Reconstructing Polyhedral Models of Architectural Scenes from Photographs

Camillo J. Taylor and Paul E. Debevec and Jitendra Malik

EECS Department, U.C. Berkeley
Berkeley, CA 94720-1776
Fax: (510) 642 5775
email: {*camillo,debevec,malik*}*@cs.berkeley.edu*

Abstract. This paper presents a new image-based modeling method that facilitates the recovery of accurate polyhedral models of architectural scenes. The method is particularly effective because it exploits many of the constraints that are characteristic of architectural scenes. This work is placed in the context of the Façade project, whose goal is to use images to produce photo-realistic novel views of architectural scenes.

1 Introduction

The goal of our research is to develop a system that starts with multiple photographs of an architectural scene, such as King's College at Cambridge University, and produces photo-realistic images of the scene as it would appear from arbitrary virtual camera positions. Such a system has numerous potential applications: architectural walk-throughs, virtual tourism, virtual museums, and video games. Current methods for producing models of existing architectural sites are extremely labor intensive, error-prone, and do not produce photorealistic results. Our aim is to apply modern computer vision techniques to make the reconstruction problem tractable and photorealism achievable.

In the computer vision community, this problem statement evokes the use of techniques for scene reconstruction from multiple views. While the genesis of these techniques goes back to the photogrammetric literature, there has been a considerable development in the computer vision community under the topics of stereopsis and structure from motion. A good survey of the knowledge in this field up to 1992 is provided in Faugeras's book [Fau93]. Later work has focused on the use of uncalibrated cameras [FLR+95].

Most of the work that has been done on recovering the geometry of a scene from multiple images tackles the problem in its most general form. Their goal is typically to recover the 3D positions of a set of features, points or lines, from multiple image measurements. The mathematics of reconstruction from multiple views of a set of points or lines does not depend on whether these points/lines constitute a disembodied cloud of features, or lie on a smooth surface, or lie on a well constrained volumetric primitive. This generality comes at a price–it is well known that the recovery of scene structure from multiple views is sensitive to noise. Long image sequences help [WHA89, WHA93, TK92], but we believe that

at least part of the difficulty comes from the fact that the traditional formulation does not exploit all of the available constraints.

The main insight of this work is that architectural scenes can be well approximated as a collection of volumetric primitives linked together by specific geometrical relationships. Any remaining structure that is not captured in the coarse model can be expressed in terms of depth deviations from the coarse model. This suggests a two stage process for modeling the scene:

1. Recover a coarse model of the building – We have formulated this problem as one of recovering the parameters of a model supplied by the user rather than recovering the positions of individual points or lines. The advantage of modeling the scene with blocks instead of individual points and lines is that the scene can be represented with far fewer parameters (45 for the clock tower model in Fig. 1) which makes the reconstruction problem simpler. These models capture the constraints that are characteristic of architectural scenes such as rectangularity and parallelism.
2. Recover residual detail – Once we have obtained this coarse description of the scene, we apply a *model-based* stereo algorithm to pairs of images in our data set to recover the remaining geometric detail. Unlike previous approaches, we exploit the availability of a coarse model to simplify the problem of computing stereo-correspondences: the model is used to *pre-warp* the images in such a way that factors out foreshortening differences between the images and eliminates all image disparity except those resulting from deviations from the model.

Our approach splits the task of modeling from images into sub-tasks which are easy for a computer algorithm (but not a person), and sub-tasks which are easy for a computer algorithm (but not a person). The user chooses a parameterized model for the scene, which is precisely the type of information that would be difficult to recover automatically. Conversely, the computer recovers the parameters of the model from the image measurements which would be a very difficult task for a person. The geometric detail recovery is done by an automated stereo correspondence algorithm, which has been made feasible and robust by a pre-warping step based on the coarse geometric model. In this case, corresponding points must be computed for a dense sampling of image pixels, a job far too tedious to assign to a human, but feasible for a computer to perform using model-based stereo. Novel synthetic views of the scene are produced by combining the results of this two stage modeling process with the photometric information contained in the original images.

This paper describes the techniques we have developed to solve the first stage of our modeling process: to interactively obtain polyhedral models of architectural scenes from images. A complete description of the entire system can be found in [DTM96]. Section 2 of this paper describes the interactive polyhedral modeling program, Façade, along with the algorithms that have been developed to recover the parameters of a polyhedral model from image measurements. Some of the results obtained with the Façade system are presented in this section. In Section 3 we discuss our conclusions and future work.

2 The Interactive Modeling System

In this section we present *Façade*, a simple interactive modeling system that allows a user to construct a geometric model of a scene from a set of digitized photographs. In Façade, the user constructs a parameterized model of the scene and the program computes the parameters that best make the model conform to the photographs. We first describe the user interface, and then describe the model representation and algorithms involved.

2.1 User Interface

Constructing a geometric model of an architectural scene using Façade is an incremental process. Typically, the user selects a small number of photographs to begin with, 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.

Figure 1 shows the two types of windows used in the Façade program: image viewers and model viewers. The user supplies input to the program by instancing the components of the model, marking features of interest in the images, and indicating which features in the images correspond to which features in the model. Façade then computes the sizes and relative positions of the components of the model that best fit the features 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 each part of the scene as such a primitive; the user may also create new classes of blocks if desired. What the user does not need to specify are the numerical values of the blocks' parameters; these parameters are recovered automatically from the digitized photographs.

The user may choose to constrain the sizes and positions of any of the blocks. In Figure 1, most of the blocks have been constrained to have equal length and width. Additionally, the four pinnacles have been constrained to have the same proportions. Blocks may also be placed in constrained relations to one other. For example, many of the blocks in Fig. 1 have been constrained to sit centered and on top of the block below. Such constraints are easily specified using a graphical 3D interface. When such constraints are given, they are automatically used by Façade to simplify the reconstruction problem.

The user marks edge features in the images using a simple point-and-click interface; features may be marked with sub-pixel accuracy by zooming into the images. Façade uses edge rather than point features since they are easier to localize and less likely to be completely obscurred. Only a section of any particular edge needs to be marked, so it is possible to make use of partially visible edges. Façade is able to compute accurate reconstructions with only a portion of the visible edges marked in any particular image, particularly when the user has provided constraints on the model.

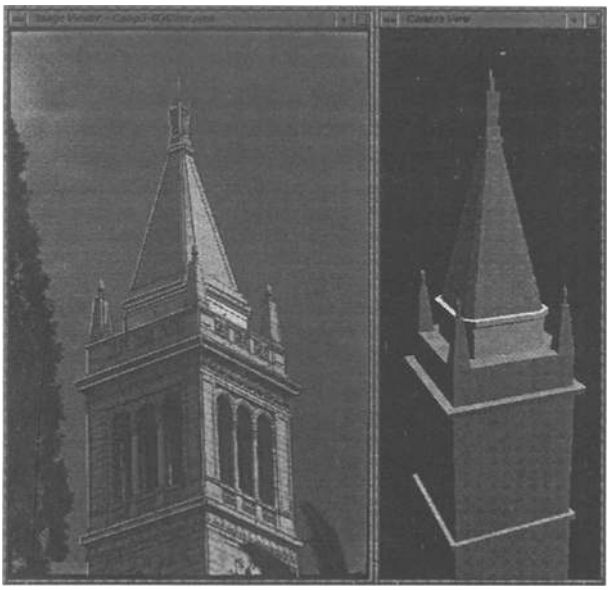

Fig. 1. An image viewer and a model viewer from the Façade modeling system. The left window shows a single photograph of a clock tower, with features the user has marked shown in green. The right window shows the recovered model, the parameters of which have been computed by Façade's reconstruction algorithm. Note that only the left pinnacle has been marked - the remaining three (including one not visible) have been recovered by encoding symmetry into the model. Although Façade allows any number of images to be used, in this case constraints in the model have made it possible to recover 3D structure from a single photograph.

With the edges marked, the user needs to specify which features in the model correspond to which features in the images. This is accomplished by clicking on an edge in the model and then clicking the corresponding edge in one of the images. The user can rotate his view of the model into a perferred orientation to make the process of forming correspondences easier.

At any time, the user may instruct the computer to reconstruct the model. The computer then solves for the parameters of the model that cause it to align with the observed features in the images. During the reconstruction, the computer also determines and displays the locations from which the original photographs were taken.

2.2 Model Representation

In Façade, an architectural scene is represented as a set of polyhedral blocks. Each block has a set of parameters which define its size and shape. The coordinates of the vertices of these polyhedra with respect to the blocks internal frame of reference can be expressed as a linear function of the block's parameters. For

example, in the wedge primitive shown in figure 2, the coordinates of the vertex P_o can be computed using the expression $P_o = (-width, -height, length)^T$. Each block has an associated bounding box, the maximum and minimum extent of this bounding box along each dimension can also be expressed as linear functions of the blocks parameters. For example, the minimum extent of the block shown in Figure 2 along the x dimension, $wedge_x^{MIN}$, can be computed from the expression $wedge_x^{MIN} = -width$;

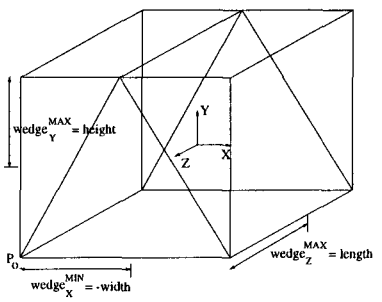

Fig. 2. A wedge primitive and its associated internal parameters.

Each class of block used in our modeling program is defined in a simple block template file. This file specifies the parameters of the blocks, provides the linear equations used to compute the vertices of the block and the limits of the bounding box, and defines how the vertices are connected by edges and faces. The user can add custom blocks to her project simply by creating appropriate template files.

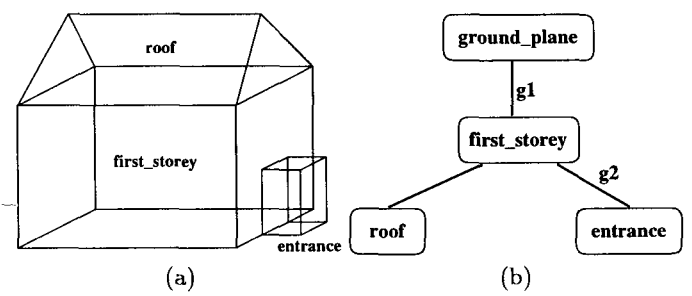

Fig. 3. (a) A simple geometric model of a building (b) The tree representation used in Façade. The nodes in this tree represent geometric primitives while the edges encode spatial relationships between the blocks.

The blocks in Façade are organized in a hierarchical tree structure as shown in figure 3(b). Each node of the tree represents an individual polyhedral block. The

root of the tree represents the ground plane, which defines the world coordinate frame of reference. The links in the tree encode spatial relationships amongst the blocks - each edge defines how a particular block is situated with respect to its parent. Similar tree representations are used in a number of commercial computer graphics packages, such as SGI's Inventor, to model complicated objects in terms of simple geometrical primitives.

The relationships between blocks in this model can be represented in terms of a rotation matrix R and a translation vector t. This type of representation would usually involve 6 degrees of freedom, 3 to specify the rotation, and three for the translation. However, in architectural scenes the relationships between the blocks are often constrained to a particular form which means that they can typically be represented with fewer parameters. Facade allows the user to capture these constraints on R and t in the model. First, the rotation R can be represented in one of three ways: by an unconstrained rotation, as a rotation about a particular coordinate axis or as a null rotation i.e. $R = I$.

Second, Facade allows the user to specify the constraints on each component of the translation vector t independently. He can either choose to represent the translation along a given dimension by an unconstrained variable, or he can choose to specify how the bounding boxes of the two blocks are aligned along that dimension. For example, in order to ensure that the roof block in Figure 3 lies on top of the first storey block the user would specify that the maximum y extent of the first storey block should be aligned with the minimum y extent of the roof block. This would imply that the translation term along the y axis would be computed from the following expression $t_y = (first\_storey_y^{MAX} - roof_y^{MIN})$.

Each parameter of each instantiated block is actually a reference to a named symbolic variable, as illustrated in Figure 4. As a result, two parameters of different blocks (or of the same block) can be equated by having each parameter reference the same symbolic variable. This facility allows the user to specify that two or more of the dimensions in a model are the same, which makes modeling symmetrical blocks and repeated structure more convenient. Importantly, these constraints reduce the number of degrees of freedom of the model, simplifying the model recovery problem.

Once the blocks and the relations have been parameterized, we can derive expressions which yield the coordinates of the vertices of the blocks in world coordinates. Consider the set of edges which link a specific block in the model to the ground plane as shown in figure 3. Let $g_1(X), g_2(X) \in SE(3)$ represent the rigid transformations associated with each of these links (X represents a vector of the unknown parameters). The coordinates of a particular vertex $P_w(X)$ are given by the expression:

$$P_w(X) = g_1(X)g_2(X)P(X) \tag{1}$$

Similarly, the orientation of a particular line segment with respect to the ground plane $v_w(X)$ could be computed with:

$$v_w(X) = g_1(X)g_2(X)v(X) \tag{2}$$

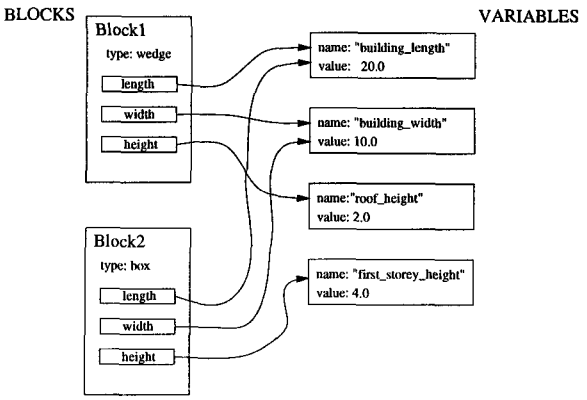

Fig. 4. Implementation of the dimensions of a block primitive in terms of symbol references. A single variable can be referenced in multiple places in the model, allowing constraints of symmetry.

In these equations, the point vectors, $P(X)$ and $P_w(X)$, and the orientation vectors, $v(X)$ and $v_w(X)$, are represented in homogeneous coordinates.

Modeling the scene with these polyhedral blocks, as opposed to points, line segments, surface patches, or polygons, offers a number of advantages which are listed below:

- Most buildings can be readily decomposed into a set of blocks.
- Blocks implicitly model common architectural constraints such as parallel lines and right angles.
- The user can conveniently manipulate the block primitives since they are at a suitably high level of abstraction, whereas individual features such as points and lines can be cumbersome.
- A surface model of the scene is readily obtained from the blocks, so there is no need to infer surfaces from discrete features.
- Modeling in terms blocks and relationships can greatly reduce the number of parameters that the reconstruction algorithm needs to recover. For example, the model in Fig. 1 is parameterized by just 45 variables. If each block in the scene were unconstrained (in its dimensions and position), it would require 240 parameters; if each line segment in the scene were treated independently, it would require 2896 parameters. This reduction in the number of parameters greatly enhances the robustness of the system relative to traditional structure from motion algorithms.

2.3 Reconstruction Algorithm

The reconstruction algorithm used in Façade is posed in terms of an objective function $\mathcal{O}$ which measures the disparity between the projected edges of the recovered model and the edges observed in the images, that is, $\mathcal{O} = \sum Err_i$ where

Err_i represents the disparity computed for correspondence i. Estimates for the unknown model parameters and camera positions are obtained by minimizing the objective function with respect to these variables. The error function used to measure the disparity Err_i is presented in [TK95]. This non-linear objective function is minimized using a variant of the Newton-Raphson method described in [TK94].

The minimization procedure involves calculating the gradient and hessian of the objective function with respect to the model parameters and camera positions. As we have shown in the previous sections, it is simple to construct symbolic expressions for the positions and orientations of the model edges in terms of the unknown parameters - these expressions can be differentiated symbolically to obtain expressions for the desired derivatives. This procedure is computationally inexpensive since the expressions given in equations 2 and 1 exhibit a particularly simple form.

The objective function described above is non-linear and can exhibit multiple local minima. If no initial estimate were available, the algorithm could converge to a local minimum instead of the global minimum. In order to overcome this problem we have developed effective methods for computing acceptable initial estimates for the model parameters and camera positions.

Before the non-linear minimization procedure described above is applied, two separate procedures are invoked to obtain initial estimates for the the unknown parameters. The first procedure recovers initial estimates for the camera rotations while the second provides initial estimates for the camera translations and the parameters of the model. Both of these procedures are based on techniques described in [TK95].

Once initial estimates have been obtained, the non-linear minimization over the entire parameter space is applied to produce a more accurate estimate for all of the unknown parameters. This stage typically reduces the error in the reconstruction by a factor of 2 to 4.

2.4 Results

Figure 1 showed the results of using Façade to reconstruct a clock tower from a single image. Figures 5 and 6 show the results of using Façade to reconstruct a high school building from twelve photographs. The photographs were taken with a calibrated Canon AE-1 35mm still camera with a standard 50mm lens and digitized with the PhotoCD process. Images at the 1536×1024 pixel resolution were processed to correct for lens distortion, then filtered down to 768×512 pixels for use in the modeling system. Fig. 5 shows that the recovered model conforms to the photographs to within a pixel, indicating an accurate reconstruction.

3 Conclusion

We have developed a two stage procedure, embodied in a system called Façade, for recovering geometric models of architectural scenes from photographs [DTM96].

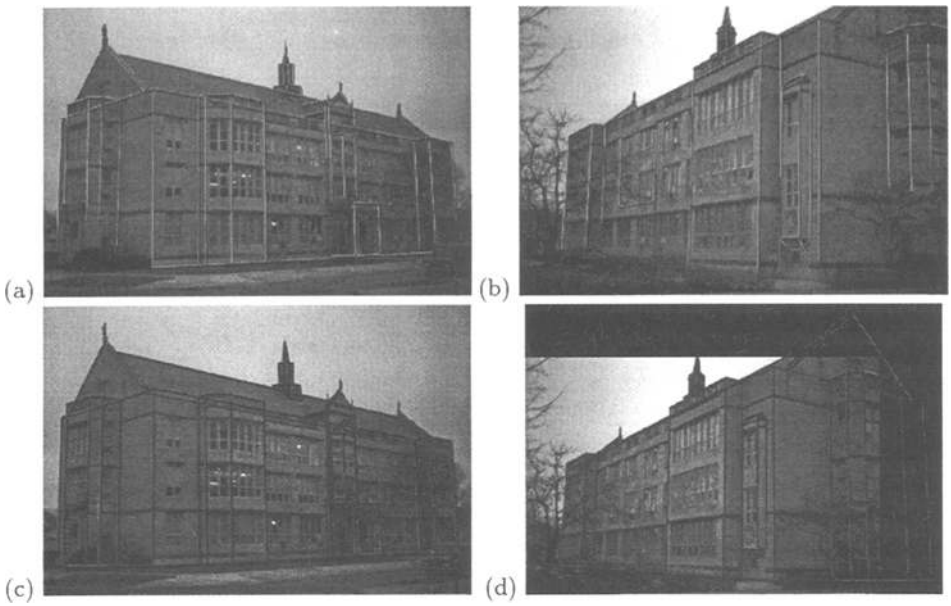

Fig. 5. Two of the twelve photographs used to reconstruct a high school building are shown in (a) and (b). The overlaid lines indicate the edges the user has marked. The edges of the reconstructed model, projected through the recovered camera positions and overlaid on the corresponding images are shown in (c) and (d). The recovered model conforms to the photographs to within a pixel in all twelve images, indicating that the building has been accurately reconstructed.

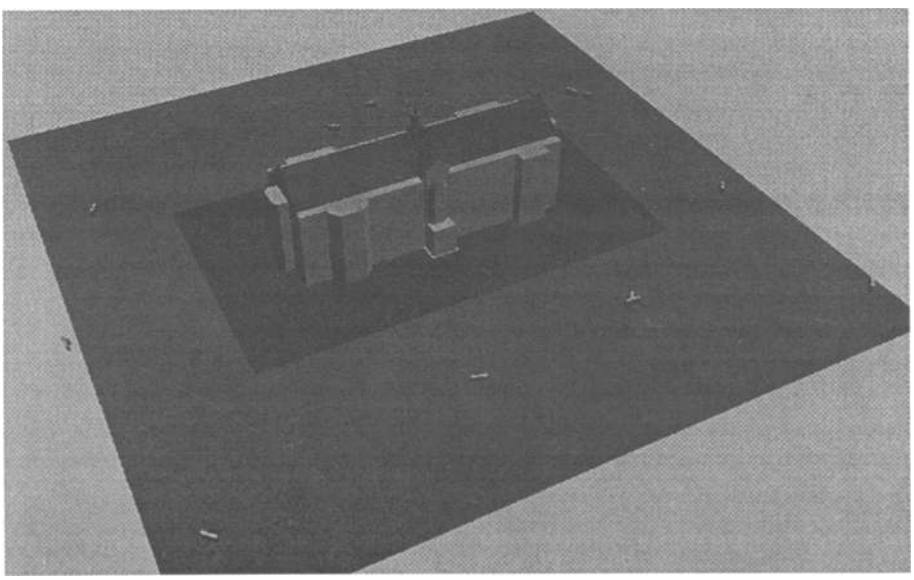

Fig. 6. Aerial view showing the recovered model and camera positions.

The first stage is that of reconstructing a coarse polyhedral model, and the second stage is that of *model-based stereo* which exploits the availability of a coarse model to simplify the problem of computing stereo-correspondences: the model is used to *pre-warp* the images in such a way that factors out foreshortening differences between the images and eliminates all image disparity except those resulting from deviations from the coarse model.

In this paper we have presented the first stage of the system: the polyhedral reconstruction algorithm. We have described how architectural scenes are modeled in terms of geometric primitives and spatial relationships and we have argued that this scheme allows the system to take advantage of many of the constraints inherent in the architectural domain. This polyhedral model is also a very natural representation for computer graphics and CAD programs which makes it much easier to interface the Façade system to these types of systems.

Allowing the user to specify this model to the system is an appropriate approach to this reconstruction problem since the human operator can easily provide a qualitative description based on her understanding of the scene being viewed. The Façade system can then recover the dimensions of the scene from image measurements much more effectively and accurately than the operator could. The approach presented in this paper can be extended to include other parameterized primitives, such as surfaces of revolution.

References

[DTM96] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. Technical Report UCB//CSD-96-893, U.C. Berkeley, CS Division, January 1996.

[Fau93] Olivier Faugeras. *Three-Dimensional Computer Vision.* MIT Press, 1993.

[FLR+95] Olivier Faugeras, Stéphane Laveau, Luc Robert, Gabriella Csurka, and Cyril Zeller. 3-d reconstruction of urban scenes from sequences of images. Technical Report Rapport de recherche 2572, INRIA Sophia-Antipolis, June 1995.

[TK92] Carlo Tomasi and Takeo Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[TK94] Camillo J. Taylor and David J. Kriegman. Minimization on the lie group so(3) and related manifolds. Technical Report 9405, Center for Systems Science, Dept. of Electrical Engineering, Yale University, New Haven, CT, April 1994.

[TK95] Camillo J. Taylor and David J. Kriegman. Structure and motion from line segments in multiple images. *IEEE Trans. Pattern Anal. Machine Intell.*, 17(11), November 1995.

[WHA93] Juyang Weng, Thomas S. Huang, and Narendra Ahuja. *Motion and Structure from Image Sequences.* Springer Series on Information Sciences. Springer-Verlag, Berlin, 1993.

[WHA89] J. Weng, T.S. Huang, and N. Ahuja. Motion and structure from two perspective views: Algorithms, error analysis, and error estimation. *IEEE Trans. Pattern Anal. Machine Intell.*, 11(5):451–476, May 89.