PHOTOGRAMMETRIC MODELING AND IMAGE-BASED RENDERING FOR RAPID VIRTUAL ENVIRONMENT CREATION

Charalambos Poullis*, Andrew Gardner, Paul Debevec

University of Southern California
Institute for Creative Technologies
Marina Del Rey, California, 90292

ABSTRACT

For realistic simulations, architecture is one of the most important elements to model and render photorealistically. Current techniques of converting architectural plans or survey data to CAD models are labor intensive, and methods for rendering such models are generally not photorealistic. In this work, we present a new approach for modeling and rendering existing architectural scenes from a sparse set of still photographs. For modeling, we use *photogrammetric modeling techniques* to recover a the geometric representation of the architecture. The photogrammetric modeling approach presented in this paper is effective, robust and powerful because it fully exploits structural symmetries and constraints which are characteristic of architectural scenes. For rendering, we use *view-dependent texture mapping*, a method for compositing multiple images of a scene to create renderings from novel views. Lastly, we present a software package, named *Façade*, which uses the techniques described to recover the geometry and appearance of architectural scenes directly from a sparse set of photographs.

1 INTRODUCTION

A great deal of research and many different techniques have been proposed in the last several years to solve the problem of modeling the geometry and appearance of objects and scenes from the real world. Thus far, none of the available methods are able to produce both fast and realistic results. The most widely used methods are geometry-based and image-based.

In geometry-based methods, a modeling software is used, and the user is required to manually position the elements of the scene. Recreating virtual environments with this method often requires surveying the site (if possible), locating and digitizing architectural plans (if available), or converting existing CAD data (if available). The largest drawback of this method is the lack of an easy way to verify the accuracy of the model. Also, the renderings of such a model tend to look computer-generated.

In image-based methods, the results produced are photorealistic because real images are used as input. Techniques like the ones described in [4,5], have proved to be quite successful in recovering the 3D structure of a scene. Using a computer vision technique known as computational stereopsis the structure of the scene can be recovered from pairs of photographs. However, computational stereopsis has numerous limitations, the most significant being they require the photographs to appear very similar in order to recover reliable results. This limitation makes it impractical to reconstruct an architectural scene due to of the number of closely spaced images required.

In this paper, we present a hybrid approach that combines the strengths of these two methods. Our method requires a small set of sparse images and minimal user input in order to reconstruct a model of the scene and produce photorealistic renderings from arbitrary viewpoints as shown in Fig 2. Fig 1, shows resulting geometry reconstructed from a small set digital photographs of the Duveen gallery in London, UK.

Figure 1: The geometry of this interior space was reconstructed using the Façade v2.0 software package.

1.1 Background and Related Work

The process of recovering 3D structure from 2D images has been a central endeavor within computer vision. Rendering recovered structures is an emerging topic in computer graphics. Although no general technique exists to derive models from images, several areas of research have provided results that are applicable to the problem of modeling and rendering architectural scenes. There has been a considerable amount of work in this area and different methods have been proposed. Pertinent methods include: Camera Calibration, Structure from Motion, Stereo Correspondence, and Image-Based Rendering

1.1.1 Camera Calibration

Determining the 3D structure of a model from multiple views becomes simpler if the intrinsic (or internal) parameters of the cameras are known. Upon recovery, the cameras are said to be *calibrated*. These parameters define a mapping between image coordinates and directions relative to the cameras. The intrinsic parameters which describe this mapping are:

Figure 2: Left: Reconstructed virtual environment, Right: Rendering of the reconstructed model from an arbitrary viewpoint. From the Campanile movie (Electronic Theater at Siggraph '96)

1. the focal length (it may be different for the x and y direction),
2. the center of projection,
3. the skew angle, and
4. the distortion coefficients; radial and tangential coefficients that describe how the image is distorted.

Depending on the lens used, the lens distortion can introduce significant nonlinearities into the image. We have found that camera calibration is a straightforward process which can simplify the problem. Some successful methods include Tsai, 1987 and Faugeras et al,1986.

1.1.2 Structure from Motion

The problem of structure from motion is to determine from a sequence of images of a scene:

1. a geometric model of the scene (structure), and
2. the pose of the camera at each image (the camera motion over the sequence).

Thus, using the 2D projections of a sufficient number of points in the world, it is theoretically possible to deduce the 3D locations of the points as well as the positions of the original camera, up to an unknown factor of scale. Kruppa,1913 has proven that if given two views of five distinct points, one can recover the rotation and translation between the camera positions as well as the 3D locations of the points. This method has been shown to be quite successful in the case of orthographic cameras, but does not yield high-quality results for perspective cameras. Moreover, the models recovered by these algorithms consist of sparse point fields or individual line segments, which are not directly renderable as solid 3D models.

1.1.3 Stereo Correspondence

The geometric theory of structure from multiple images relies on being able to solve the *correspondence problem*. The correspondence problem is defined as the difficulty in identifying points in two or more images which are projections of the same point in the world. In the case where camera positions are closely spaced, this may not be a difficult problem, but as the distance between the cameras increases, the problem becomes more difficult to solve. Some of the major sources of difficulty include:

1. Foreshortening: surfaces in the scene viewed from different positions will be foreshortened differently
2. Occlusions: depth discontinuities can create half-occluded regions in an image pair
3. Lack of texture: if there is an absence of image intensity features somewhere it is difficult for stereo to locate the correct match.

Unfortunately, the alternative to using images taken from nearby locations has the disadvantage that computing depth becomes very sensitive to noise in image measurements. Furthermore, one must take an impractical amount of photographs in order to model an architectural site.

1.1.4 Image-Based Rendering

In an image-based rendering system, the model consists of a set of images of a scene and their corresponding depth maps. When the depth of every point in an image is known, the image can be re-rendered from any nearby point of view by projecting the pixels of the image to their proper 3D locations and reprojecting them onto a new image plane. Thus, a new image of the scene is created by warping the images according to their depth maps.

The requirement that samples be close together is a serious limitation to generating a freely navigable virtual environment. Converting the size of just one city block would require thousands of panoramic images closely spaced. Again, acquiring so many photographs is impractical.

Figure 3: A synthetic view of University High School. This is a frame from an animation of flying around the entire building.

1.2 Overview

In this paper we present two new modeling and rendering techniques: photogrammetric modeling, and view-dependent texture mapping. We show how these techniques can be used in conjunction to yield a convenient, accurate, and photorealistic method of modeling and rendering architecture from photographs. These techniques are implemented in our software package, named Façade v2.0, capable of recovering the geometry and appearance of architectural scenes directly from a sparse set of photographs, and exporting this information for reuse.

In Section 2, we describe our **photogrammetric modeling** method. Our method is able to recover the parameters of a constrained hierarchy of parametric primitives instead of recovering individual point coordinates. As a result, accurate architectural models can be recovered robustly from a small set of photographs and with a minimal number of user-supplied correspondences.

In Section 3, we describe our **image-based rendering and view-dependent texture-mapping** method. Upon initial inspection, projecting images onto a reconstructed model seems simple. However, difficulties arise when multiple images must be combined in order to color the faces of the model. Our method provides a solution to this problem by interpolating between the available photographs of the scene depending on the user's point of view. This results in lifelike animations that better capture surface specularities and unmodeled geometric detail.

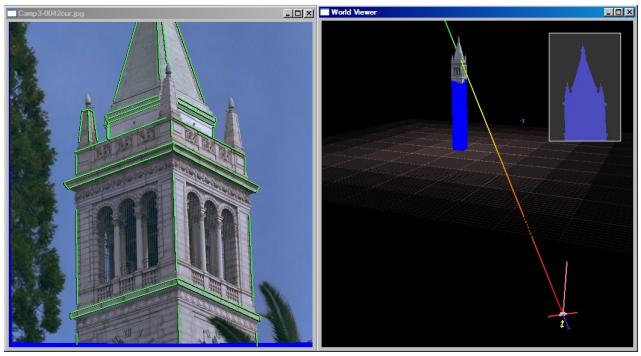

Figure 4: Two types of windows used by Façade v2.0 Left: The Image Viewer shows a photograph recovered in the field and annotated with line correspondences, Right: The World Viewer shows the reconstructed geometry and camera position of the scene

2 PHOTOGRAMMETRIC MODELING

In this section we describe our photogrammetric modeling method, in which the computer determines the parameters of a hierarchical model to reconstruct the architectural scene. In our method, the computer solves for a small number of model parameters rather than a large number of point coordinates. This requires a relatively small amount of user interaction to compute the geometric details of the scene as well as the positions from which the input photographs were taken. We have implemented this method in Façade v2.0, an easy-to-use interactive modeling program that allows the user to construct a geometric model of a scene from digitized photographs.

2.1 Overview of the Façade v2.0 photogrammetric modeling system

Façade v2.0 is a software package for image-based modeling and rendering. It was originally developed to work on Silicon Graphics workstations, but was recently ported to the Windows PC platform. We first offer an overview of Façade v2.0 from the point of view of the user, then describe our model representation, and explain our reconstruction algorithm. Lastly, we present results from using Façade to reconstruct several architectural scenes.

2.1.1 The process of recovering camera positions and model parameters

Constructing a geometric model of an architectural scene using Façade is a straightforward incremental process. Typically, the user selects a small number of photographs from a larger set, and models the scene one piece at a time. The user may refine the model and include more images in the project until the model meets the desired level of detail.

The user instantiates the components of the model, marks edges in the images, and corresponds the edges in the images to the edges in the model. When instructed, Façade computes the sizes and relative positions of the model components that best fit the edges marked in the photographs.

Components of the model, called blocks, are parameterized geometric primitives such as boxes, prisms, and surfaces of revolution. A box, for example, is parameterized by its length, width, and height. The user models the scene as a collection of such blocks, creating new block classes as desired. Of course, the user does not need to specify numerical values for the blocks parameters, since these are recovered by the program.

The user may choose to constrain the sizes and positions of any of the blocks. In Fig. 4, most of the blocks have been constrained to have equal length and width. Additionally, the four pinnacles have been constrained to have the same shape. Blocks may also be placed in constrained relations to one another. Many of the blocks in Fig. 4 have been constrained to sit centered atop of the block below. Such constraints are specified using a graphical 3D interface. When constraints are provided, they are used to simplify the reconstruction problem by reducing the number of unknown parameters the system is solving for.

Lastly, the user may set any parameter in the model to be a constant value - this needs to be done for at least one parameter of the model to provide the models scale. The user marks edge features in the images using a point-and-click interface; a gradient-based technique as in Mortensen and Barrett, 1995 can be used to align the edges with sub-pixel accuracy. We use edge rather than point features since they are easier to localize and less likely to be completely obscured. Only a section of each edge needs to be marked, making it possible to use partially visible edges. For each marked edge, the user also indicates the corresponding edge in the model. Generally, accurate reconstructions are obtained if there are as many correspondences in the images as there are free camera and model parameters. Thus, Façade reconstructs scenes accurately even when just a portion of the visible edges are marked in the images, and when just a portion of the model edges are given

correspondences.

At any time, the user may instruct the computer to reconstruct the scene. The computer then solves for the parameters of the model which cause it to align with the marked features in the images. During the reconstruction, the locations from which the photographs were taken are automatically recovered.

To verify the accuracy of the recovered model and camera positions, Façade provides functionality to project the model into the original photographs. Typically, the projected model deviates from the photographs by less than a pixel. Fig. 5 shows the results of projecting the edges of the model into the original photograph.

Finally, the user may generate novel views of the model by positioning a virtual camera at any desired location. Façade will then use the view-dependent texture-mapping method of Section 3 to render a novel view of the scene from the desired location. Fig. 2 shows an aerial rendering of the tower model.

In the example illustrated in Fig 3., a total number of twelve photographs were used to recover the structure and produce the renderings. There is no requirement on where the photographs should be taken from or how many should be used. In some cases it may be possible to recover the structure from just a few photographs. This is the case in the example illustrated in Fig 2, where using constraints of symmetry made it possible to recover an accurate 3D model from a single photograph.

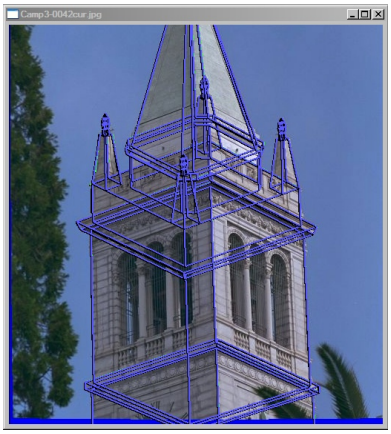

Figure 5: The reconstructed model is projected into the original photograph

2.1.2 The model representation

For the representation of the scene Façade uses a set of polyhedral blocks. Blocks are parameterized geometric primitives such as boxes, prisms, and surfaces of revolution. Each block has a set of parameters which denote its size and shape. The coordinates of the vertices of these polyhedra with respect to the block's internal frame of reference are expressed as a linear function of the block's parameters. For example, the coordinates for vertex ***Po*** of the wedge block in Fig. 6, are written in terms of the block parameters $width, height, and length$ as $Po = (-width, -height, length)^T$. A bounding box is also associated with each block as also shown in Fig. 6.

Each parameter of each instantiated block is actually a reference to a named symbolic variable, as illustrated in Fig. 7 bottom. As a result, two parameters of different blocks (or of the same block) can be equated by forcing each parameter to reference the same symbol. This facility allows the user to equate two or more of the dimensions in a model, which makes modeling symmetrical blocks and repeated structure more convenient. These constraints reduce the number of degrees of freedom in the model which simplifies the structure recovery problem.

The blocks are organized in a hierarchical tree structure where each tree node represents an individual block and the root of the tree establishes the world coordinate frame. The links in the tree contain spatial relationships between blocks as shown in Fig. 7 right. Typically, the relation between a block and its parent is represented by a rotation matrix R and a translation vector t (6-degrees of freedom). However, in architectural scenes fewer degrees of freedom(dof) may be needed. The rotation R between a block and its parent can be specified in one of three ways:

1. an unconstrained rotation in which case you have 3 dof
2. a rotation about a particular axis in which case you have 1 dof
3. no rotation in which case you have 0 dof.

The translation along a given dimension of the block can also be constrained and the user is able to force the bounding boxes of two blocks to align. Once the blocks and their relations have been parameterized, it is straightforward to derive expressions for the world coordinates of the block vertices.

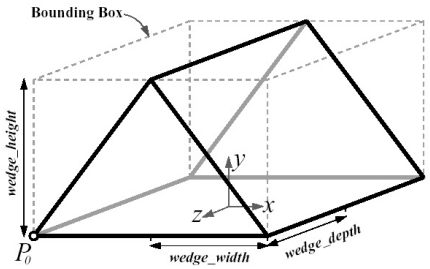

Figure 6: A wedge block with its parameters and bounding box

2.2 The reconstruction algorithm

This section presents the reconstruction algorithm used in Façade, which optimizes over the parameters of the model and the camera positions to make the model conform to the observed edges in the images. The algorithm also uses a two-step initial estimate procedure that automatically computes an estimate of the camera positions and the model parameters which are near the correct solution; this keeps the nonlinear optimization out of local minima and facilitates a swift convergence. We first present the nonlinear objective function which is optimized in the reconstruction.

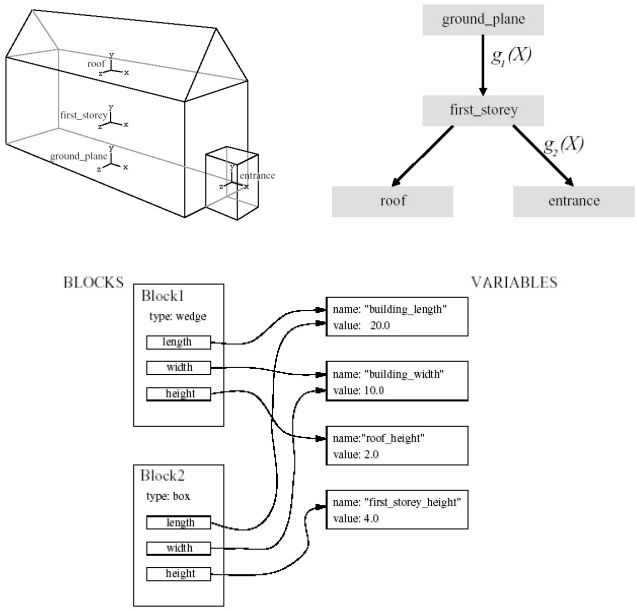

Figure 7: Left: A geometric model of a simple building, modeled with three parametric blocks. Right: The models hierarchical representation. The nodes in the tree represent parametric primitives (called blocks) while the links contain the spatial relationships between the blocks. Bottom: Representation of block parameters as symbol references.

Our reconstruction algorithm works by minimizing an objective function O that sums the disparity between the projected edges of the model and the edges marked in the images, i.e. $O = \sum Err_i$ where Err_i represents the disparity computed for edge feature i. The reconstruction computes the unknown model parameters and camera positions by minimizing O with respect to these variables. We minimize O using a variant of the Newton-Raphson method, which involves calculating the gradient and Hessian of O with respect to the parameters of the camera and the model. This nonlinear algorithm, however, may cause problems. If the algorithm begins at a random location in parameter space it may become trapped in local minima. For this reason we have developed a way to compute a reasonable initial estimate for the model parameters and camera positions. This initial estimate method consists of two procedures performed in sequence:

1. estimate the camera rotations first
2. then estimate the camera translation and parameters of the model.

Typically, the optimization requires fewer than ten iterations and adjusts the parameters of the model by at most a few percent from the initial estimates. The edges of the recovered models typically conform to the original photographs to within a pixel.

3 IMAGE-BASED RENDERING AND VIEW-DEPENDENT TEXTURE-MAPPING

Once a model of an architectural scene is recovered, photorealistic renderings may be produced. A traditional approach consistent with current rendering hardware Akeley,1993 and model file formats (e.g. VRML, OBJ, PLY, etc) involves specifying a texture map for each polygon in the model. Often, people who model architectural scenes pursue more realistic results by using texture maps cropped out of pictures of the scene itself. Unfortunately, the texture-mapped models obtained using such piecemeal texturing methods are often visually unconvincing. There are several reasons for this, but the most important are:

1. The textured-mapped models fail to capture the global illumination properties of the scene: radiosity, shadows, and varying levels of illumination from one part of the scene to another.
2. When texture mapping is used to simulate more than just changes in albedo (diffuse reflectance) over a surface, it can cause particularly dull effects. Especially when simulating inhomogeneous material properties, such as would be found on a mosaic with both marble and gilded tiles, or a building façade of glass and stone.
3. Even with texture blending, neighboring pixels can still be sampled from different views at the boundary of a projected image, which results in visible seams in the renderings.
4. The photographs may feature unwanted objects in front of the architecture, like trees, people, etc, which will project onto the model. Moreover, for complex structures there may be self-occlusion as shown in Fig. 8. The two top renderings and bottom left rendering, have occlusions which result in missing texture at some places of the architecture.

The process of texture-mapping a single image onto the model can be thought of as replacing each camera with a projector that projects the original image onto the model. This operation is known as *projective texture-mapping*, and Façade v2.0 uses this technique to map the photographs.

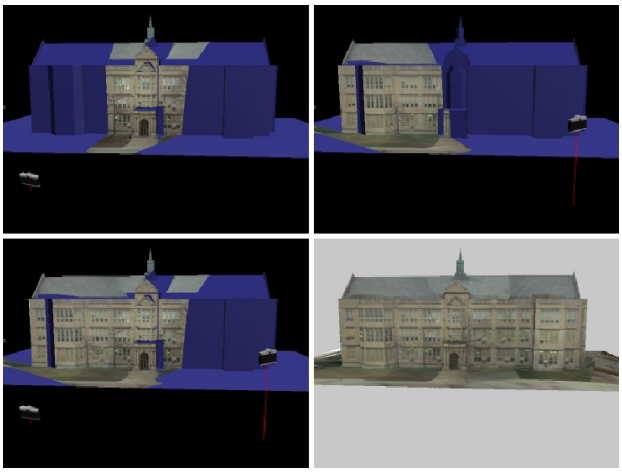

Figure 8: The process of assembling projected images to form a composite rendering. The top two pictures show two images projected onto the model from their respective recovered camera positions. The lower left picture shows the results of compositing these two renderings using our view dependent weighting function. The lower right picture shows the results of compositing renderings of all twelve original images.

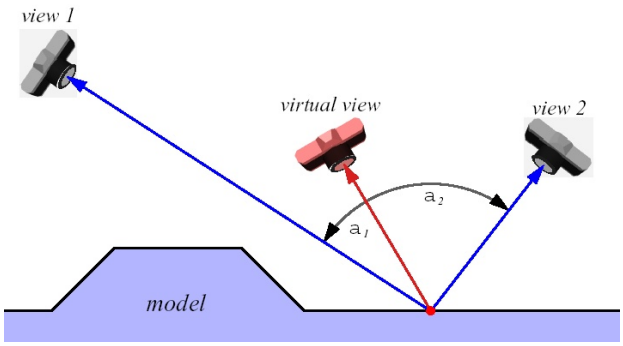

Figure 9: The weighting function used in view-dependent texture mapping. The pixel in the virtual view corresponding to the point on the model is assigned a weighted average of the corresponding pixels in actual views 1 and 2. The weights w1 and w2 are inversely proportional to the magnitude of angles a1 and a2. Alternately, more sophisticated weighting functions based on expected foreshortening and image resampling can be used.

3.1 Image-based rendering using multiple views

In general, each photograph will view only a piece of the model. Thus, it is usually necessary to use multiple images in order to render the entire model from a novel point of view. The top images of Fig. 8 show two different images mapped onto the model and rendered from a novel viewpoint. Some pixels are colored in just one of the renderings, while some are colored in both. These two renderings can be merged into a composite rendering by considering the corresponding pixels in the rendered views. If a pixel is mapped in only one rendering, its value from that rendering is used in the composite. If it is mapped in more than one rendering, the renderer must decide which image (or combination of images) to use. It would be convenient, of course, if the projected images would agree perfectly where they overlap. However, the images will not necessarily agree if there is unmodeled geometric detail in the building or if the surfaces of the building exhibits non-Lambertian reflection. Using the image closest in angle at every pixel means that neighboring rendered pixels may be sampled from different original images. To avoid this problem, we smooth these transitions through weighted averaging as shown in Fig. 9. Moreover, the pixel weights are ramped down near the boundary of the projected images, since the contribution of an image must be zero outside its boundary.

4 FAÇADE v2.0

In this section we describe the newer additions to the Façade v2.0 software package. Originally, Façade v1.0 was capable of only using geometric primitives like those described in section 2.1.2. It would then use the correspondences between the model and the images, specified as edges rather than points, and finally compute the model parameters and camera positions. A new data type of geometric primitive is now supported in Façade v2.0, namely a mesh block. A mesh block can be any type of geometry loaded as a polygonal OBJ file, which may contain texture coordinates and normals. By default, each mesh block is parameterized by width, height and length, and is a reference to a named symbolic variable, as illustrated in Fig. 7 bottom.

With this new functionality the user is able to load arbitrary geometry. High-resolution geometry, such as that acquired by a time-of-flight LIDAR scanner, can also be imported using a mesh block. When working with high-resolution scanned geometry, the user requires only the camera positions and none of the model parameters. Thus, Façade v2.0 also provides a method for the user to retrieve only the extrinsic parameters of the cameras. A point-to-point correspondence tool is used to mark points in the images and points in the 3D geometry. Once the correspondences are specified, the computer solves for the extrinsic parameters of the camera as shown in Fig.10. This method is very sensitive to noise in image measurements (the points clicked by the user) and can yield varying results. However, we have found it to be a good alternative to the original Façade method, especially when dealing with point-cloud data.

Another new feature of Façade v2.0, is the addition of a second numeric solver which allows an extended set of parameters to be solved for. In previous versions, the camera's intrinsic (or internal) parameters were fixed and could not be optimized except from the focal length. The new numeric solver makes it possible to optimize the camera positions, the model parameters, and the camera's intrinsic parameters; namely, center of projection and focal length. Moreover, the speed of the existing optimization technique was improved. A fairly complex scene can now be reconstructed in a few seconds. This is due to the improved speed of the algorithms as well as the use of more powerful computing hardware.

Lastly, Façade v2.0 has new and enhanced geometric model and texture features. High-dynamic range imagery can now be used as input and output. Furthermore, the geometric results can be exported into file formats such as OBJ, and VRML making it easier to incorporate with other software.

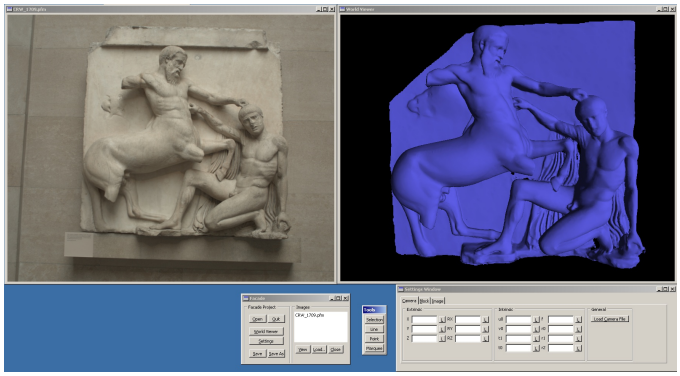

Figure 10: High-resolution mesh loaded into Façade v2.0. After specifying the correspondence between image points and world points, the computer can solve for the camera position.

4.1 Ongoing and future work

There are several other improvements we would like to make to Façade v2.0, in order to improve the functionality and usability of the system. Some of them include:

1. Visual hull recovery. Visual hulls, as described in Szeliski 1991, Matusic 2000, are a useful method for organically-shaped real-world geometry from different photographic viewpoints.

2. Camera calibration within Façade. As described in Section 1.1.1 camera calibration can simplify the problem of reconstructing virtual environments. For this reason we will include a camera calibration system within Façade v2.0. Thus the user can easily calibrate the cameras prior to reconstructing the model.

3. Export functionality to more widely used file formats. This will require creating texture atlases and visibility diagrams for each of the exported pieces of geometry.

4. Ability to add user-defined edges in a mesh block. Thus, being able to use a point-to-point correspondence algorithm as well as edge-to-edge correspondence algorithm.

ACKNOWLEDGEMENTS

We gratefully acknowledge Tim Hawkins, Chris Tchou, Andrew Jones and Andreas Wenger for helping with the software development and testing, John Lai for designing the poster for this paper and Charles-Felix Chabert for his contribution to this project. This research was originally sponsored by a National Science Foundation Graduate Research Fellowship and grants from Interval Research Corporation, the MICRO program, and JSEP program F49620-93-C-0014. The current work has been sponsored by U.S. Army contract number DAAD19-99- D-0046 and the University of Southern California; the content of this information does not necessarily reflect the position or policy of the sponsors and no official endorsement should be inferred.

REFERENCES

Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and Rendering Architecture from Photographs. In SIGGRAPH '96, August 1996.

Paul E. Debevec, George Borshukov, and Yizhou Yu. Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping. In 9th Eurographics Rendering Workshop, Vienna, Austria, June 1998.

Kurt Akeley. Realityengine graphics. In SIGGRAPH '93, pages 109-116, 1993

Oliver Faugeras and Giorgio Toscani. The calibration problem for stereo. In *Proceedings* IEEE CVPR 86, pages 15-20, 1986

E.Kruppa. Zur emitlung eines objectes aus zwei perspektiven mit innerer orientierung. Sitz-Ber. Akad. Wiss., Wien, Math. Naturw. Kl., Abt. Ila., 122:1939-1948, 1913

Leonard McMillan and Gary Bishop, Plenoptic modeling: An image-based rendering system. In SIGGRAPH '95, 1995

R.Szeliski. Image mosaicing for tele-reality applications. In IEEE Computer Graphics and Applications, 1996

Roger Tsai. A versatile camera calibration technique for high accuracy 3 machine vision metrology using off-the-shelf tv cameras and lenses. IEEE Journal of Robotics and Automation, 3(4):323-344, August 1987

W. Matusik, C. Buehler, R. Raskar, S. J. Gortler, and L. McMillan "Image-Based Visual Hulls", SIGGRAPH 2000 , Computer Graphics Proceedings, Annual Conference Series,

R. Szeliski, Shape from rotation, IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'91), pp. 625-630 (1991).

Eric N. Mortensen and William A. Barrett. Intelligent scissors for image composition. In SIGGRAPH '95, 1995

CONCLUSION

The technique used by Façade v2.0, is powerful because it solves directly for the architectural dimensions of the scene: the lengths of walls, the widths of doors, the heights of roofs, rather than multitude of vertex coordinates that a standard photogrammetric approach would try to recover. It can recover accurate models using the least amount of human effort of any commercially available photogrammetric modeling system. Moreover, Façade v2.0 toolset provides support for importing high-resolution geometry, such as that acquired by time-of-flight LIDAR scanners, thus allowing greater accuracy in the resulting virtual environment.