

Image Interpolation by Adaptive Subdivision for Image-Based Walkthrough of Large Scale Real-World Scenes

Shiaofen Fang Yaraswy Bhupalam

Department of Computer and Information Science
Indiana University Purdue University Indianapolis
723 W. Michigan St., SL 280, Indianapolis, IN 46202, USA

Abstract

This paper presents a new image interpolation algorithm for image-based interactive walkthrough. An adaptive subdivision approach is developed to progressively establish feature correspondence between images taken at different locations within a real world scene. Using such correspondence, image interpolations can be interactively carried out within a triangulated domain area for real-time walkthrough. The algorithm is robust and fast, and is particularly suitable for digital library applications.

1. Introduction

The digital distribution of information and knowledge encompasses a wide range of different forms of multimedia data and application tools. New technologies in data acquisition, archiving and processing often lead to tremendous social, cultural, and economic implications. One of the most intriguing research frontiers in this process is the digitization, archiving, and visualization of large-scale 3D real world scenes, including both natural scenes and artificial structures. The ability to preserve, distribute, and visualize real world 3D visual information offers great potentials for a wide range of applications such as digital museum, tourism, e-commerce and entertainment. 3D visualization provides realism and immersive experience, with rich visual details and accurate perceptions, which are not only desirable but also imperative in many applications. One of these applications is the digital preservation of cultural heritage sites within a 3D virtual environment for online 3D digital exhibitions and virtual explorations and visitations. The work of this paper is conducted in the context of this application.

Our goal is to provide realistic 3D visual reproduction for large-scale real world scenes using image-based walkthrough technique as part of the virtual visitation and simulation experience. Using a set of image samples taken at the vertices of a triangulated region in a real world scene, we will first establish image correspondences between images at the vertices of each triangle. This will allow us to interactively reconstruct a new image at any point within the triangle by image interpolation. This scheme provides a complete visual model of the scene over the viewing region that can be used for virtual visitation in a virtual environment.

The key component of this process is the establishment of the accurate image correspondence, defined as a mapping (ideally one-to-one) between two images that would map the same feature in 3D to each other in the two images. It is extremely difficult to achieve accurate feature mapping for every pixel in the image. For visual walkthrough applications, however, approximated correspondence is usually good enough to generate novel images that are visually unambiguous and realistic. For real-time applications, simple representation of this correspondence is also highly desirable in order to facilitate real-time image interpolations.

Existing techniques for image-based walkthrough suffer two main problems: (1) robustness and (2) interpolation speed. Robustness in automatic image correspondence has always been a major problem. Most of the current techniques are not sufficiently scalable and reliable for applications such as digital library with large scale real world scenes. Many of the techniques also rely on real-time feature detection and optimizations, and thus often cannot meet the speed requirement for real-time applications. Our strategy in this work is to build a simple piecewise linear representation of image correspondence using adaptive triangular subdivision to facilitate fast image interpolation. To ensure robustness and scalability, we allow some manual selections of initial feature points to build an initial correspondence which will then be automatically refined to a desired resolution. As a practical solution, this is particularly suitable for digital library applications in which some manual pre-processing is appropriate.

In the following, we will first discuss some related work in Section 2. The data acquisition process will be described

in Section 3. In Section 4, the adaptive subdivision algorithm will be presented. Some implementation details and results will be given in Section 5. We will conclude the paper with additional remarks and future applications in Section 6.

2. Related work

Image-based rendering and modeling attempts to generate novel views from a set of existing images of a given scene [KAN99]. There are several variations. The simplest is panoramic viewing, such as QuickTime VR and IPIX images [CHE95]. Image morphing or interpolation is used to generate a sequence of intermediate images from two or more reference images so that transitions between the reference images can appear visually smooth. When morphing is used to represent camera movements between different view positions, a view morphing technique can be applied to reconstruct the correct viewing perspectives [SD96]. The main difficulty in image morphing and interpolation is to build a correct image correspondence between reference images. There is no robust way of automatically finding the image correspondence for morphing without additional information (e.g. depth, or scene geometry). Layered-depth image (LDI) technique is also used to solve the occlusion problem due to the change of visibility during re-projection [SGHS98]. Unfortunately, depth information is very hard to acquire, particularly for complex and natural scenes.

Image-based view reconstruction methods use large collections of images to approximate and reconstruct novel views. A complete visual representation of a 3D scene requires the construction of a five-dimensional plenoptic function [MB95], which captures the radiances of all rays from all positions and in all directions. In practice, however, only subsets of the plenoptic function are constructed due to complexity concerns and physical constraints. A well known technique in this category is the light field [LH96] and lumigraph [GGSC96] methods, which use a large set of images from known positions to create a database of light rays. As a very large number of images will be needed to reconstruct even a small object, it is not a proper technique for large scale real world scenes. Several extensions have been made recently, using non-uniform and adaptive sampling [SCG97][SHS99], unstructured input images coupled with geometric proxies [BBM*01], and visual hull rendering [MPN02].

Several recent works address the problem of image-based walkthrough of larger scale real world scenes. In the plenoptic stitching method given in [AC01], omnidirectional images are collected in an unobstructed space. Images are taken in a looped structure, allowing column-by-column interpolation between images on the opposite sides of the loop. Light rays are parameterized by their intersections with the viewing plane and the ruled surface to define the plenoptic function. Feature tracking is

necessary to keep track of image correspondence. This method is specifically designed for image-based walkthrough, but requires very dense and large number of images samples, which will not be feasible with digital library applications. The interpolation process also appears error-prone. A more recent work in [ATDC03] improves the results by computing a globally consistent set of image feature correspondences across all image samples. It starts from an initial set of point features, and tracks these features from image to image, identifying potential correspondences when two features track to the same position in the same image. Since automatic feature tracking and correspondence are not very reliable, robustness will be a problem in practical applications. Similar approaches using feature correspondence can be found in [JT01][MK94][PKG98]. In [SL05], iteratively generated feature points are also used to generate correspondence through Delaunay triangulations. 3D reconstruction without image correspondence was studied in [DSTT00]. It uses a maximum likelihood approach based on Markov Chain Monte Carlo sampling. But this approach appears to only work for relatively simple and large structures. General representation methods for image-based walkthrough have also been developed. An example is the IRW representation in [GMM02].

3. Image Acquisition

Our approach starts by collecting a set of static sample images of a given scene at the vertices of an initial domain mesh that covers the viewing area of a real world scene. The viewing area will be first carefully measured and draw as a domain mesh. A triangulation of the domain mesh will generate a triangle mesh of the field that will be used as the parametric domain for image reconstruction. The triangles in the domain mesh are called view triangles. Figure 1 gives a simple example (only two view triangles) of such scene configuration. The goal is to use the set of points that

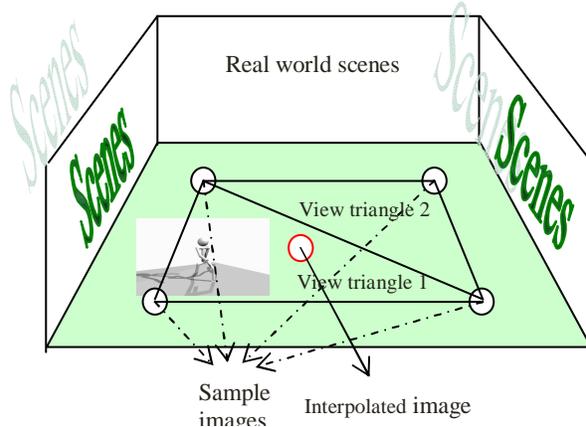


Figure 1. A simple example of scene configuration

generates the maximum view coverage of the scene. This is mostly a manual process, but may involve some simple scene analysis using a 2D map of the scene.

In our current implementation, only regular rectangular images are used as sample images. But in real applications, either cylindrical or spherical panoramic images can be used to cover the entire scene. But the image interpolation principles and algorithms are essentially the same. A new image can be defined at every point within a view triangle by image interpolation from the three sample images at the vertices of the view triangle.

4. Image Correspondence and Interpolation

This section describes the image reconstruction algorithm using the sample images acquired as discussed in the last section. There are two steps in the algorithm: (1) image correspondence and (2) image interpolation. Image correspondence is a pre-processing step that builds a pixel mapping between the three images within each view triangle. The image correspondence will then be used in the image interpolation process for real-time walkthrough.

4.1. Image Triangulation

Image correspondence, in our case, refers to the pixel correlations between the three sample images at the vertices of a view triangle. Ideally, the corresponding pixels on the multiple images should come from the same point in the 3D space. Image correspondence is extremely difficult to establish without depth information, and it is generally believed that no existing method is reliable enough to automatically compute image correspondence for arbitrary scenes, particularly for large scale, and complex natural scenes. For this reason, and because of the fact that most real world scenes in digital library applications are static scenes and can be heavily pre-processed, a semi-automatic approach (i.e. allowing some level of manual assistance) ought to be used to ensure the accuracy, reliability and robustness of the image correspondence process. In this approach, we first establish an initial image triangulation on each sample image based on a set of matching features points. These initial feature points can be selected manually, or we can also use an automatic feature detection process but apply manual matching and adjustment.

The initial feature points will be triangulated in each sample image to form three matching triangle meshes within each view triangle. The triangles within a sample image are called *image triangles* (as opposite to *view triangles*). This triangulation will then be further refined in an automatic process that adaptively subdivide the triangles to achieve accurate correspondence based on fitness function, which measures the "differences" between the three sample images of each view triangle. Once the correspondence is established, it will be stored within each

triangle's data structure and used for image interpolation during real time rendering.

4.2. Feature Computation

Image correspondence within each view triangle is defined by the three matching triangle meshes on the three sample images of the view triangle. This is done by a piecewise linear interpolation using the barycentric coordinates.

Each point P in a triangle ΔABC is uniquely identified by a barycentric coordinate (u, v, w) , defined by:

$$\begin{aligned} u &= S_{\Delta PAB} / S_{\Delta ABC} \\ v &= S_{\Delta PBC} / S_{\Delta ABC} \\ w &= S_{\Delta PCA} / S_{\Delta ABC} \end{aligned}$$

where S denotes the area. Clearly $0 \leq u, v, w \leq 1$, $u + v + w = 1$ and $P = uA + vB + wC$. The correspondence mapping is then defined as follows:

- Each image triangle corresponds to its matching triangle in the other two triangle meshes.
- Each point within the image triangle corresponds to the two points in the other two matching image triangles that have the same barycentric coordinate.

Let the matching triangles of an image triangle ΔABC be $\Delta A'B'C'$ and $\Delta A''B''C''$. Then the correspondence mapping are

$$\begin{aligned} f'(P) &= P' = uA' + vB' + wC' \\ f''(P) &= P'' = uA'' + vB'' + wC'' \end{aligned}$$

4.3. Adaptive Subdivision

The refinement algorithm attempts to recursively subdivide each image triangle to achieve desired correspondence accuracy. The fitness function is used to evaluate the "difference" between the three sample images based on the correspondence mapping. For each image triangle X , let its matching triangles be Y and Z . The fitness function defined on X is :

$$F_X(Y, Z) = \sum_{P \in X} (\|I(P) - I(f'(P))\| + \|I(P) - I(f''(P))\|)$$

where f' is the correspondence mapping from X to Y , f'' is the correspondence mapping from X to Z , and I is the intensity or color function. The total fitness function for the three sample images within each view triangle is then:

$$F(Y, Z) = \sum_X F_X$$

For each image triangle X, if $F_X > \delta$ for some threshold δ , we consider that this triangle requires subdivision. A simple triangle subdivision scheme is used as shown in Figure 2. New vertices are introduced along the edges of the triangle. The positions of these new vertices are determined numerically by minimizing the values of the fitness function of the resulting three triangles; i.e. finding the best new vertices [P, Q, R] along the edges such that function $F_{APR} + F_{BQP} + F_{CRQ}$ reaches the minimum.

When each new vertex is restricted to be on a triangle edge, it can be simply parameterized by the parameter of the edge. But for nonlinear features, shift in the perpendicular direction of the edge is necessary to bend the edge in different images. Thus an additional parameter is introduced for each new vertex in the perpendicular direction of the edge, as the vertex "U" in Figure 2(b). When an edge is subdivided by a new vertex, the vertex will also need to be included as part of the subdivision of the neighboring triangle in order to maintain image continuity. This can happen with two different configurations, as shown in vertex "V" or the vertices "U" and "W".

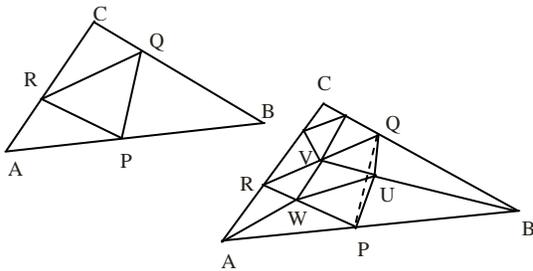


Figure 2: Triangle subdivision scheme

4.4. Image Interpolation

The refined triangles meshes define an image correspond for each view triangle that satisfy a predefined error bound. Barycentric coordinate will again be used to reconstruct novel images for any point within the view triangle.

Let ΔABC be a view triangle, and let P be a point within this view triangle with barycentric coordinate (x, y, z) . To define the new image at P , we will first generate the new image triangle mesh at P by computing a weighted average of the vertices of the triangle meshes at points A, B and C , with $[x, y, z]$ as the weights. This establishes an image correspondence between the pixels of the new image and the three samples images of this view triangle.

For each pixel Q of the new image, let ΔABC be the image triangle Q belongs to, and (u, v, w) be Q's

barycentric coordinate. Also, let the three matching image triangles of ΔABC in the three sample images be $\Delta A_1 B_1 C_1, \Delta A_2 B_2 C_2$ and $\Delta A_3 B_3 C_3$. The corresponding pixels of Q in the three sample images can then be computed as:

$$\begin{aligned} Q_1 &= uA_1 + vB_1 + wC_1 \\ Q_2 &= uA_2 + vB_2 + wC_2, \\ Q_3 &= uA_3 + vB_3 + wC_3 \end{aligned}$$

The intensity or color value at Q can be interpolated as:

$$I(Q) = uI(Q_1) + vI(Q_2) + wI(Q_3)$$

5. Results

The algorithm presented here has been implemented in Java, and applied in an image-based rendering experiment with the university library building. The view filed configuration is shown in Figure 3. We were able to interactively move the mouse within the view field and view smooth and continuous images in real-time.

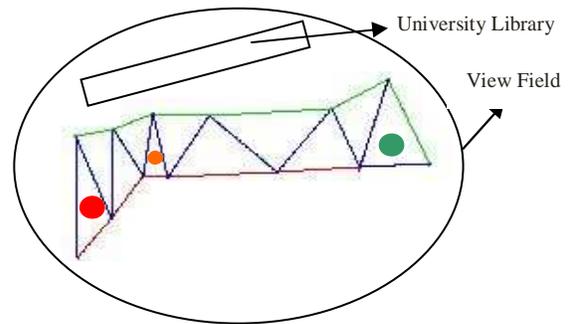


Figure 3. The view field used in the experiment

Figure 4 shows the subdivision process. Figure 4(a) is an example of the initial triangulations of a sample image in a view triangle. Figure 4(c) shows the subdivided triangle mesh after a number of adaptive subdivisions. Figure 4(b) is the interpolated image at a point inside a view triangle using the initial triangle meshes. Figure 4(d) shows an improved interpolation result using the mesh in Figure 4(c) (after subdivisions).

6. Conclusions

In this paper we presented an adaptive image subdivision algorithm for real-time image interpolation in image-based walkthrough applications. This technique is particularly suitable for applications with static scenes such as digital library, as heavy pre-processing is not a major issue in these applications. The main advantages of our algorithm

are its robustness and fast rendering speed. The initial triangle meshes ensure that the image correspondence will be confined within the given triangulations and subsequent subdivisions will only improve the correspondence. The piecewise linear interpolation using triangular meshes and barycentric coordinates is very simple and allows real-time interpolation and rendering.

Currently, we are in the process of applying this technique to a digital library system called CLIOH at Indiana University. CLIOH's mission is to digitally document and preserve fragile and threatened world heritage sites and artifacts as completely and accurately as possible (<http://clioh.informatics.iupui.edu/>) [PTTH02][FZPT03]. To date, the CLIOH team has documented, archived and indexed digital and multimedia footage from the Mayan cities of Chichen Itza and Uxmal in Yucatan, Mexico, the Angel Mounds, a Native American Mississippian site in southern Indiana and its attendant collection of archaeological artifacts, as part of an initial partnership with the Indiana State Museum. 3D digitization of these cultural heritage sites will provide invaluable enhancement to the CLIOH system, as well as the global effort in cultural heritage preservation.

References

- [AC01] D. G. Aliaga and I. Carlbom, "Plenoptic Stitching: A Scalable Method for Reconstructing 3D Interactive Walkthroughs," *Proceedings of SIGGRAPH'01*, pp. 443–450, Los Angeles, CA, August, 2001.
- [ATDC03] Daniel G. Aliaga, Thomas Funkhouser, Dimah Yanovsky, Ingrid Carlbom, "Sea of Images: A Dense Sampling Approach for Rendering Large Indoor Environments", *Computer Graphics & Applications*, Special Issue on 3D Reconstruction and Visualization, pp. 22-30, Nov/Dec 2003.
- [BBM*01] C. Buehler, M. Bosse, L. McMillan, S. Gortler and M. Cohen, "Unstructured Lumigraph Rendering," *Proceedings of SIGGRAPH'01*, pp. 425–432, Los Angeles, CA, August, 2001.
- [CHE95] S. C. Chen, "QuickTime VR – An Image-Based Approach to Virtual Environment Navigation," *Proceedings of SIGGRAPH'95*, pp. 29–38, Los Angeles, CA, August, 1995.
- [DSTT00] Frank Dellaert, Steven M. Seitz, Sebastian Thrun, and Charles Thorpe, "Feature Correspondence: A Markov Chain Monte Carlo Approach", in *NIPS*, 2000, pp. 852-858.
- [FZPT03] S. Fang, H. Zhou, M. Palakal, and S. Tennant, "Haptic Interaction for Virtual Museum", 9th International Conference on Virtual Systems and Multimedia, Montreal, 96-103, 2003.
- [GGSC96] S. J. Gortler, R. Grzeszczuk, R. Szeliski and M. Cohen, "The Limigraph," *Proceedings of SIGGRAPH'96*, pp. 43–54, New Orleans, August, 1996.
- [GMM02] David Gotz, Ketan Mayer-Patel, Dinesh Manocha, "IRW: An Incremental Representation for Image-Based Walkthroughs", *ACM Multimedia*, 2002.
- [JT01] David Jelinek and C.J. Taylor "Quasi-Dense Motion Stereo for 3D View Morphing" International Symposium on Virtual and Augmented Architecture (VAA01) pp 219-229, June 2001.
- [KAN99] S. Kang, "A survey of image-based rendering techniques," in *VideoMetrics, SPIE Vol 3641*, pp. 2–16, 1999.
- [LH96] M. Levoy and P. Hanrahan, "Light Field Rendering," *Proceedings of SIGGRAPH'96*, pp. 31–42, New Orleans, August, 1996.
- [MPN*02] W. Matusik, H. Pfister, A Ngan, P. Beardsley, R. Ziegler and L. McMillan, "Image-Based 3D Photography Using Opacity Hulls," *Proceedings of SIGGRAPH'02*, pp. 427–437, 2002.
- [MB95] L. McMillan and G. Bishop, "Plenoptic Modeling: An Image-Based Rendering System," *Proceedings of SIGGRAPH'95*, pp. 39–46, Los Angeles, CA, August, 1995.
- [MK94] Morita T. and Kanade T., "A Sequential Factorization Method for Recovering Shape and Motion from Image Streams", *Proc. ARPA Image Understanding Workshop*, Vol. 2, 1177–1188, 1994.
- [PTTH02] M. Palakal, S. Tennant, T. Townsend and J. Huang, "CLIOH: A Cultural digital Library Indexing Our Heritage", *Virtual Systems and Multimedia Conference*, Seoul, Korea, 2002.
- [PKG98] M. Pollefeys, R. Koch, and L. Van Gool, "Multi viewpoint stereo from uncalibrated video sequences", in *Smile98*, 1998.
- [SHS99] H. Schirmacher, W. Heidrich and H. Seidel, "Adaptive Acquisition of Lumigraphs from Synthetic Scenes," *Proceedings of Eurograph'99*, pp. 151–159, 1999.
- [SD96] S. M. Seitz and C. M. Dyer, "View Morphing," *Proceedings of SIGGRAPH'96*, pp. 21–30, New Orleans, August, 1996.
- [SGHS98] J. W. Shade, S. J. Gortler, L. W. He and R. Szeliski, "Layered Depth Images," *Proceedings of SIGGRAPH'98*, pp. 231–241, 1998.
- [SL05] Angus Siu and Rynson Lau, "Image Registration for Image-Based Rendering", *IEEE Trans. On Image Processing*, vol. 14, no. 2, pp. 241–252, 2005.
- [SCG97] P. P. Sloan, M. F. Cohen and S. J. Gortler, "Time Critical Lumigraph Rendering," *Proceedings of Symposium on Interactive 3D Graphics*, pp. 17–23, 1997.



(a)

(b)



(c)



(d)

Figure 4. A subdivision example.