region to a one-dimensional (1-D) line. In compression, camera position and motion can be used with geometric proxy to compute the motion vectors. In rendering, camera pose information is indispensable for novel view synthesis.

Image registration can be used to estimate the camera pose. If the camera lens is fixed, the intrinsic parameters will not change. We only need to estimate them once before acquiring the reference images. For extrinsic parameters, they change according to the location and orientation of the camera. It is necessary to find out a small number of correspondences among the images in order to compute the extrinsic parameters for each image. In [38], methods were suggested to compute both intrinsic and extrinsic parameters. Since the number of correspondences required is small and the accuracy of the matching needs not be high, the camera pose estimation process can work properly even if the images are taken at distant positions.

Aiming at reducing the spatial sampling rate, our focus of this paper is on geometric proxy recovery. In summary, we would like to restate two points. First, in order to effectively reduce the spatial sampling rate, a new image registration method able to support a large search range is needed for geometric proxy recovery. We have developed such a method and the details will be presented in the next section.

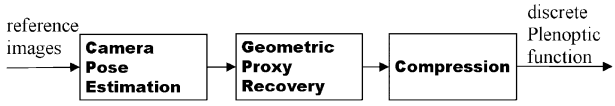


Fig. 1. Discrete plenoptic function construction pipeline.

Second, in a generic Plenoptic function, such as the unstructured lumigraph, the inputs may include reference images, camera pose and geometric proxy. With an appropriate image registration technique, both camera pose and geometric proxy can be derived from the reference images. Hence, the reference images become the only necessary input. In addition, the date size of the Plenoptic function can be further reduced by compression. In fact, while camera pose estimation provides information for geometric proxy recovery, both camera pose and geometric proxy are useful in computing the motion vectors for compression. As a result, camera pose estimation, geometric proxy recovery and compression can be integrated in a systematic discrete Plenoptic function construction pipeline as shown in Fig. 1.

IV. Image Registration for Geometric Proxy Recovery

In this section, we present our image registration technique for geometric proxy recovery. By supporting a large search range (50–200 pixels) for matching correspondences, the new method can effectively reduce the spatial sampling rate in IBR. In this section, we first give a brief overview of our method and then describe each of the operations in detail.

Our image registration technique is based on an iterative approach to construct triangular meshes for modeling the scene, starting from the most reliable correspondences to the less reliable ones. Since the most reliable correspondences can be obtained from matching feature points in the textured regions, our algorithm first applies the Harris operator [13] on the first image to extract a set of feature points (see Section IV-B). We start with a higher threshold for the operator to extract the initial set of most reliable feature points. The set of feature points are then matched with those in the second image by computing the zero-mean normalized cross correlation (ZNCC) scores [18] (see Section IV-C). The matched correspondences are checked against the third image by transferring them with the trifocal sensor [34] (see Section IV-A). This produces a set of reliable correspondences, $\mathbf{S} = \{\mathbf{S}_1, \mathbf{S}_2, \mathbf{S}_3\}$, among the three reference images, where $\mathbf{S}_i$ is the set of matched feature points in image i.

We triangulate the set of matched feature points $\mathbf{S}_1$ found in the first image by Delaunay triangulation to obtain an initial set of triangles (see Section IV-D). Based on the same triangulation topology, we match the set of triangles in the first image against those in images 2 and 3. After checking for topological consistency and pattern similarity, a set of corresponding triangular meshes, $\mathbf{R} = \{\mathbf{R}_1, \mathbf{R}_2, \mathbf{R}_3\}$, among

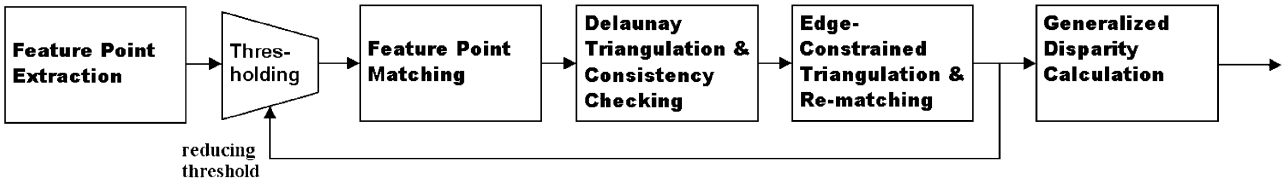


Fig. 2. Image registration pipeline for geometric proxy recovery.

the three images are produced, where $\mathbf{R}_i$ is the set of matched triangular meshes in image i. The matched triangular meshes of each image $\mathbf{R}_i$ are then joined together through edge-constrained triangulation (ECT) to form a new set of unmatched triangles (see Section IV-E). These unmatched triangles of the first image are again checked for topological consistency and pattern similarity with those of the other images. The matched triangles are added to the matched triangular meshes $\mathbf{R}_i$. Since the resulting $\mathbf{R}_i$ is constructed from a set of high threshold feature points, it represents a set of reliably matched triangular meshes.

By applying the above image registration process $\boldsymbol{G}$ on the set of reference images $\boldsymbol{I} = \{\boldsymbol{I}_1, \boldsymbol{I}_2, \boldsymbol{I}_3\}$, we obtain $\mathbf{R} = \boldsymbol{G}(\boldsymbol{I})$. The size of $\mathbf{R}$ can be extended outward through additional iterations of the image registration process, $\mathbf{R}^j = \boldsymbol{G}(\boldsymbol{I}, \mathbf{R}^{j-1})$, where $j \geq 1, \mathbf{R}^j$ is the set of matched triangular meshes obtained in the jth iteration of $\boldsymbol{G}$, and $\mathbf{R}^0 = \emptyset$. For each successive iteration, we lower the feature point selection threshold of the Harris operator to select additional feature points that are outside the regions covered by the matched triangular meshes $\mathbf{R}^{j-1}$ obtained from the previous iteration. We then attempt to construct a new set of matched triangular meshes $\mathbf{R}^j$ based on the additional feature points. This iterative process will continue until we have reached the lowest feature selection threshold. Finally, we compute the disparity values for all the vertices of the matched triangular meshes (see Section IV-F).

Fig. 2 summarizes our image registration pipeline for geometric proxy recovery. Our method first constructs matched triangular meshes on the "more reliable" feature points and gradually extends them to the "less reliable" feature points. We describe each of the operations in detail in the following subsections.

A. Preliminary Process—Determining the Trifocal Tensor

Prior to recovering the geometric proxy, we need to determine the intrinsic geometric relationship of the reference images, such as the trifocal tensor [34] that correlates three views. If the camera pose information is available, the trifocal tensor can be easily determined. Otherwise, we may perform weak calibration to determine the correspondences of the reference images for computing the trifocal tensor. By applying Carlosson–Weinshall duality [5], the camera matrix for three reference images can be recovered from six or more matched image points up to a projective stratum. In [14], a detailed algorithm for computing the trifocal tensor automatically from a set of correspondences is discussed. Our experiments show that the accuracy of the trifocal tensor can be improved by refining the position of correspondences through subpixel matching before computing the tensor. The trifocal tensor can also be computed using some software tools, such as the *Projective Vision Toolbox* [28].

B. Feature Point Extraction

The image registration method starts with feature point extraction. We compute the interest points in each image with the Harris operator [13]. Consider the Hessian matrix

$$\boldsymbol{M}(x,y) = \begin{bmatrix} \frac{\partial^2 \boldsymbol{I}}{\partial x \partial x} & \frac{\partial^2 \boldsymbol{I}}{\partial x \partial y} \\ \frac{\partial^2 \boldsymbol{I}}{\partial y \partial x} & \frac{\partial^2 \boldsymbol{I}}{\partial y \partial y} \end{bmatrix}$$

where $\partial \boldsymbol{I}/\partial x$ and $\partial \boldsymbol{I}/\partial y$ are the image gradients in the x and y directions, respectively. The result of the Harris operator is

$$\boldsymbol{H}(x,y) = \boldsymbol{D}et(\boldsymbol{M}(x,y)) - k \cdot \boldsymbol{T}race(\boldsymbol{M}(x,y))$$

where $k = 0.04$. The Hessian matrix is computed for each image point over a D × D neighborhood. In our experiments, D was set to 5. By applying an appropriate threshold on the result of the Harris operator, we can obtain a set of feature points for subsequence iterations. Note that to reduce the effect of noise on feature extraction, we may apply a Gaussian filter on the gradient images as we are performing the Harris operator. In our experiments, we used a 5 × 5 Gaussian filter for smoothing.

C. Feature Point Matching

With a set of extracted feature points in each reference image, we attempt to establish correspondences among the images. Since our method is to support a large search range, it may give rise to a large number of bad matches due to occlusion and noise. In order to reduce the number of bad matches, we apply the geometric constraint of trifocal tensor to guide the matching. This will confine the search region from a 2-D to a 1-D space, and significantly reduce the probability of bad matches and computation time. In addition, unlike most of the other image registration methods which are based on globally dense matching (e.g., [27] and [39]), we select only a small number of reliably matched feature points for mesh construction and for consistency checking. This is because globally dense correspondences are seldom achieved in practice due to the existence of nontextured regions, and the computational cost of dense matching is extremely high as most IBR applications use a large number of reference images.

The detail of our three view matching algorithm is as follows:

1) Two-View Correspondences: First, we find the matched feature points between images 1 and 2. Let $\boldsymbol{T} = [\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]$ be the trifocal tensor of the three reference images and $\mathbf{u}_i$ and v_i be the left and right null vectors of $\mathbf{T}_i$, respectively. Then, the epipoles $\mathbf{e}'$ and $\mathbf{e}''$ on images 2 and 3 are

$$\mathbf{e}'^{\mathrm{T}}[\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3] = \mathbf{0} \quad \text{and} \quad \mathbf{e}''^{\mathrm{T}}[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3] = 0.$$

Let $\mathbf{x}$ and $\mathbf{x}'$ be a match point pair in images 1 and 2, respectively. The fundamental matrix F_{21} (i.e., $\mathbf{x}'^{\mathrm{T}}\mathrm{F}_{21}\mathbf{x} = \mathbf{0}$) can be obtained from the trifocal tensor $\boldsymbol{T}$ by

$$\mathrm{F}_{21} = [\mathbf{e}']_X[\mathbf{T}_1, \mathbf{T}_2, \mathbf{T}_3]\mathbf{e}''.$$

The corresponding point $\mathbf{x}'$ of $\mathbf{x}$ can be matched along the epipolar line using the ZNCC measure. The epipolar line $\mathbf{l}'$ on image 2 can be calculated with $\mathbf{l}' = \mathbf{F}_{21}\mathbf{x}$. The $\mathrm{ZNCC_k}(ij, pq)$ measure for the two $(2k+1) * (2k+1)$ windows centered at pixels (i, j) and (p, q) is shown in the equation at the bottom of the page, where $\mathbf{m_{ij}}$ is the luminancy of pixel (i, j) and

$$\bar{m}_{ij} = \frac{1}{(2k+1)^2} \sum_{-k \le dx, dy \le k} m_{i+dx, j+dy}.$$

This correlation measure has the advantage of invariant to linear luminance changes.

2) Three-View Correspondences: After the matched points between images 1 and 2 are obtained, they are checked against the points in image 3 by transferring them to image 3 with the trifocal tensor. We can compute line $\mathbf{l}'$, which passes through $\mathbf{x}'$ and is perpendicular to $\mathbf{l}'_{\mathrm{e}} = \mathbf{F}_{21}\mathbf{x}$. If $\mathbf{l}'_{\mathrm{e}} = (\mathrm{l}_1, \mathrm{l}_2, \mathrm{l}_3)$, and $\mathbf{x}' = (\mathbf{x}_1, \mathbf{x}_2, 1)^{\mathsf{T}}$, then $\mathbf{l}' = (\mathrm{l}_2, -\mathrm{l}_1, -\mathbf{x}_1\mathrm{l}_2 + \mathbf{x}_2\mathrm{l}_1)^{\mathrm{T}}$. The transferred point is

$$\mathbf{x}''^{\mathbf{k}} = \mathbf{x}^{\mathbf{i}}\mathbf{l}'_{\mathbf{j}}\mathbf{T}_{\mathbf{i}}^{\mathbf{jk}}.$$

With the above algorithm, we can obtain a set of matched points among the three images for triangulation.

D. Delaunay Triangulation and Consistency Checking

Although we have now obtained a set of correspondences, these correspondences do not provide information about object continuity nor occlusion. In addition, reliable matching of a feature point pair depends on a set of surrounding pixels. However, these surrounding pixels sometimes cause mismatches. The objectives of this step are to identify object segments and to ensure their consistency among the reference images with topological constraints and pattern similarity checking. Such consistency checking may also help remove mismatches from the previous step.

Here, we first perform Delaunay triangulation on the unstructured correspondences and obtain a set of preliminary triangles for the first image. We then compare the topology and pattern similarity of each triangle in the first image with those of the corresponding triangles in the second and the third images. If both topology and pattern of the same triangle in the three images agree, we would consider it as a matched triangle and attempt to match other triangles adjacent to it. This process repeats until all triangles in the set are checked.

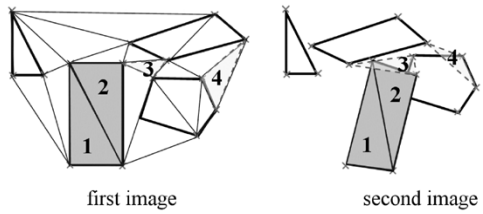


Fig. 3. Some examples of topological inconsistency between two images.

Topological Constraints: There are two constraints for topology matching.

1) The new triangle should not intersect with any existing matched triangular meshes. For example, the first image in Fig. 3 shows its triangulation. Triangles 1 and 2 are checked and accepted. When checking triangle 3, the corresponding triangle in the second image is intersected with matched triangles 1 and 2. Since triangle 3 may represent occluded region, it is rejected.
2) The orientation of the corresponding triangles in different images should not be reversed. For example, triangle 4 in the two images of Fig. 3 has opposite orientations and is thus rejected. The checking can be done by comparing the signs of the determinants of the area matrices. The determinant of triangle Δ with the vertices $\{\mathrm{p_1(a_x, a_y)}, \mathrm{p_2(b_x, b_y)}, \mathrm{p_3(c_x, c_y)}\}$ is computed as

$$\mathrm{Det}(\Delta) = \begin{vmatrix} a_x & a_y & 1 \\ b_x & b_y & 1 \\ c_x & c_y & 1 \end{vmatrix}.$$

Pattern Matching and Plane Homography: Another implication of allowing a large search range (or large camera motion) in our method is that relative object motion can become significant, affecting the pattern matching process. In existing pattern-matching techniques that are used for motion estimation in video compression [21], [22], [26], the search range and camera motion between successive frames are assumed to be small (16 pixels), and, hence, object motion can be approximated by simple translational motion in image domain in the matching process. However, if we are to support large search range and camera motion in IBR, object motion can no longer be approximated by simple translational motion. Instead, projective transformation should be considered. Unfortunately, the computation of the plane homography of a triangular patch for projective transformation in the image domain is not straightforward due to insufficient degrees of freedom.

A plane homography $\mathbf{H}_{12}$ is a linear transformation of homogeneous coordinates from image 1 to image 2. It is induced by a world plane, such that $\lambda\mathbf{x}' = \mathbf{H}_{12}\mathbf{x}$, where $\mathbf{H}_{12}$ is a 3×3 matrix, λ is a constant, $\mathbf{x}$ and $\mathbf{x}'$ are homogenous 2-D points in images 1 and 2, respectively. With the plane homography, we

$$\mathrm{ZNCC}_k(ij, pq) = \frac{\sum_{-k \le dx, dy \le k} (m_{i+dx, j+dy} - \bar{m}_{ij})(m_{p+dx, q+dy} - \bar{m}_{pq})}{\sqrt{\sum_{-k \le dx, dy \le k} (m_{i+dx, j+dy} - \bar{m}_{ij})^2} \sqrt{\sum_{-k \le dx, dy \le k} (m_{p+dx, q+dy} - \bar{m}_{pq})^2}}$$

may now perform projective transformation of a triangle from image 1 to image 2.

Since the homography has $9-1=8$ degrees of freedom, four matches $(\mathbf{u_i}, \mathbf{u_i'})$ from the two images are required to compute it. Hence, the shape of the plane to determine the homography needs to be quadrangle (with four vertices). This will involve expensive computations in solving an 8×8 matrix. Since triangles are more suitable for representing image objects, we propose here a method to compute the plane homography from triangles by virtue of intrinsic geometric information. It relaxes the restriction of plane homography calculation and determines the homography with a much lower computational cost as it does not need to solve the 8×8 matrix. Referring to Fig. 4, since a 3-D triangle may define a plane in the projective domain, the plane homography can be obtained even though the shape of the patch is triangular. Let $\mathbf{C}$ and $\mathbf{C}'$ be the centers of projection of images 1 and 2, respectively. Let $\mathbf{H}_1$ be a 4×3 transformation matrix to transform a homogenous 2-D point in image 1 into the projective 3-D space and $\mathbf{H}_2$ be a 3×4 transformation matrix to transform a homogenous 3-D point into a 2-D point in image 2. The plane homography $\mathbf{H}$ is then given by

$$\mathbf{x}' = \mathbf{H}\mathbf{x} = \mathbf{H}_2\mathbf{H}_1\mathbf{x} = \mathbf{H}_2\mathbf{X} \tag{1}$$

where $\mathbf{x}$ and $\mathbf{x}'$ are the corresponding homogenous 2-D point pair in images 1 and 2, respectively. $\mathbf{X}$ is the homogenous 3-D point in the projective space, which can also be expressed in terms of camera center and projective matrix of an image, as follows:

$$\mathbf{X} = \mathbf{H}_1\mathbf{x} = \mathbf{C} + t\begin{pmatrix}\mathbf{N}\mathbf{x}\\ 0\end{pmatrix} \tag{2}$$

where $\mathbf{N}$ is the 3×3 projective matrix of an image, i.e., $\mathbf{P} = (\mathbf{N}^{-1}|\mathbf{p}_4), \mathbf{P}$ is the 3×4 camera matrix of an image, $\mathbf{p}_4$ is the fourth column of matrix $\mathbf{P}$ and t is a constant.

In addition, we may determine the projective plane, π, which contains all the three projective 3-D vertices of the triangle. Suppose that the three projective points defining plane π are

$$\mathbf{X}_1 = \begin{pmatrix}\tilde{\mathbf{X}}_1\\ 1\end{pmatrix} \quad \mathbf{X}_2 = \begin{pmatrix}\tilde{\mathbf{X}}_2\\ 1\end{pmatrix} \quad \mathbf{X}_3 = \begin{pmatrix}\tilde{\mathbf{X}}_3\\ 1\end{pmatrix}.$$

Then

$$\boldsymbol{\pi} = \begin{pmatrix}(\tilde{\mathbf{X}}_1 - \tilde{\mathbf{X}}_3)\times(\tilde{\mathbf{X}}_2 - \tilde{\mathbf{X}}_3)\\ -\tilde{\mathbf{X}}_3^{\mathrm{T}}(\tilde{\mathbf{X}}_1\times\tilde{\mathbf{X}}_2)\end{pmatrix}.$$

Since

$$\mathbf{X}^{\mathsf{T}}\boldsymbol{\pi} = \boldsymbol{\pi}^{\mathsf{T}}\mathbf{X} = 0. \tag{3}$$

From (2) and (3), we have

$$\begin{aligned}\boldsymbol{\pi}^{\mathsf{T}}\mathbf{X} &= \boldsymbol{\pi}^{\mathsf{T}}\mathbf{C} + t\boldsymbol{\pi}^{\mathsf{T}}\begin{pmatrix}\mathbf{N}\mathbf{x}\\ \mathbf{0}\end{pmatrix}\\ &= \boldsymbol{\pi}^{\mathsf{T}}\mathbf{C} + t\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}\mathbf{x} = 0\\ \frac{1}{t} &= -\frac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\boldsymbol{\pi}^{\mathsf{T}}\mathbf{C}}\mathbf{x} = -\frac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\mathbf{x}\end{aligned} \tag{4}$$

where $\pi^{\mathsf{T}} = [\tilde{\pi}^{\mathsf{T}}, 1]$, and $\mathbf{C}^{\mathsf{T}} = [\tilde{\mathrm{C}}^{\mathsf{T}}, 1]$.

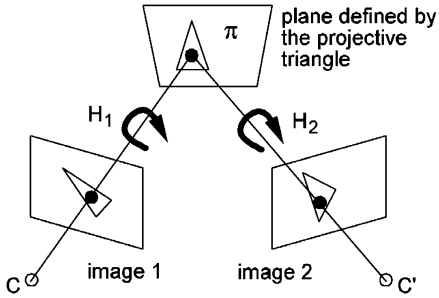


Fig. 4. Implicit geometry of a triangle.

From (2) and (4), we have

$$\begin{aligned}\mathbf{X} &\doteq \frac{1}{t}\mathbf{C} + \begin{pmatrix}\mathbf{N}\mathbf{x}\\ \mathbf{0}\end{pmatrix}\\ &= \mathbf{C}\begin{pmatrix}-\dfrac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\end{pmatrix}\mathbf{X} + \begin{pmatrix}\mathbf{N}\\ \mathbf{0}\end{pmatrix}\mathbf{x}\\ &= \begin{pmatrix}\left(-\dfrac{\tilde{\mathbf{C}}\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\right)+\mathbf{N}\\ -\dfrac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\end{pmatrix}\mathbf{x}\\ &= \begin{pmatrix}\left(\mathbf{I}-\dfrac{\tilde{\mathbf{C}}\tilde{\boldsymbol{\pi}}^{\mathsf{T}}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\right)\mathbf{N}\\ -\dfrac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\end{pmatrix}\mathbf{x}\\ &= \mathbf{H}_1\mathbf{x}.\end{aligned}$$

Hence, $\mathbf{H}_1$ in (1) can be determined as follows:

$$\mathbf{H}_1 = \begin{pmatrix}\left(\mathbf{I}-\dfrac{\tilde{\mathbf{C}}\tilde{\boldsymbol{\pi}}^{\mathsf{T}}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\right)\mathbf{N}\\ -\dfrac{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\mathbf{N}}{\tilde{\boldsymbol{\pi}}^{\mathsf{T}}\tilde{\mathbf{C}}+1}\end{pmatrix}. \tag{5}$$

$\mathbf{H}_2$ is, in fact, the camera matrix of image 2, $\mathbf{P}'$. Through offline computation of the trifocal tensor from correspondences [5], we can retrieve the camera matrices $\mathbf{P}, \mathbf{P}'$ and the projective matrix $\mathbf{N}$ in a fixed projective space. The 3-D points in the projective space from the 2-D image correspondences can be obtained to calculate plane π. Then, the plane homography can be obtained from (1).

After obtaining the plane homography, we may warp the corresponding triangles onto the same plane for pattern similarly check. Many distortion functions have been proposed to check for pattern similarity, such as mean absolute difference (MAD) [16], mean square difference (MSD) [2], and Pel difference classification (PDC) [6]. Here, we employ the PDC method because unlike other methods that cumulate the total difference of all the pixel pairs from the two image blocks, the PDC distortion function compares each pixel pair and classifies them as either matched or nonmatched. The PDC function is defined as follows:

$$\sum_{q=1}^{n}\sum_{p=1}^{m}\left[\mathrm{ord}\left(\left|m_{pq}^1 - m_{pq}^2\right| \le t_s\right)\right]$$

where t_s is the similarity threshold, m_{pq}^i is the luminancy of pixel (p, q) in image i and $\mathrm{ord}(e)$ evaluates to 1, if e is true and 0 otherwise.

E. Edge-Constrained Triangulation and Rematching

There are two reasons for the existence of some unmatched triangles. First, they may represent occluded or nontextured regions, which cannot be matched. Second, they may represent regions that could be matched but due to matching errors, they are not matched.

In this step, we try to identify regions belonging to the second situation and rematch them. By fixing the boundary vertices of the matched meshes, we connect the meshes together to form a new set of triangles among the meshes and ensure that the newly formed triangles do not overlap with the existing matched triangular meshes. We again compare the pattern of each of the newly formed triangles with those of the corresponding triangles in the other images. The matched triangles are then merged into the set of matched triangular meshes. The nonmatched triangles are simply removed. However, in the final iteration, we will keep the nonmatched triangles produced from this process because the triangles that are found only in some of the images likely represent some occluded regions of the scene. Hence, we merge these nonmatched triangles found in an image together with the matched triangular meshes to form the final output meshes for the image.

F. Generalized Disparity Calculation

In order to support arbitrary view synthesis, we need to compute the generalized disparity for each vertex of the triangular meshes. However, without performing a subpixel search, the accuracy of point matching is no better than a pixel. Considering also the error introduced by the estimated camera matrix, point correspondences may not be accurate enough to satisfy the geometric relations. Here, we use the *linear triangulation* method to obtain the maximum likelihood estimate of a 3-D point $\mathbf{x}$ from its 2-D correspondences $\mathbf{x}, \mathbf{x}'$ and $\mathbf{x}''$ in reference images 1, 2, and 3, respectively. Let $\mathbf{P}, \mathbf{P}'$ and $\mathbf{P}''$ be the 3×4 camera matrices of images 1, 2, and 3. Then, $\mathbf{x} = \mathbf{PX}, \mathbf{x}' = \mathbf{P}'\mathbf{X}$, and $\mathbf{x}'' = \mathbf{P}''\mathbf{X}$. These equations can be combined into the form of $\mathbf{AX} = 0$, where

$$\mathbf{A} = \begin{bmatrix} x\mathbf{p}^{3\mathsf{T}} - \mathbf{p}^{1\mathsf{T}} \\ y\mathbf{p}^{3\mathsf{T}} - \mathbf{p}^{2\mathsf{T}} \\ x'\mathbf{p}'^{3\mathsf{T}} - \mathbf{p}'^{1\mathsf{T}} \\ y'\mathbf{p}'^{3\mathsf{T}} - \mathbf{p}'^{2\mathsf{T}} \\ x''\mathbf{p}''^{3\mathsf{T}} - \mathbf{p}''^{1\mathsf{T}} \\ y''\mathbf{p}''^{3\mathsf{T}} - \mathbf{p}''^{2\mathsf{T}} \end{bmatrix} \quad \mathbf{p} = \begin{pmatrix} \mathbf{p}^1 \\ \mathbf{p}^2 \\ \mathbf{p}^3 \end{pmatrix} \quad \mathbf{x} = (\mathrm{x}, \mathrm{y}, 1)^{\mathsf{T}}.$$

This is a redundant set of equations. We perform singular value decomposition (SVD) 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 $\mathbf{A}$. After obtaining the value of $\mathbf{X}$, the generalize disparity $\delta(\mathbf{x})$ for a 2-D image point $\mathbf{x}$ can be estimated as

$$\delta(\mathbf{x}) = \frac{|\mathbf{Nx}|}{|\tilde{\mathbf{C}} - \tilde{\mathbf{X}}|} \quad \text{where} \quad \mathbf{C} = \begin{pmatrix} \tilde{\mathbf{C}} \\ \mathbf{1} \end{pmatrix} \quad \text{and} \quad \mathbf{X} = \begin{pmatrix} \tilde{\mathbf{X}} \\ \mathbf{1} \end{pmatrix}.$$

G. Geometric Proxy Representation and Rendering

The output of the image registration process is represented by a geometric proxy, called *relief occlusion-adaptive mesh* (ROAM) [36], for rendering. A ROAM is a hybrid 2-D/3-D mesh. For the matched triangular meshes, they represent some common regions of the scene in 3-D spatial domain and thus a single set of 3-D triangular meshes may represent the whole set of reference images. For the nonmatched triangular meshes, they represent the occluded regions visible only to individual images and hence they are simply modeled as 2-D meshes for the images. After storing the image registration result as ROAM's, arbitrary views can be generated using a rendering algorithm such as [36].

Unlike the standard mesh model, ROAM does not enforce the continuity of motion across the whole image and is thus able to represent object discontinuities. On the other hand, unlike existing 2-D meshes used in image morphing, ROAM contains the disparity information and is therefore able to support arbitrary view synthesis instead of purely linear interpolation.

V. Results and Discussions

We have implemented the proposed image registration method in Java and tested it on a PC with a Pentium 4 2.2-GHz CPU, 512 MB RAM. We have captured three reference images of an outdoor scene at a resolution of 640×480 as shown in Fig. 5(a) and applied the registration process to the images. The performance of this off-line process is shown in Table I.

In Fig. 5(a), the marked points represent the correspondences among the three images. We can see that by using the epipolar constraints, the matching accuracy of the feature points can be very high. Fig. 5(b) shows the constructed corresponding triangular meshes after applying Delaunay triangulation on the matched feature points. For the region between the background and the head of "Peppermint Patty" (the left most doll), some triangular patches in images 2 and 3 are flipped, with respect to those in image 1. The flipping of the patches is due to the relative motion between the background and the foreground object. Through the consistency checking, we may identify these regions as nonmatched.

Fig. 6(a) shows the matched meshes after the topological and pattern similarity checking. This consistency checking generally removes the mismatched feature points and triangles that cross the object boundaries. Fig. 6(b) shows the constructed triangular meshes after the edge-constrained triangulation process. The additional triangles are then rematched to ensure that their topology and pattern are consistent. Fig. 7 shows the resulting meshes after the rematching process.

After the first iteration of the image registration process, the matched meshes produced usually cover only a small portion of the image. We may conduct additional iterations to extend the matched meshes by lowering the feature point selection threshold. Fig. 8(a)–(c) shows the resulting matched meshes after the first, second, and third iterations, respectively. Fig. 8(d) includes the nonmatched meshes obtained from the thirrd iteration. We can see that much larger matched meshes can be

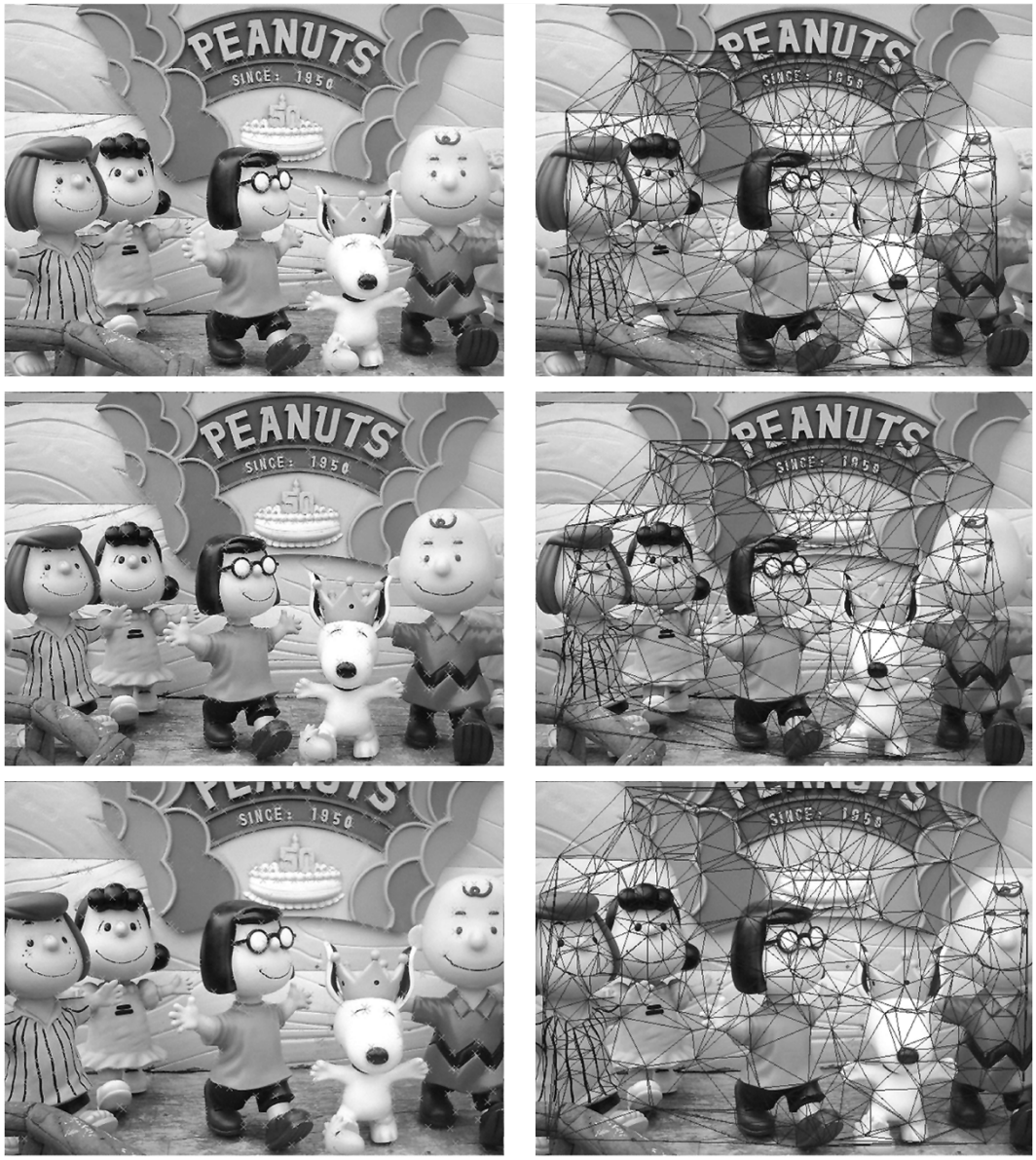


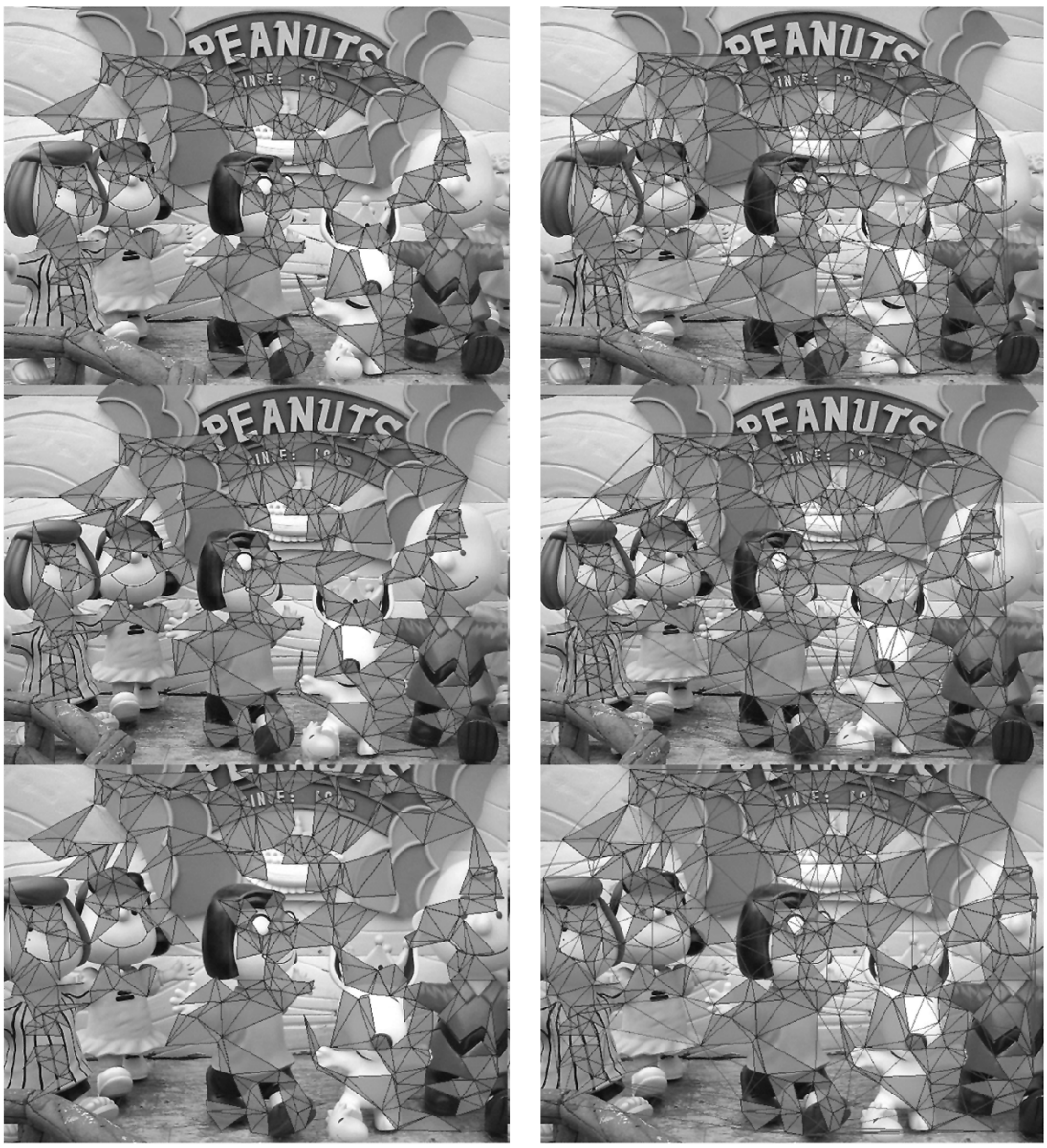
Fig. 5. (a) Correspondences among the three reference images. (b) Corresponding triangular patches.

TABLE I
LIST OF IMAGE REGISTRATION TIME

Steps	Operations	Time (sec)
Initial	Feature Point Extraction & Computation	0.6
- 1st iteration	Matching Point Correspondences	6.3
	Triangulation and Consistency Checking	2.7
- 2st iteration	Matching Point Correspondences	13.1
	Triangulation and Consistency Checking	4.7
- 3st iteration	Matching Point Correspondences	26.4
	Triangulation and Consistency Checking	11.2
Final	Generalized Disparity Calculation	0.8
Total		65.8

obtained after three iterations of the image registration process. Table I shows the computational costs of individual iterations. If only a single iteration is required, the total processing time is less than 10 s. Even with three iterations, the total processing time is only around 1 minute. The processing time increases for later iterations because there is a significant increase in the number of extracted feature points in later iterations as we reduce the feature point selection threshold.

Comparing with existing stereo matching methods and the feature tracking method proposed by [3] for geometric proxy recovery, the experiment here demonstrates that our method has a significant improvement in performance, both in computation time and search range. The computation time of existing stereo matching methods is about one minute to several minutes for a search range of 20 to 40 pixels. However, Table I shows that our method requires only about one minute to finish the whole registration process even with a much larger search range of 200 pixels.



(a) (b)

Fig. 6. (a) Matched triangular meshes after consistency checking. (b) Corresponding triangular meshes after edge-constrained triangulation.

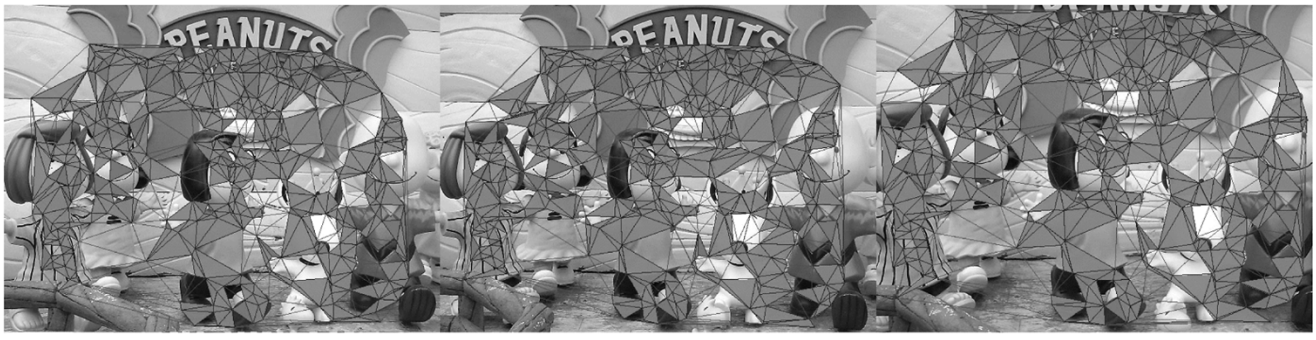


image 1 image 2 image3

Fig. 7. Meshes after rematching.

Comparing with existing IBR methods, our method has a much lower spatial sampling rate due to its ability to support a much larger search range. Hence, it can significantly reduce the preprocessing time and model size. In Lumigraph and light

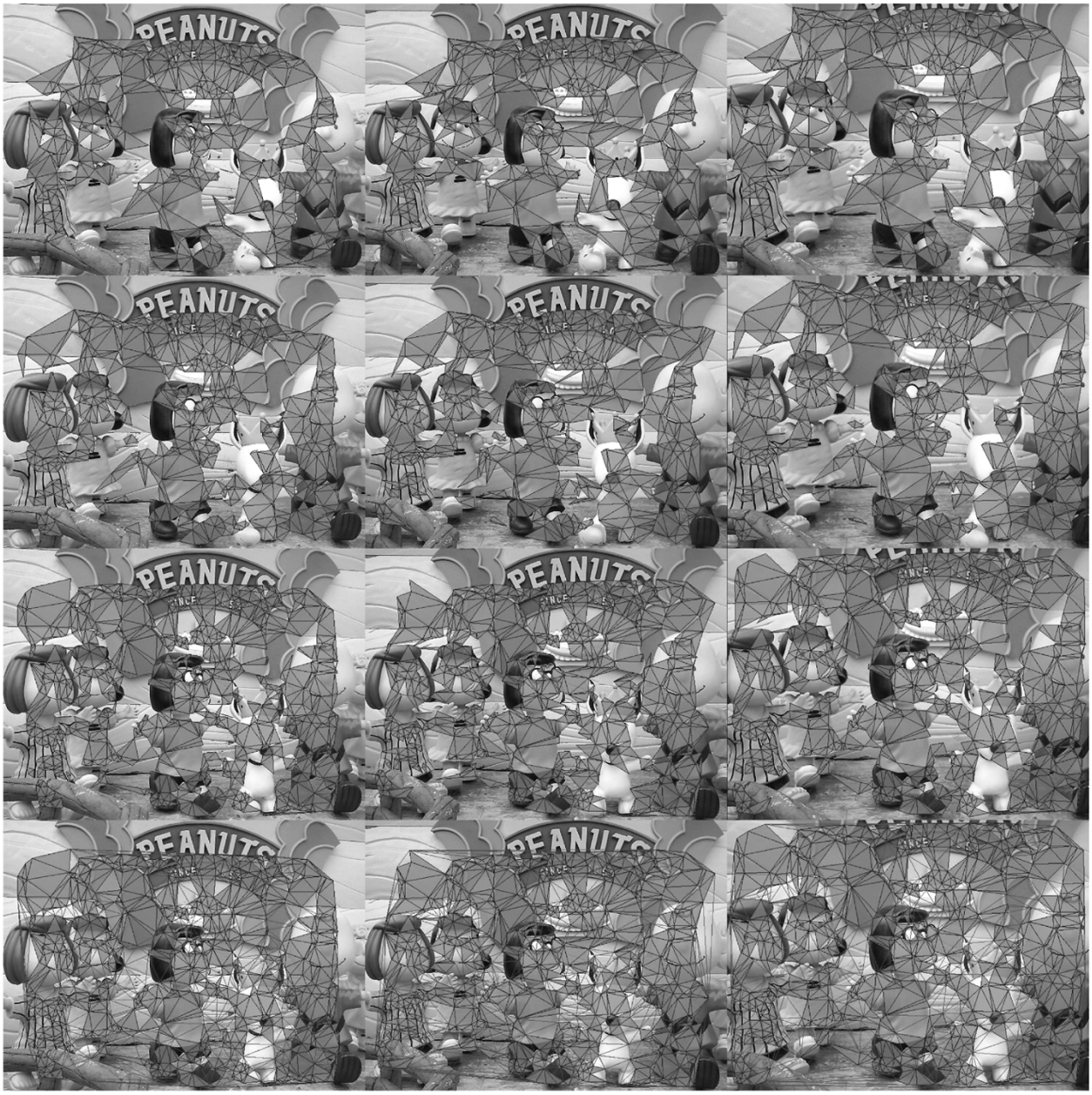


Fig. 8. Resulting meshes after (a) one iteration, (b) two iterations, (c) three iterations, and (d) three iterations with nonmatched meshes added.

field rendering, as the spatial sampling rate is high, they require hours of preprocessing time for rebinning and compression of large amount of images. For example, our model size for the experiment shown here is only 0.5 MB, which has already included the source images and geometric proxy. On the other hand, the model size of existing IBR methods is in the order of tens and hundreds of megabytes to model an object [19] and an environment [3], respectively.

Fig. 9 shows four synthesized arbitrary views of the three reference images shown in Fig. 5(a). Note that these reference images contain a large number of occluded regions, which are difficult to handle with existing stereo matching methods, and a lot of nontextured regions, which cannot be handled with existing dense feature matching algorithms. From Fig. 9, we can see that our image registration method can produce satisfactory results. Some video demos produced with our method can be obtained from: http://www.cs.cityu.edu.hk/~rynson/projects/ibmr/imagereg.html.

VI. Conclusion and Future Work

This paper analyzes various functions of the image registration process in image-based rendering and proposes a novel image registration method for recovering the geometric proxy. The method is based on a coarse-to-fine iterative approach. It starts with a high threshold value for extracting feature points. These points are then triangulated, matched, and checked for topological and pattern consistency among the three reference images to obtain a high quality geometric proxy. Building on the established geometric proxy, this image registration process can be iterated with a gradually reducing threshold.

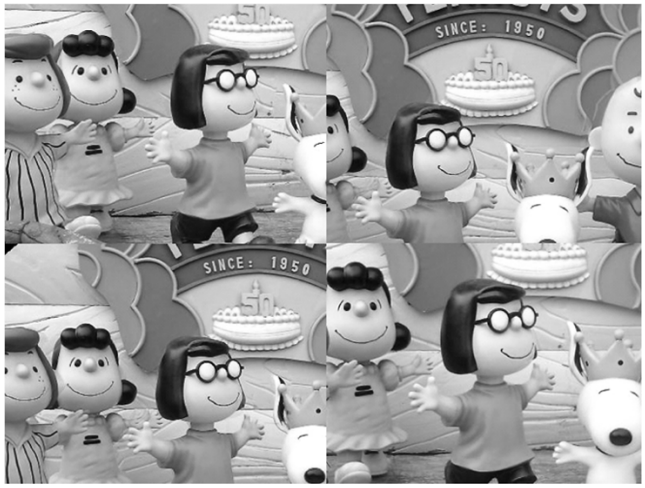


Fig. 9. Synthesis of arbitrary views.

With the distinguishing feature of supporting a large search range, our method can effectively reduce the spatial sampling rate. As a result, the image acquisition effort, the preprocessing time and the model size can all be significantly reduced. Hence, our method can be extended to support a scalable walkthrough environment, which is extremely expensive to do with existing IBR methods.

Since the spatial sampling rate can now be reduced significantly with the proposed image registration method, it may be more practical to develop higher dimension Plenoptic functions. As a future work, we are currently investigating the application of the proposed method in an IBR application to support the change of lightings and animation.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful and constructive comments on this paper. They would also like to thank A. Wan for some of the discussions.

REFERENCES

[1] E. Adelson and J. Bergen, "Chapter 1: The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*, M. Landy and J. Movshon, Eds. Cambridge, MA: MIT Press, 1991.

[2] M. Ahmad Fadzil and T. Dennis, "A hierarchical motion estimator for interframe coding," in *Proc. IEE Electron. Division Colloq. Applications of Motion Compensation*, Oct. 1990, pp. 3/1–3/6. Dig. No. 1990/128.

[3] D. Aliaga and I. Carlbom, "Plenoptic stitching: A scalable method for reconstructing 3-D interactive walkthroughs," in *Proc. ACM SIGGRAPH*, Aug. 2001, pp. 443–450.

[4] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured lumigraph rendering," in *Proc. ACM SIGGRAPH*, Aug. 2001, pp. 425–432.

[5] S. Carlsson, "Duality of reconstruction and positioning from projective views," in *Proc. IEEE Workshop Representation of Visual Scenes*, Jun. 1995, pp. 85–92.

[6] E. Chan, A. Rodriguez, R. Ghandi, and S. Panchanathan, "Experiments on block-matching techniques for video coding," *Multimedia Syst.*, vol. 2, no. 3, pp. 228–241, Dec. 1994.

[7] C. Chang, G. Bishop, and A. Lastra, "LDI tree: A hierarchical representation for image-based rendering," in *Proc. ACM SIGGRAPH*, 1999, pp. 291–298.

[8] J. Chai, X. Tong, S. Chan, and H. Shum, "Plenoptic sampling," in *Proc. of ACM SIGGRAPH*, Jul. 2000, pp. 307–318.

[9] E. Chen, "QuickTime VR: An image-based approach to virtual environment navigation," in *Proc. ACM SIGGRAPH*, 1995, pp. 29–38.

[10] K. Cornelis, M. Pollefeys, and L. Van Gool, "Tracking based structure and motion recovery for augmented video productions," in *Proc. ACM VRST*, Nov. 2001, pp. 17–24.

[11] P. E. Debevec, G. Borshukov, and Y. Yu, "Efficient view-dependent image-based rendering with projective texture-mapping," in *Proc. Eurographics Rendering Workshop*, Jun. 1998, pp. 105–116.

[12] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The lumigraph," in *Proc. ACM SIGGRAPH*, Aug. 1996, pp. 43–54.

[13] C. Harris and M. Stephens, "A combined corner and edge detector," in *Proc. Conf. Alvey Vision*, 1988, pp. 147–151.

[14] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambrige, U.K.: Cambridge Univ. Press, 2000.

[15] B. Heigl, R. Koch, M. Pollefeys, J. Denzler, and L. Van Gool, "Plenoptic modeling and rendering from image sequences taken by hand-held camera," in *Proc. DAGM*, 1999, pp. 94–101.

[16] F. Jaureguizar, J. Rona, and N. Garcia, "Motion compensated prediction on digital HDTV signal processing V: Theories and applications," in *Proc. EUSIPCO*, vol. II, Sep. 1990, pp. 753–756.

[17] C. Kurashima, R. Yang, and A. Lastra, "Combining approximate geometry with view-dependent texture mapping—A hybrid approach to 3-D video teleconferencing," in *Proc. SIBGRAPI*, Oct. 2002, pp. 112–119.

[18] Z. D. Lan and R. Mohr, "Robust location based partial correlation," INRIA Sophia-Antipolis, France, Res. Rep. 3186, 1997.

[19] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. ACM SIGGRAPH*, 1996, pp. 31–42.

[20] M. Lhuillier, L. Quan, H. Shum, and H. Tsui, "Relief mosaics by joint view triangulation," in *Proc. CVPR*, Dec. 2001, pp. 785–790.

[21] R. Li, B. Zeng, and M. Liou, "A new three-step search algorithm for block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 4, no. 4, pp. 438–442, Aug. 1994.

[22] L. Liu and E. Feig, "A block-based gradient descent search algorithm for block motion estimation in video coding," *IEEE Trans. Circuits Syst. Video Technology*, vol. 7, no. 4, pp. 419–422, Aug. 1996.

[23] D. McAllister, L. Nyland, V. Popescu, A. Lastra, and C. McCue, "Real-time rendering of real-world environments," in *Proc. Eurographics Workshop on Rendering*, Jun. 1999, pp. 21–23.

[24] L. McMillan, "An Image-based approach to three-dimensional computer graphics," Univ. North Carolina, Chapel Hill, NC, Tech. Rep. 97-013, 1997.

[25] S. Nayar, "Catadioptric omnidirectional camera," in *Proc. Comput. Vis. Pattern Recognit.*, Jun. 1997, pp. 482–488.

[26] L. Po and W. Ma, "A novel four-step search algorithm for fast block motion estimation," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 313–317, Jun. 1996.

[27] M. Pollefeys and L. Van Gool, "From images to 3-D models," *Commun. ACM*, pp. 50–55, Jul. 2002.

[28] Projective Vision Toolkit, Computational Video Group. IIT, NRC.

[29] S. Roy, "Stereo without epipolar lines: A maximum-flow formulation," *Int. J. Comput. Vis.*, vol. 34, no. 2/3, pp. 147–161, 1999.

[30] J. Shi and C. Tomasi, "Good features to track," in *Proc. Computer Vision and Object Recognition*, 1994, pp. 593–600.

[31] H. Shum and L. He, "Rendering with concentric mosaics," in *Proc. ACM SIGGRAPH*, Jul. 1999, pp. 299–306.

[32] H. Shum and R. Szeliski, "Stereo reconstruction from multiperspective panoramas," in *Proc. ICCV*, Sep. 1999, pp. 14–21.

[33] J. Shade, S. Gortler, L. He, and R. Szelishi, "Layered depth images," in *Proc. ACM SIGGRAPH*, 1998, pp. 231–242.

[34] A. Shashua, "Algebraic functions for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 779–789, Aug. 1995.

[35] C. Sun, "Fast stereo matching using rectangular subregioning and 3-D maximum-surface techniques," *Int. J. Comput. Vis.*, vol. 47, no. 1/2/3, pp. 99–117, 2002.

[36] A. Siu and R. W. H. Lau, "Relief occlusion adaptive mesh," in *Proc. IEEE ICME*, Jul. 2003, pp. 101–104.

[37] C. Tomasi and T. Kanade, "Detection and tracking of point features,", CMU Tech. Rep. CMU-CS-91-132, 1991.

[38] R. Tsai, "A versatile camera calibration technique for high-accuracy 3-D machine vision metrology using off-the-shelf TV cameras and lenses," *IEEE J. Robot. Automat.*, vol. RA-3, no. 4, pp. 323–344, Aug. 1987.

[39] Y. Yu, A. Ferencz, and J. Malik, "Extracting objects from range and radiance images," *IEEE Trans. Vis. Comput. Graph.*, vol. 7, no. 4, pp. 351–364, Apr. 2001.

Angus M. K. Siu received the B.Eng. degree in civil and structural engineering from the Hong Kong University of Science and Technology, Hong Kong, SAR, in 1996 and the B.Sc. (first-class honors) and Ph.D. degrees in computer science from the City University of Hong Kong in 2001 and 2004, respectively.

Prior to his research study, he was a System Architect with a network solution company. His research focuses on image-based modeling and rendering and camera pose estimation.

Rynson W. H. Lau received the B.Sc. degree (first-class honors) in computer systems engineering from the University of Kent, Canterbury, U.K., in 1988 and the Ph.D. degree in computer graphics from the University of Cambridge, Cambridge, U.K., in in 1992.

He is currently an Associate Professor with the City University of Hong Kong, Hong Kong, SAR. Prior to joining the university in 1998, he taught at the Hong Kong Polytechnic University. From 1992 to 1993, he was with the University of York, York, U.K., working on a defense project on image processing. His research interests include computer graphics, virtual reality, and multimedia systems.