Building Models of Regular Scenes from Structure-and-Motion

Anton van den Hengel, Anthony Dick, Thorsten Thormählen, Ben Ward
School of Computer Science, University of Adelaide, AUSTRALIA
http://www.cs.adelaide.edu.au/~vision/

Philip H. S. Torr*
Department of Computing, Oxford Brookes University, UK
http://cms.brookes.ac.uk/staff/PhilipTorr/

Abstract

This paper describes a method for generating a model-based reconstruction of a scene from image data. The method uses the camera models and point cloud typically generated by a structure-and-motion process as a starting point for developing a higher level model of the scene. The method relies on the user to provide a minimal amount of structural seeding information from which more complex geometry is extrapolated. The regularity typically present in man-made environments is used to minimise the interaction required, but also to improve the accuracy of fit. We demonstrate model based reconstructions obtained using this method.

1 Introduction

The structure-and-motion approach to reconstructing a scene on the basis of a set of images is becoming a well understood process (see, for example, [6]). The results of this process are a parametric description of the cameras that took the images and a point-based reconstruction of the scene. However for many applications a model-based reconstruction, in which the scene is reconstructed as a set of interacting objects, is preferable to a point-based one. A model-based reconstruction allows the incorporation of semantic information about the scene, for example, as well as representing the shape of the objects therein. This makes the reconstruction far more versatile, allowing more powerful forms of manipulation, and the extension of models to areas not visible in the image set.

The approach we propose for generating model-based reconstructions does not rely on encoding large amounts of semantic information about plausible scenes [11]. Instead we allow the user to provide high level scene information. The method we present here utilises the relationships between groups of objects in a scene in order to enhance the reconstruction of each object instance and minimise the user interaction required. These *object-group relationships* reflect properties such as the fact that a set of objects are expected to be of the same size or equally spaced along one direction.

*Philip Torr would like to acknowledge EPSRC grant EP/C006631/1(P) for support

The method fits multiple, possibly interdependent, 3D models to image data. The first step in the algorithm is to calculate a full point-based reconstruction along with the associated camera parameters. This point-based reconstruction, the camera parameters and the 2D appearance information guide the fitting process.

The user also guides the fitting process by providing partial high level scene information about which objects are present and the relationships between them. These relationships encompass basic geometric concepts such as 'on top of', 'adjacent to' or 'within'. Importantly, it is also possible to specify that a set of objects in the scene are instances of the same model. This relationship can specify that the shape, orientation, size, etc of the objects is expected to be the same, and that they are regularly spaced (in 3D).

The intended application domain for this technique is imagery of man-made objects. Such imagery typically exhibits repetition and regularity. The primary shape models are relatively simple, but the conjunction of these shapes allows the modelling of more complex objects. Often, the set of reconstructed points associated with an object of interest will be too sparse to provide a useful guide to the shape of the object. If multiple instances of the object exist in the scene, however, the set of reconstructed points spanning these instances can be expected to provide more information.

Previous work has addressed the area of interactive scene reconstruction. The Facade system [9] reconstructs architectural scenes as a collection of polyhedra, but requires the user to outline each block in each image, and manually label corresponding features in each image—a time consuming process. In contrast our system can identify a block with a single mouse click in one image. Photobuilder [7] is an architectural modelling system that works by having the user highlight enough lines in each image to identify vanishing points in 3 orthogonal directions. Again, this is demanding and not always possible. The work of Sturm [8] operates on a similar principle, given a single image and significant user markup as input. Wilczkowiak presents a more general approach to interactive modelling based on parallelepipeds as scene primitives [12]. However this still requires the corners of the primitives to be marked manually in each image. This is because it does not make use of automated structure and motion estimation, instead trying to estimate all camera and scene information from user interaction. Other work which has made extensive use of prior information but is not interactive is necessarily limited in the range of scenes it can reconstruct, or slow to execute if the prior is not informative [3].

In the manipulation of 3D range data, model fitting is used to match either a pair of 3D point clouds [1] (and thereby estimating their relative pose and orientation) or a 3D point cloud and a pre-existing 3D model [4]. In neither case is fitting performed on the basis of both 2D and 3D data.

The contribution of this paper is a means by which recent progress in the recovery of structure-and-motion information may be incorporated into the process of interactively modelling scenes from image sets. The method goes beyond the estimation of structure to also recover semantic information about the contents of a scene. Additionally, although it does not provide a fully automated means of recovering such information, the method requires minimal guidance while still allowing total user control.

The rest of this paper is organised as follows. Section 2 describes the form of the models used by the system and the relationships between them. Section 3 explains the Bayesian interpretation which underlies the fitting process, and Section 4 the likelihoods which inform this interpretation. The interactive process of calculating the best fitting set of models is described in Section 5, and a selection of results are shown in Section 6.

2 Model Specification

Before describing the modelling algorithm itself, we define what a model is and the means by which we represent the relationships between models. Each model describes a particular class of object (for example, it might be a cube or a sphere), and each object of interest in a scene is identified with a corresponding instance of a model. Each such instance is identified by a label such as M, which by a slight abuse of notation, we also use to identify the vector of model parameters associated with that instance.

The form of the parameter vector associated with a model depends on its class. In general, the definition of any model includes a position $\mathbf{T}$ and a scale S. For a simple model such as a sphere, this may be all that is required. However, in general, models will also contain other parameters specifying their orientation, elongation and other relevant geometric properties. The models are organised hierarchically according to their parameters, a child model inheriting the parameters of its parent, and adding extra parameters specific to it. This allows us to formulate a general strategy for fitting models to data, as will be described later.

2.1 Example: The Bounded Plane Model

The bounded plane has a position $\mathbf{T}$ which is a point on the boundary of the plane. The plane is further defined by two orthogonal unit vectors $\mathbf{U}$ and $\mathbf{V}$ that intersect at $\mathbf{T}$ and belong to the plane. The scale S has a different meaning depending on the shape of the plane. If it is a general shape, there are two scale factors $S_{i,u}$ and $S_{i,v}$ for each point $\mathbf{P}_i$ on the boundary. The boundary of the plane is defined as a sequence of points enumerated in a counterclockwise direction when looking along the plane normal. The points are defined in terms of the plane position and scale, and the vectors $\mathbf{U}$ and $\mathbf{V}$: $\mathbf{P} = \mathbf{T} + S_{i,u}\mathbf{U} + S_{i,v}\mathbf{V}$. If it is a more regular shape, such as a square, the scale factors are shared between boundary points. In the case of a square there is only a single scale factor S which defines each of the 4 boundary points ($\mathbf{T}$, $\mathbf{T} + S\mathbf{U}$, $\mathbf{T} + S\mathbf{U} + S\mathbf{V}$, and $\mathbf{T} + S\mathbf{V}$).

2.2 Further Examples

Many models can be constructed as a set of bounded planes. A cuboid, for example, is made up of 6 rectangular bounded planes meeting at right angles, where opposing planes are the same size and have opposing normals. The parameters of the cuboid, $[\mathbf{T}, \mathbf{U}, \mathbf{V}, S_1, S_2, S_3]$, in turn define the parameters of the planes. In general, more complex models can be constructed from simple ones by specifying hierarchically that the parameters of each of the constituent models depend on another set of parameters that is global to the complex object. Spheres and other polyhedra can also be modelled in this way.

2.3 Relationships Between Models

There are a wide variety of relationships that might exist between even the small set of objects outlined above. These relationships are defined in terms of model parameters. One particularly important realtionship is 'abutting', which means that two objects have a face in common. This relationship can be used to represent the very common situation of one model resting on top of another. Other relationships are important for particular

types of scene. For office scenes 'edge aligned' is particularly important, and if multiple repeated objects are to be considered then so are 'co-linear' and 'equally spaced'. Such relationships are encoded probabilistically, as will be described in the next section.

3 Probabilistic Model Specification

We aim to find the set of models $\mathcal{M} = \{M_\xi : \xi = 1 \ldots \Xi$ that are most probable given the data $\mathcal{D}$ (images, camera parameters and 3D points) and any prior information $\mathcal{I}$. We represent the estimation problem as a Markov Random Field with a hidden node corresponding to each object in the scene and an observed node for each object observation. Hidden nodes are also added to capture the object-group relationships. Observed nodes are linked to the corresponding model nodes, as would be expected, with the pair-wise relationships between models providing the links between model nodes.

The Hammersley-Clifford theorem states that we can factorise the joint probability over the model set $\mathcal{M}$ as the (normalised) product of the individual clique potential functions of the graph formed by the nodes and their links [2]. The cliques in this case are all of size 2. The potential function adopted for the cliques containing an observed node and a model node is based on the probability of the model given the observation and the prior. For a model M,

$$\Pr(M|\mathcal{D}\mathcal{I}) \propto \Pr(\mathcal{D}|M\mathcal{I}) \Pr(M|\mathcal{I}). \tag{1}$$

It is the right hand side of this expression which forms the clique potential function for cliques containing an observed node and a model node.

The potential function for cliques which represent pair-wise relationships between two models M and N is the joint probability of the models: $\Pr(M,N)$. The potential function for cliques representing object-group relationships is similarly the joint probability $\Pr(M,R)$ of the model M and the relationship R.

The full joint probability of the set of models $\mathcal{M}$ and the set of object-group relationships $\mathcal{R}$ given the data set $\mathcal{D}$ and the prior information $\mathcal{I}$ is thus

$$\Pr(\mathcal{M}|\mathcal{D}\mathcal{I}) = \frac{1}{Z} \prod_{M \in \mathcal{M}} \Pr(\mathcal{D}|M\mathcal{I}) \Pr(M|\mathcal{I}) \prod_{N \in \mathcal{N}_M} \Pr(M,N) \prod_{R \in \mathcal{R}_M} \Pr(M,R), \tag{2}$$

where $\mathcal{N}_M$ represents the set of nodes N connected to M with $\varphi(M) > \varphi(N)$. The function $\varphi(\cdot)$ provides an ordering on nodes in order to ensure that each clique potential function is counted only once as is required under the Hammersley-Clifford Theorem. The set $\mathcal{R}_M$ represents the set of object-group relationships involving M, and the scalar Z a constant chosen such that $\Pr(\mathcal{M}|\mathcal{D}\mathcal{I})$ integrates to 1.

Our goal is to find $\mathcal{M}$ that maximises the joint probability. Because the joint is a product of probabilities, $\log \Pr(\mathcal{M}|\mathcal{D}\mathcal{I})$, whose extrema coincide with those of $\Pr(\mathcal{M}|\mathcal{D}\mathcal{I})$, can be written as a sum of log probabilities. We minimise the negative log joint probability (i.e. the sum of the negative logs of the terms on the right hand side of Equation (2)), which is easier to work with as described in the next section.

4 Evaluation of Probabilities

4.1 Model Likelihood

We now examine the calculation of the likelihood term $\Pr(\mathcal{D}|M\mathcal{I})$ in Equation (2) in more detail. There are in fact a number of likelihood functions that might be used to measure the degree of correspondence between a model and the data. Each likelihood gives rise to a different interpretation of the joint probability. In this section we describe 2 such functions, one based on the 3D point cloud and the other on the image data.

Recall that the data consists of a set of images of a scene (which we will call $\mathcal{D}_2$), but that from this data the camera parameters $\mathcal{D}_C$ and some 3D points in the scene $\mathcal{D}_3$ can be derived. The 3D point cloud is based on only a subset of the image data. However this subset was selected by a feature detector because it is likely to be informative. It is therefore often useful to fit a model first to the 3D point cloud, and then to refine the estimate using the image data more directly.

Given that the point cloud has been calculated by bundle adjustment, we use a likelihood for each point which is closely related to the reprojection error (i.e. the projection of each 3D point into each image). Let $\mathbf{P}_M$ be the point on the surface of the model M which is closest to the reconstructed data point $\mathbf{P}$. If we label the projection of a 3D point $\mathbf{P}$ into image I as $\mathbf{p}(\mathbf{P},I)$ then we wish to measure the distance between $\mathbf{p}(\mathbf{P},I)$ and $\mathbf{p}(\mathbf{P}_M,I)$ in each of the images that were used in the estimation of $\mathbf{P}$. The distance in image I is

$$d_2(\mathbf{p}(\mathbf{P},I),\mathbf{p}(\mathbf{P}_M,I)) \tag{3}$$

where $d_2(\cdot,\cdot)$ represents the Euclidean 2D image-based distance. Not all points in the reconstruction necessarily belong to the model that is being fitted, so a Huber function [5] $h(\cdot)$ is applied to the distance measure, to diminish the influence of points far from the model. The distance measure for a 3D point $\mathbf{P}$ thus becomes $h(d_2(\mathbf{p}(\mathbf{P},I),\mathbf{p}(\mathbf{P}_M,I)))$.

If we label the set of images containing the features from which point $\mathbf{P}$ was calculated as $\mathcal{K}_\mathbf{P}$, and assume that the reprojection errors conform to a Gaussian distribution, the negative log likelihood of a set of 3D points $\mathcal{P}$ given a model M is

$$\mathcal{J}_3(\mathcal{P},M) = -\log\Pr(\mathcal{D}_3|M\mathcal{I}) = f_3 \sum_{\mathbf{P}\in\mathcal{P}} \sum_{I\in\mathcal{K}_\mathbf{P}} h(d_2(\mathbf{p}(\mathbf{P},I),\mathbf{p}(\mathbf{P}_M,I))) \tag{4}$$

where f_3 is a constant scale factor. This assumes that the reconstructed points $\mathcal{D}_3 = \{\mathbf{P}_i\}$ are conditionally independent given the model M. The likelihood $\Pr(\mathcal{D}_3|M\mathcal{I})$ is substituted into (2) to form $\Pr(\mathcal{M}|\mathcal{D}_3\mathcal{I})$ which is used to initialise the model fitting process.

Other likelihood functions use the image data more directly, rather than the 3D point cloud derived from it. One such likelihood is based on the assumption that edges in the model will give rise to intensity gradients in the image. Edges have a number of advantages over corners or other features that might be used to guide model fitting, including rapid detection and relative robustness to changes in lighting. In order to calculate the degree to which a hypothesised model is supported by the image intensities the visible edges are projected back into the image set and the negative log likelihood $\mathcal{J}_2(\mathcal{D}_2,M) = -\log\Pr(\mathcal{D}_2|M\mathcal{I})$ is measured by the weighted distance to local intensity gradient maxima [10]. The likelihood $\Pr(\mathcal{D}_2|M\mathcal{I})$, is substituted into Equation (2) to form $\Pr(\mathcal{M}|\mathcal{D}_2\mathcal{I})$ which is used to refine the final fit.

4.2 Model to Model Probability Density Functions

The 'abutting' relationship is captured by a distribution over the parameters of the two models with its peak at the point corresponding to the co-location of the objects' faces. In testing a distribution has been used with parameters causing the intersection of the objects having zero probability and parameters causing the misalignment of the planes diminishing quadratically to zero.

The form of Equation (2) is based on cliques of size 2. In order to maintain this structure, 'abutting' is the only relationship defined to exist between a pair of objects. All other relationships must be mediated by an intermediate node. Most other relationships either involve groups of more than 2 objects, or require some property of the group to be derived, and are thus more naturally represented as an identifiable node in the graph. The 'co-linear' relationship, for example, requires the estimation of the line of best fit through the centres of the objects. The probability $\Pr(M,R)$ of a model in a 'co-linear' relationship decays as the distance of the model from the current line estimate increases, according to a Gaussian distribution.

5 Model Fitting

Having defined the model representation, and the associated density functions, we now describe an algorithm for fitting such models to image data. The final goal is to maximise the joint probability specified in Equation (2). Rather than using an iterative graph-based optimisation method, we aim to generate a good initial estimate which we improve through numerical optimisation. This is feasible because the graph is relatively simple, and ultimately the user can intervene (by adding more information) to ensure that the desired result is achieved.

Generating a suitable initial estimate is crucial to the success of the method. One approach to this problem might be to attempt a sample and test strategy, but the number of parameters involved preclude this as a means of effectively exploring $\Pr(\mathcal{D}|M\mathcal{I})$. Instead we use a strategy which exploits the nature of the functions $\Pr(\mathcal{D}_3|M\mathcal{I})$ and $\Pr(\mathcal{D}_2\mathcal{D}_C|M\mathcal{I})$ and the constraints between models to guide our search for a suitable initialisation.

The function $\Pr(\mathcal{D}_3|M\mathcal{I})$ relates the model to a set of reconstructed points and is well suited to gross localisation of the object in the scene, due to the relatively smooth nature of the associated probability distribution. The function $\Pr(\mathcal{D}_2\mathcal{D}_C|M\mathcal{I})$ relates the model to the appearance of the object in the image set, and is typically only applicable when the model is very close to the true location of the object. When this criterion is satisfied, however, it can achieve very precise localisation, as the associated probability distribution is typically strongly peaked. Thus the 3D likelihood function is better suited to initial localisation, while the 2D likelihood is appropriate for further optimisation based on this initial estimate.

The functions $\Pr(M,N)$ and $\Pr(M,R)$ from Equation (2) describe the relationships between the objects in the scene. Most objects of interest in a scene are expected to participate in such relationships, although this is not enforced as a criterion of reconstruction. The successful reconstruction of even a single object in a scene thus implies other areas that might be usefully interrogated. By following a chain of such implications a set of related objects can be identified without resorting to exhaustive search. This conditioning

of future search on previous success is feasible in part because the process is interactive. The fact that the set of possible relationships is prescribed is also an enabling factor.

5.1 Fitting Initialisation

We now explain the interactive fitting process in terms of a particular image set. Further examples follow.

The fitting process is initialised by the user choosing an object type from the set of available models and outlining an area in one of the images (see Figure 1). The set of

Figure 1: User selection of points corresponding to the first cuboid

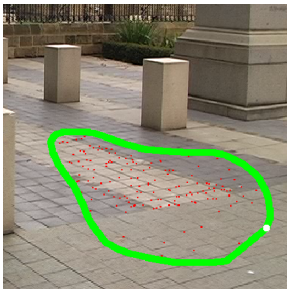

Figure 2: User selection of points corresponding to the plane.

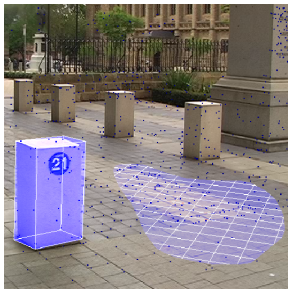

Figure 3: The best 3D fit (conjoined) cuboid and plane models.

3D points which reproject into this area in the selected image are taken to belong to an instance of the specified model. A numerical minimisation process is used to derive the parameters of the object on the basis of the 3D likelihood and this set of points. In the example shown in Figure 1 the result of this process is a model which fits the 3D data well, but does not match the underlying scene geometry accurately. This occurs due to the small number of reconstructed points associated with the object of interest. This problem can be solved, however, by manually adjusting the model, or by the addition of constraints as described above. The simplest constraint is that the cuboid is resting on the ground plane. In order to exploit this constraint the user selects a set of points on the plane as shown in Figure 2, and applies the constraint to the plane and previously identified cuboid. Figure 3 shows the result of fitting the plane and the cuboid collectively. The result of this interactive process is an initial estimate for the parameters of the cuboid and the plane, and the fact that the two share a common face.

5.2 Model Refinement and Replication

An instance of the MRF-based cost function which represents the joint probability of the cuboid and the plane is constructed and numerical minimisation of the associated negative log of the posterior $\Pr(\mathcal{M}|\mathcal{D}_3\mathcal{I})$ carried out. This numerical optimisation process is initialised with the estimate calculated by the method described in the previous section.

Having successfully fit plane and cuboid models to the data we use the resulting scene understanding to guide the remainder of the fitting process. To fit models to the remaining cuboids in the scene the user specifies that they are evenly spaced along a line. This information is recorded by the addition of an appropriate node to the MRF. The user then

selects a line of replication in the image. This line must be parallel to the plane, so there are 2 degrees of freedom in its specification, which is appropriate for an image-based interaction.

Instances of the cuboid model are generated along the line of replication as the user moves the cursor. An example of this process is shown in Figure 4. The instances are generated along the line so as to abut the previously identified plane, and are of the same size and orientation as the previously identified cuboid. Thus the only undetermined factor is the number of replications of the object. There are only a range of numbers of replicas possible without overlap, each of which is evaluated by comparing the colour histograms of the hypothesised objects with that measured from the original cuboid. It is thus not necessary to manually specify the number of replications of the object. Note that the hypothesised instances of the cuboid are rendered interactively, as they would appear when projected into the image, and that each instance is appropriately constrained. It should be noted that without the camera and scene information recovered by the structure-and-motion process this would not be possible.

Figure 4: Interactive specification of the line of replication.

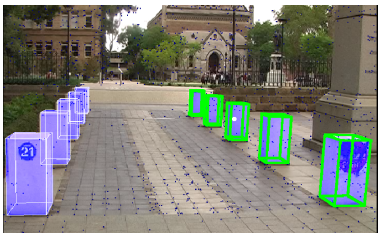

Figure 5: Subsequent replication of the optimised group.

The MRF at this point contains nodes for each of the objects, and a node representing the replication relationship. The plane node is connected to each cuboid by the 'abutting' relationship. The replication node is connected to each of the cuboids. A numerical minimiser is applied to the negative log of an instance of $\Pr(\mathcal{M}|\mathcal{D}_2\mathcal{I})$ representing this graph. The minimiser accurately models the 5 cuboids on the left of the scene. These cuboids, and the relationship that binds them, are then duplicated, and their representation in the image dragged toward the row of cuboids on the right of the scene. The two rows are assumed to be parallel and to abut the plane, so once again there are only 2 degrees of freedom in this interaction. The parallelism constraint is not represented in the MRF, however, and thus disappears once this interaction is complete. The numerical minimiser is once again applied sequentially to the negative log of $\Pr(\mathcal{M}|\mathcal{D}_3\mathcal{I})$ and $\Pr(\mathcal{M}|\mathcal{D}_2\mathcal{I})$ resulting in the final scene parameterisation.

At any stage of the fitting process it is possible to manually add information to the posterior represented by Equation (2), and thus to affect the outcome of the various optimisations. This information is added in the form of the term $\Pr(M|\mathcal{I})$ in Equation (2). The user adds this information by specifying the reprojection of an identified feature of a model in a particular image. As an example, the user might specify that the bottom left corner of a particular model projects into a specific location in the third image. This is added as a further Gaussian term in the posterior and is thus easily incorporated into the estimation process.

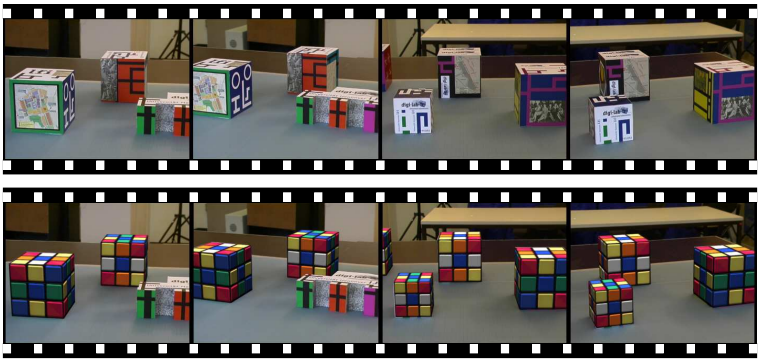

Figure 6: Reconstructed scene containing cubes resting on a tabletop.

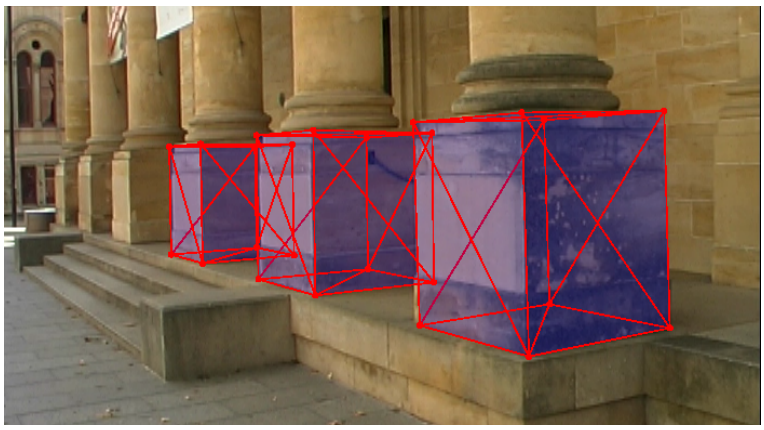

Figure 7: Reconstructed architectural scene.

6 Results

In Section 5 we demonstrated the modelling capabilities of the system by generating a model-based reconstruction of an outdoor scene. Figure 6 shows an indoor image sequence and a version of the same sequence which has been modified on the basis of a recovered model-based reconstruction. The processing has replaced each identified cube with a Rubik's cube to demonstrate the accuracy of the final fit. The cubes in the scene abut a common ground plane but are not aligned so as to be suitable for duplication by replication. The fitting is accurate nonetheless. Figure 7 shows models fit to an architectural scene using the replication mechanism.

7 Conclusion

Given a sequence of images of a scene, this paper has described a method for generating a parameterised model of the objects it contains and their relationships to each other. The method is based on the images themselves, the results of structure and motion estimation,

and the interactive specification of some shapes in the scene, and regularity in their layout. The model is generated very quickly, with a small amount of user interaction. It is visually convincing and can be used in ways not possible for point based reconstructions. These include object insertion with physical interaction, and the straightforward creation of special effects.

References

[1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[2] J. Besag. Spatial interaction and statistical analysis of lattice systems. *Journal of the Royal Statistical Society. Series B (Methodological)*, 32(2):192–236, 1974.

[3] A.R. Dick, P.H.S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, November 2004.

[4] R. Fisher, A. Fitzgibbon, M. Waite, E. Trucco, and M. Orr. Recognition of complex 3-d objects from range data. In *Proc. 7th International Conference on Image Analysis and Processing*, pages 509–606, 1993.

[5] P. Huber. Robust estimation of a location parameter. *Annals of Mathematical Statistics*, 35:73–101, 1964.

[6] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *International Journal of Computer Vision*, 59(3):207–232, 2004.

[7] D. Robertson and R. Cipolla. An interactive system for constraint-based modelling. In *Proc. 11th British Machine Vision Conference*, pages 536–545, 2000.

[8] P.F. Sturm and S.J. Maybank. A method for interactive 3d reconstruction of piecewise planar objects from single images. In *Proc. 10th British Machine Vision Conference*, pages 265–274, 1999.

[9] C.J. Taylor, P.E. Debevec, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *ACM SIGGraph, Computer Graphics*, pages 11–20, 1996.

[10] A. van den Hengel, A. Dick, T. Thormaehlen, P. H. S. Torr, and B. Ward. Fitting multiple models to multiple images with minimal user interaction. In *Proc.International Workshop on the Representation and use of Prior Knowledge in Vision (WRUPKV), in conjunction with ECCV'06*. (to appear), May 2006.

[11] D. Waltz. Understanding line-drawings of scenes with shadows. *Artificial Intelligence*, 2:79–116, 1971.

[12] M. Wilczkowiak, P.F. Sturm, and E. Boyer. Using geometric constraints through parallelepipeds for calibration and 3d modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 27(2):194–207, February 2005.