A Shape Hierarchy for 3D Modelling from Video

A. van den Hengel*, A. Dick, T. Thormählen, B. Ward
School of Computer Science
University of Adelaide
Adelaide 5005, Australia

P. H. S. Torr
Department of Computing
Oxford Brookes University
Oxford OX33 1HX, UK

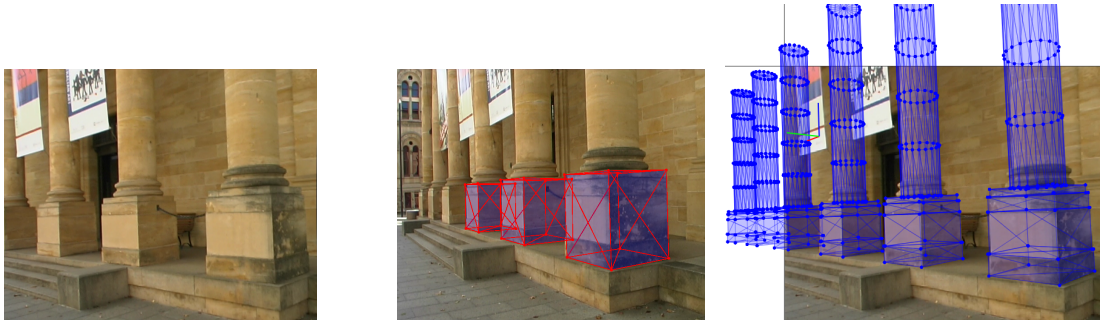

Figure 1: *Three images showing (left to right) a frame from the input sequence, part of the modelling process, and the final model*

Abstract

This paper describes an interactive method for generating a model of a scene from image data. The method uses the camera parameters and point cloud typically generated by structure-and-motion estimation as a starting point for developing a higher level model, in which the scene is represented as a set of parameterised shapes. Classes of shapes are represented in a hierarchy which defines their properties but also the method by which they are localised in the scene, using a combination of user interaction, sampling and optimisation. Relations between shapes, such as adjancency and alignment, are also specified interactively. The method thus provides a modelling process which requires the user to provide only high level scene information, the remaining detail being provided through geometric analysis of the image set. This mixture of guided, yet automated, fitting techniques allows a non-expert user to rapidly and intuitively create a visually convincing 3D model of a real world scene from an image set.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Surface Fitting; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling—Modeling packages

Keywords: Image-Based Modelling, Model-Based Reconstruction, Structure-from-motion

1 Introduction

This paper describes a system for the interactive 3D modelling of scenes from video footage. The first stage of the process is the recovery of camera parameters and a point cloud representing the location of various feature points within the scene by structure from motion anaylsis [Thormählen 2006]. The user is then presented with an interface which allows the selection of a particular shape (a cube, or plane for example), and a means by which its projection in one of the original images may be indicated. The indication appropriate for each shape varies according to the application and the expected density of the point cloud compared to the size of the objects to be modelled. Potential interactions include a single 'click' in order to specify the location of a cube in an image, or a plane might be identified by drawing a loop enclosing the corresponding group of scene points. One of the key advantages of the system presented here is the flexibility it offers in the interactions supported and the structured means by which new shape models may be incorporated.

The speed of the fitting methodologies developed allows an interactive fitting process, with each iteration through user interaction and fitting producing successively more accurate results. This process of iterative interaction means that the user need never provide more than the minimal information required to achieve the desired level of accuracy. The resulting shape model is that which best accommodates both the image data and the user-provided information. The user thus provides high-level shape information and specifies the relationships between shapes (if such a relationship exists). The system interprets this user input on the basis of projective geometry and information gained through automated image analysis techniques to derive a

The model that is generated is a collection of parameterised 3D shapes and a set of relationships between them. In order to fit such a model the following system components are presented:

- We define a hierarchical shape model of shape. At the topmost level of the hierarchy is the simplest shape—a point—while more complex shapes apppear at lower levels of the hierarchy. We will describe this model in Section 2.
- A strategy is devised for sampling for shapes that belong to the hierarchy. The strategy makes use of the hierarchy to efficiently search for a wide variety of shapes in the scene.
- A means by which a user can interact with the system and intuitively influence its operation is also defined.

*email: anton.vandenhengel@adelaide.edu.au
This work was funded by ARC Discovery Grant DP0558318, U.S. Air Force Research Laboratory Grant AOARD-074065 and EPSRC Grant EP/C006631/1(P).

- A method for optimising the fit of multiple models to a scene is described which takes into account the relationships between shapes, user interaction, and the fit of the shapes to the video.

Previous work has addressed the area of interactive scene modelling. The Facade system [Taylor et al. 1996] 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 [Robertson and Cipolla 2000] 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 [Sturm and Maybank 1999] 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 [Wilczkowiak et al. 2005]. 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 [Dick et al. 2004].

In the manipulation of 3D range data, model fitting is used to match either a pair of 3D point clouds [Arun et al. 1987] (and thereby estimating their relative pose and orientation) or a 3D point cloud and a pre-existing 3D model [Fisher et al. 1993]. In neither case is fitting performed on the basis of both 2D and 3D data.

The work described in this paper builds on previous work in two key ways: it uses a hierarchical model of shape, and it integrates user interaction in an intuitive and undemanding way, including the specification of relations between shapes. These features result in a 3D modelling system that takes advantage of scene properties to generate scene models extremely rapidly.

The hierarchical model of shape serves two purposes. It defines the spatial properties of the objects in the scene, and it defines a strategy for searching for these objects in the scene. When adding a new class of shape to the system, placing it in the hierarchy automatically defines its parameters and an algorithm for localising it. The shape hierarchy is described more fully in Section 2, and its role in sampling and fitting is described in Sections 4 and 3 respectively.

User interaction is an important factor in this system. The user specifies the class of each shape in the scene, and roughly where it is. This is done in an intuitive way, by drawing on the image data where the object to be modelled appears. The exact form of user interaction is not prescribed, and can vary from very slight (for instance clicking on an image to locate an object) to fine adjustment of the shape of an object, depending on the requirements of the user. Interaction is accepted at almost any stage in the modelling process, and it is accordingly described throughout this paper, although it is the focus of Section 6.

The ability to define relationships between shapes, such as adjacency or repetition, greatly improves the accuracy achievable while simultaneously reducing the user input required. These relationships are encoded as probabilistic constraints on the parameter vectors describing the related shapes. The power of these relationships lies in the fact they allow information both user and automatically generated information to be propagated throughout the scene model, exploiting each to the full extent possible. The specification of shape relations is the subject of Section 5.

2 The Shape Model

A shape model is defined by a parameter vector M. This vector describes the 3D position, orientation, shape and size of the model. It can contain the following:

- A position vector $\mathbf{T}$ (3-vector): the translation between the world and object coordinate system
- A rotation axis $\mathbf{U}$ (2-vector): the axis of rotation between world and object coordinates
- A rotation V (scalar): rotation about $\mathbf{U}$; together with $\mathbf{U}$ completely defines rotation between world and object coordinates
- Scale factors $S_1, S_2, ...$ (scalar): define a distance, scale or size.

The simplest model is a single 3D point, whose parameter vector is just the 3D position vector $\mathbf{T}$. A typical reconstruction produced by a feature based structure and motion algorithm is a collection of these models. By incrementally adding parameters to this model, we define a hierarchy of richer shapes, as shown in Figure 2.

By adding a scale factor S_1 to the point model we arrive at a sphere model: $M = \{\mathbf{T}, S_1\}$. By adding another scale parameter to the sphere: $M = \{\mathbf{T}, S_1, S_2\}$, we model a pair of concentric spheres. This is not a generally useful model, and so we disregard this branch of the tree. Adding an axis of rotation to the sphere model ($M = \{\mathbf{T}, S_1, \mathbf{U}\}$) we model a cylinder of infinite length and diameter S_1. Adding another scale factor ($M = \{\mathbf{T}, S_1, \mathbf{U}, S_2\}$) allows us to specify a length and thereby model a cylinder, or indeed other surfaces of revolution such as an ellipsoid, a torus or a cone. Adding further scale factors, we can model surfaces of revolution that are increasingly complex (asymmetrical).

Adding a full rotation to the sphere model ($M = \{\mathbf{T}, S_1, \mathbf{U}, V\}$), we specify a polyhedron that is symmetric about its centre—for example, a cube, tetrahedron, or dodecahedron. Adding another scale parameter to this model ($M = \{\mathbf{T}, S_1, \mathbf{U}, V, S_2\}$), we can model a slightly less symmetric polyhedron, such as a prism with square or triangular ends. Adding a further scale factor ($M = \{\mathbf{T}, S_1, \mathbf{U}, V, S_2, S_3\}$), we can model a general cuboid or any regular polyhedron with different scales along its 3 axes. Adding further scale factors, we can model increasingly complex polyhedra, such as cuboid with rounded corners, or a house with a roof, analogous to the case for surfaces of revolution.

We now return to the point model at the root of the shape hierarchy. By adding an axis of rotation $\mathbf{U}$ to this model, we define a plane whose normal is that axis: $M = \{\mathbf{T}, \mathbf{U}\}$. By adding a scale factor to this model, we have a planar circle ($M = \{\mathbf{T}, \mathbf{U}, S_1\}$). If we fully define the plane's coordinate system by including the final rotation parameter ($M = \{\mathbf{T}, \mathbf{U}, S_1, V\}$) we can define any symmetrical planar shape, such as an equilateral triangle or a square. Adding further scale factors, we can model increasingly complex planar shapes, by analogy with the polyhedra. A scale factor can also define a displacement normal to the plane, thereby creating a 3D shape such as those discussed previously. See the links in the tree.

Although this may not seem the most intuitively obvious arrangement of shapes and choice of parameters, it has been chosen because it meshes with a strategy for sampling each shape in the scene, as will be described in Section 4. Note also that order matters. And it divides shapes into 3 types: surface of revolution, polyhedron, and planar shape, but hypothesised shapes can switch between these categories.

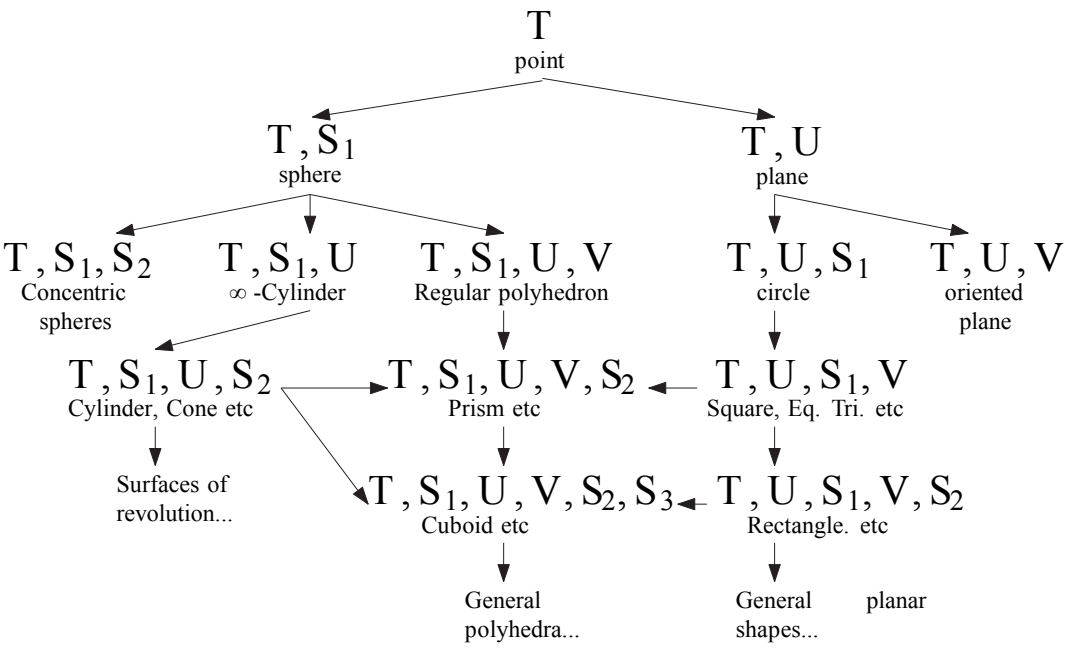

Figure 2: *The shape hierarchy.*

Each modelled object represents an instance of a shape, and associated with each instance is a label describing class of represented, such as 'cube', and a corresponding parameter vector, such as M for example. By a slight abuse of notation we also refer to the object itself by this label (M in this case).

3 Problem formulation: single shape

Recall that our goal is to model a scene as a collection of shapes. To do this we need a mechanism for finding a shape model that fits the scene well given the data (we focus on a single shape for now). More precisely, we wish to find a shape model M that, given image, camera and sparse 3D point data D and any prior information I (more later on this), maximises the posterior probability

$$\Pr(M|DI) \propto \Pr(D|MI)\Pr(M|I). \quad (1)$$

The posterior depends on likelihood $\Pr(D|MI)$ and prior $\Pr(M|I)$ terms, which are explained in the following sections.

3.1 Likelihood

The likelihood of the image data given a shape model 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 $-\log\Pr(D|MI)$ is measured by the weighted distance to local intensity gradient maxima [van den Hengel et al. 2006].

This likelihood function is based on image data and is therefore extremely sensitive and difficult to optimise. To initialise the optimisation, we make use of the sparse 3D data generated as a by-product of the initial structure from motion algorithm. Becuase this 3D data is based on corresponding image features, it is likely to occur in regions near the edges and corners of objects.

For each 3D point, we evaluate a distance to the shape model which is closely related to the reprojection error (after all, the 3D points were generated by bundle adjustment, which minimises this error). 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 Im as $\mathbf{p}(\mathbf{P}, Im)$ then we wish to measure the distance between $\mathbf{p}(\mathbf{P}, Im)$ and $\mathbf{p}(\mathbf{P}_M, Im)$ in each of the images that were used in the estimation of $\mathbf{P}$. The distance in image Im is

$$d_2(\mathbf{p}(\mathbf{P}, Im), \mathbf{p}(\mathbf{P}_M, Im)) \quad (2)$$

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 [Huber 1964] $h(\cdot)$ 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 $h(d_2(\mathbf{p}(\mathbf{P}, Im), \mathbf{p}(\mathbf{P}_M, Im)))$. The model parameters corresponding to the 3D shape that minimises this measure are used to initialise the subsequent image based optimisation.

3.2 Prior distribution

Clearly, the problem of finding shapes in the scene is impractical without the use of prior information. One option would be to specify limits on the ranges of shape parameters, and likely combinations of shape parameters. However this limits the range of scenes to which the system applies. We instead make use of user interaction to provide prior information about shape parameters. This works well in practice because the kind of global prior information about the scene that is required is exactly the information that is easy for a human observer to provide.

There are some things in a scene that are obvious to a human observer but surprisingly difficult to infer computationally. For example, a user can easily click on a prominent object in a scene. This

defines a ray in scene space, effectively constraining the object position in 2 dimensions. If instead of clicking the user draws a shape on the image, some idea of scale can also be determined. Interactions are selected for particular classes of shape on the basis of the amount of prior information required to guide the fitting process in practice and the ease with which this information may be provided by the user.

4 Shape sampling

In accordance with the Bayesian formulation of modelling, we seek a sampling strategy to characterise the posterior distribution $\Pr(M|DI)$. The dimension of the distribution depends on the number of parameters in the shape model M, but is usually too high to effectively sample directly. Additionally, the posterior is expected to have many local extrema due to the complexity of image data and sparseness of the 3D data.

To counter these problems, we sample from a succession of marginal posteriors. Each marginal is based on a subset of the shape parameters. The choice of shape parameters is related to the shape hierarchy defined in Section 2. For example, consider the cube model, $M = \{\mathbf{T}, S_1, \mathbf{U}, V\}$. By marginalising over the orientation paramters $\mathbf{U}$ and V, we return to the sphere model which has only 4 parameters. Thus, instead of sampling from the joint parameter space for a cube (7 dimensions), we can sample initially from a marginal distribution parameterised by position and scale:

$$\begin{aligned}\Pr(\mathbf{T}S_1|DI) &= \int_{\mathbf{U}}\int_{V} \Pr(\mathbf{T}S_1\mathbf{U}V|DI)\, dV d\mathbf{U} \\ &= \int_{\mathbf{U}}\int_{V} \Pr(\mathbf{T}S_1|\mathbf{U}VDI)\Pr(\mathbf{U}V|DI)\, dV d\mathbf{U} \quad (3)\end{aligned}$$

This equation can be directly evaluated for a given value of $\mathbf{T}$ and S_1 by summing probabilites over all orientations. As an approximation to this, we can instead evaluate the probability of the sphere with parameters $\mathbf{T}$ and S_1. This does not require integration and is therefore much faster.

Having sampled $\mathbf{T}$ and S_1 in this manner, the samples are locally optimised to reflect local modes in the marginal distribution. We can then sample for the remaining cube parameters (i.e. orientation) from the following conditional distribution:

$$\Pr(\mathbf{U}V|\mathbf{T}S_1DI) \quad (4)$$

based on the values of $\mathbf{T}$ and S_1 sampled previously. This strategy of sampling from marginal distributions followed by conditional distributions applies to any shape in the hierarchy. Furthermore, new shapes when added to the hierarchy already have an associated sampling strategy defined by their ancestors in the hierarchy. Note that integrating directly over scale factors is also possible because they have a constrained range. As well as the scale of the bounding box containing the scene, each scale factor in a shape model is constrained to be smaller than its predecessors.

For example, consider the cube model, with 7 parameters. In the hierarchy it is descended from the sphere model with 4 parameters, which is in turn descended from the point model with three. This suggests the following strategy. First, define a distribution over the scene space based on the point model (we assume a bounding box around the scene). This distribution is based on the distance to points in the neighbourhood. We then sample spheres in the scene space, based on the distribution for $\mathbf{T}$ derived from the point distribution. Finally, we sample cubes in the scene space based on the distribution for $\mathbf{T}$ and S resulting from sphere sampling. In this way, we obtain a set of cube samples, while never needing to sample more than 3 parameters at once. The idea is to define a sequence of sampling steps in subspaces of the full parameter space, to make sampling high parameter models feasible.

In general, a strategy for sampling a model can be derived by sampling its ancestors in the tree. Some shapes have multiple ancestries, and this is reflected in the fact that they can be sampled in multiple ways. The best way might depend on the nature of the shape. For example, a cuboid might be almost cubic—in which case the sphere route is best—or it might be an almost flat planar surface like a tabletop—in which case the planar route is suitable—or it might be long and thin like a table leg, in which case the cylinder path is appropriate. A nice feature of this formulation is that the most effective path chooses itself as part of the sampling process. For example, when sampling for a cuboid, we begin by sampling separately for planes and spheres. If one of these shapes does not give a positive overall result, we can forego sampling its children and instead sample only from the other's children. Thus a tabletop is likely to be found by sampling planes, whereas a table leg will be found by sampling cylinders.

5 Shape relations

So far, we have considered each shape model in a scene in isolation. In doing so, we ignore the highly constrained nature of many scenes, in which the position of each object is strongly dependent on the positions of others. For example, the positions of a collection of objects resting on a table all depend on the position of the table. We encode these constraints probabilistically. To do so, we first define a graph where each object in the scene is a node in that graph. Objects that are related are linked via a relation node R that expresses how they are dependent.

Fitting models to the scene is now a matter of optimising a joint probability over all models. We now have a set of models $\mathcal{M} = \{M_i \ : i = 1 \ldots N\}$, some data D (images, camera parameters and 3D points) and prior information I based on user interaction, shape relations and priors on individual shape parameters. Our goal is once more to maximise the posterior probability of the models given the data and prior: $\Pr(\mathcal{M}|DI)$.

As the parameters of separate shape models are independent except where linked by a relation, we represent this problem graphically. Each shape model is a node in the graph, and is linked to an observation node. Shape models are linked via relation nodes which express their dependency, as illustrated in Figure 3.

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 relations [Besag 1974]. The cliques in this case are all of size 2 because of the tree structure of the scene graph. The potential function adopted for the cliques containing an observed node and a model node is the unnormalised posterior $\Pr(M|DI) \propto \Pr(D|MI)\Pr(M|I)$. The potential function for cliques representing object relationships is 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 relationships $\mathcal{R}$ given the data set D and the prior information I is thus

$$\begin{aligned}\Pr(\mathcal{M}|DI) &= \frac{1}{Z}\prod_{M\in\mathcal{M}} \Pr(D|MI)\Pr(M|I) \prod_{R\in\mathcal{R}_M} \Pr(M,R) \\ &= \frac{1}{Z}\prod_{M\in\mathcal{M}} \Pr(D|MI)\Pr(M|I) \prod_{R\in\mathcal{R}_M} \Pr(M|R)\Pr(R) \quad (5)\end{aligned}$$

The set $\mathcal{R}_M$ represents the set of object-group relationships involving M, and the scalar Z a constant chosen such that $\Pr(\mathcal{M}|DI)$ integrates to 1.

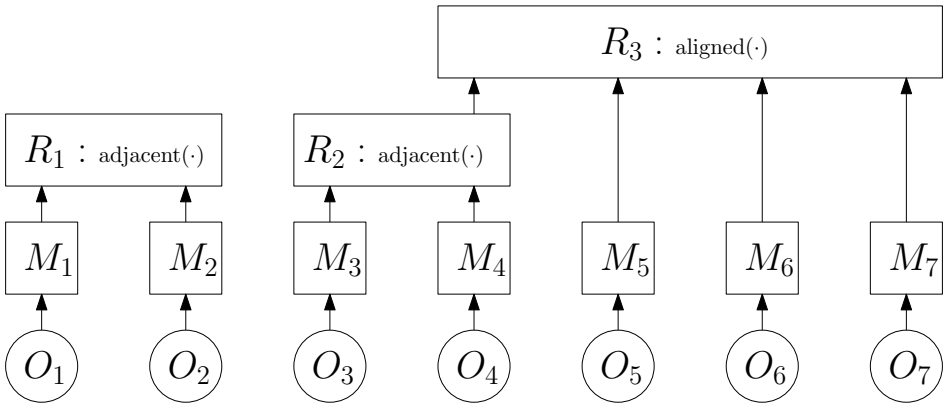

Figure 3: *Example scenegraph. Observation nodes O_i are linked to corresponding model nodes M_i. Models are linked by one or more relations R_i.*

The prior and likelihood functions for a single shape have been discussed in Section 3. The conditional relation PDF $\Pr(M|R)$ is defined differently for each type of relation. For example, adjacency is a common inter-object relation, expressing the constraint that two objects contain a face that coincides. Like user interaction, this constraint is expressed probabilistically, as a PDF linking the parameters of the two shape models involved. In the case of adjacency, this PDF is defined in terms of the distance between the nearest pair of faces in each object.

Relations can apply to more than just a pair of objects. For example, regularity and repetition is often a feature of urban and man-made environments, such as windows on a building, a set of steps, and so on. In this case a relation might express the height of a row of windows, and apply to all windows in that row. However each window in the row is still only part of a pairwise relation with the relation node. The PDF describing each window's probability is based on the difference between the height of that window and the height stored in the relation node.

6 Putting it all together

We now demonstrate how the terms from previous sections are incorporated into a problem formulation, and how this problem is solved. The process of modelling a scene is an iterative one, in which computation is driven by user interaction. In fact it involves a series of Bayesian estimation problems, in which the set of model parameters may be different each time. It proceeds in general as follows (see also Figure 4):

1. User highlights an object in the scene by clicking or drawing on an image.
2. The parameters of this object are optimised (see Section 4).
3. User highlights another object or adjusts an existing one, or specifies a link between two or more existing objects.
4. The parameters of all known objects are optimised as in Section 5. Further shapes belonging to a relation are found automatically if required.
5. Return to step 3.

This process continues until the scene is modelled with the fidelity required by the user. We now illustrate how this works in practice, with a couple of examples.

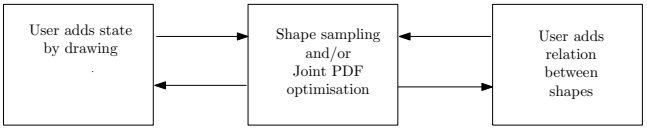

Figure 4: *Steps in interactive modelling.*

6.1 Example A - University Forecourt

To test this modelling approach, a video was taken of the front courtyard of the Unviersity of Adelaide. Frames from the sequence are shown in Figure 5. The video was taken with a handheld camcorder whose properties were unknown, as was its motion.

A sparse 3D reconstruction of the scene was obtained along with camera calibration information using an existing structure from motion technique [Thormählen 2006]. This reconstruction was sufficient to begin the interactive modelling process.

The first step in this process is that the user identifies a part of the scene by drawing on an image. In this case the user draws on the ground plane, which results in a plane being fitted to the 3D points whose projections lie within the drawn curve and to other points that are approximately coplanar with them. Fitting occurs by maximising the single shape posterior, as described in Section 3, where the prior is defined by the user's interaction and the likelihood is optimised first by fitting to the available 3D data, and then by fitting to the images. The fitting is fast enough that the user sees the fitted plane, which is superimposed on the image, instantly after drawing on the image.

Having established one part of the scene, further objects in it can be modelled and linked to existing parts. The user next draws on one of the bollards in the scene, as shown in Figure 6. This begins a sampling process as described in Section 4, where cuboids are sampled for in the volume of space defined by the user's interaction, first by sampling spheres, and then cubes and then cuboids. The cuboid samples that have the highest likelihood are then optimised—this often results in nearby samples resolving into a single solution, as is the case in this example. The resulting sample is fitted to the image, but because there are so few 3D points on this structure, the estimate can be very inaccurate. However by constraining the bollard to lie on the ground plane, we are able to obtain an accurate fit. This fitting occurs by maximising the joint probability (Equation 5).

Multiple instances of the bollard can be modelled without explic-

Figure 5: *(a) Frame from sequence. (b) User selects region of ground plane. (c) The fitted plane.*

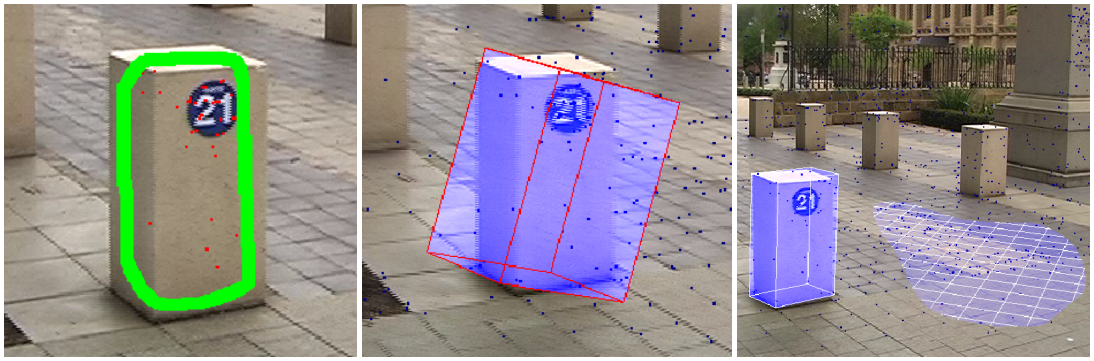

Figure 6: *(a) Drawing on the bollard. (b) Initial fit. (c) After constraining to lie on plane.*

itly modelling each one individually. By specifying a repetition relation, the user can specify that the bollard that has already been modelled is repeated at regular intervals. This is done by dragging the pointer along the axis of repetition in the image as shown in Figure 7. Note that although a 3D direction is being specified, because it is anchored to the ground plane, a 2D interaction (dragging on the image) is sufficient. This turns out to be a much more intuitive way of modelling than by faking a 3D interface. As each shape is added to the model, the joint probability is once again optimised over all shapes. This means that the number of repeated shapes, and the spacing between them, does not need to be specified manually. They are determined as those which maximise the probability function 5.

6.2 Example B - Art Gallery

The art gallery scene shown in Figure 8 contains similar constraints to the previous example, but a slightly different set of shape primitives: planes, cuboids and cylinders. The planes and cuboids are sampled and optimised as described in the previous section. The position of the pillar nearest to the camera is initialised by a user selecting an area on it. This is used to initialise a set of sphere samples which are then combined to form a cylinder sample. Once fitted, this shape is replicated as before.

7 Conclusion

The system is still work in progress but shows promise as a means of quickly and intuitively modelling real world scenes. Future work involves the inclusion of a wider range of shapes and constraints. When choosing which shape and shape relation primitives to include, there is often a tradeoff between flexibility and the amount of user interaction required. The inclusion of more shapes also increases the difficulty of sampling to find the best fitting model. We plan to investigate ways in which sampling can be optimised; one promising approach is to use context to constrain the sample space.

We have also been working on more general means of interaction via sketching on video [van den Hengel et al. 2007]. This has proved to be a fast and flexible means of acquring models of low to moderate complexity, and as a way to add detail to the types of models described in this paper.

References

ARUN, K., HUANG, T., AND BLOSTEIN, S. 1987. Least-squares fitting of two 3-d point sets. *IEEE Trans. Pattern Analysis and Machine Intelligence 9*, 5, 698–700.

BAKER, S., SZELISKI, R., AND ANANDAN, P. 1998. A layered approach to stereo reconstruction. In *Proc. IEEE Computer Vision and Pattern Recognition*, 434–441.

BASTIAN, J., AND VAN DEN HENGEL, A. 2005. Computing surface-based photo-consistency on graphics hardware. In *Proceedings of Digital Image Computing: Techniques and Applications,Cairns, Australia*, 249–258.

BEARDSLEY, P. A., ZISSERMAN, A., AND MURRAY, D. W. 1997. Sequential update of projective and affine structure from motion. *International Journal of Computer Vision 23*, 3, 235–259.

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

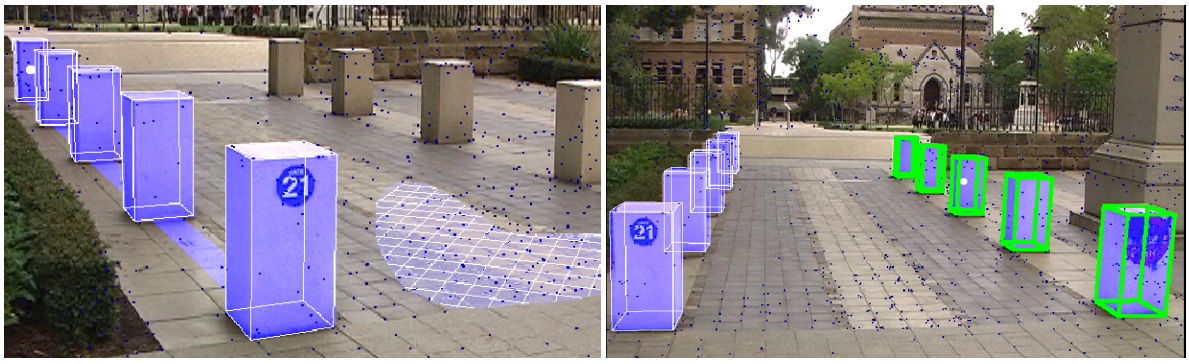

Figure 7: *(a) Replicating the bollard by dragging the mouse. (b) Replicating a row of bollards.*

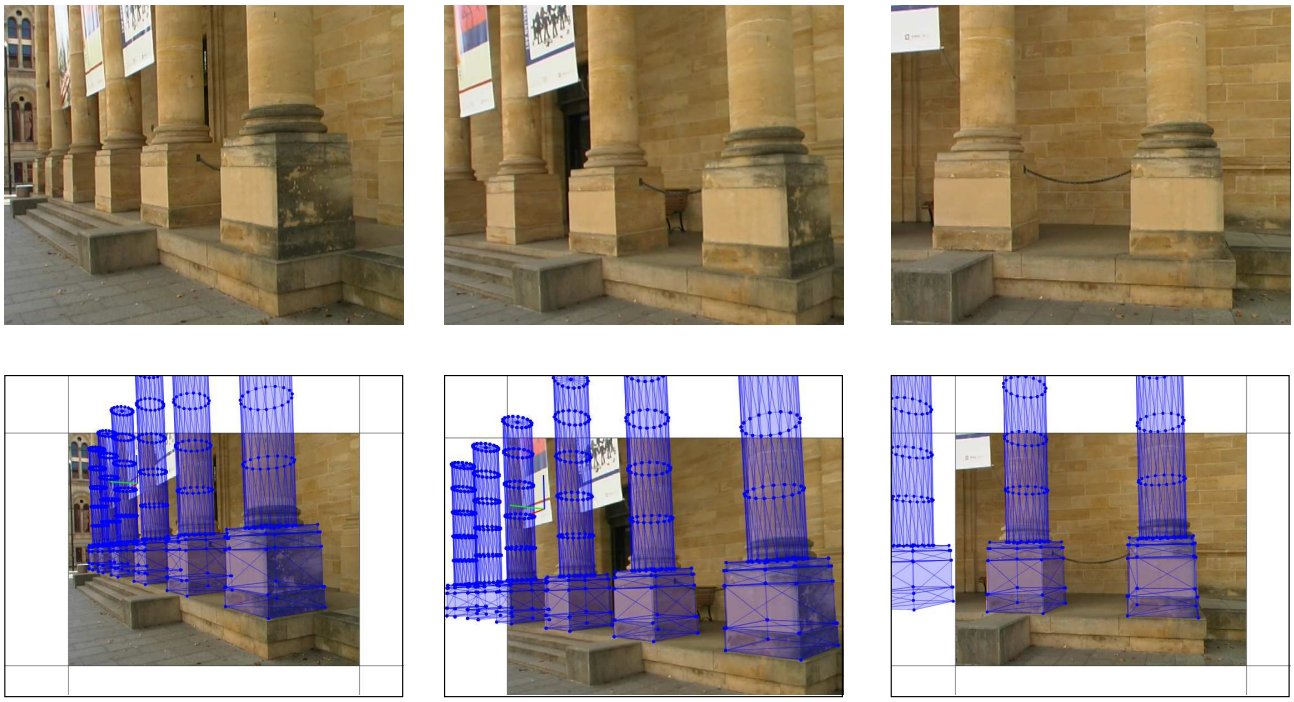

Figure 8: *Selected frames from the art gallery sequence, and the corresponding views of the final model.*

BORGEFORS, G. 1986. Distance transformations in digital images. *Journal of Computer Vision, Graphics and Image Processing. 34*, 344371.

CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *International Journal of Computer Vision 40*, 2, 123–148.

DAUCHER, N., DHOME, M., LAPRESTE, J. T., AND RIVES, G. 1993. Modelled object pose estimation and tracking by monocular vision. In *BMVC93*, 249–258.

DICK, A., TORR, P., AND CIPOLLA, R. 2004. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision 60*, 2 (November), 111–134.

DRUMMOND, T., AND CIPOLLA, R. 2000. Application of lie algebras to visual servoing. *International Journal of Computer Vision 37*, 1, 21–41.

EOS SYSTEMS, 2005. Photomodeler: A commercial photogrammetry product http://www.photomodeler.com.

FERRYMAN, J. M., WORRALL, A. D., SULLIVAN, G. D., AND BAKER, K. D. 1995. A generic deformable model for vehicle recognition. In *Proceedings British Machine Vision Conference*, 127–136.

FISHER, R., FITZGIBBON, A., WAITE, M., TRUCCO, E., AND ORR, M. 1993. Recognition of complex 3-d objects from range data. In *Proc. 7th International Conference on Image Analysis and Processing*, 509–606.

GIBSON, S., COOK, J., HOWARD, T., AND HUBBOLD, R. 2002. ICARUS: Interactive reconstruction from uncalibration image sequences. In *ACM Siggraph 2002 Conference Abstracts and Applications*.

GIBSON, S., HUBBOLD, R., COOK, J., AND HOWARD, T. 2003. Interactive reconstruction of virtual environments from video sequences. *Computers & Graphics 27*, 2 (April), 293–301.

HARTLEY, R. I., AND ZISSERMAN, A. 2000. *Multiple View Geometry*. Cambridge University Press.

HUBER, P. 1964. Robust estimation of a location parameter. *Annals of Mathematical Statistics 35*, 73–101.

JIN, H., FAVARO, P., AND SOATTO, S. 2000. Real-time 3-d motion and structure from point features: a front-end system for vision-based control and interaction. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

KULLBACK, S., AND LEIBLER, R. A. 1951. On information and sufficiency. *Annals of Mathematical Statistics 22*, 1 (March), 79–86.

MURPHY, K. P., WEISS, Y., AND JORDAN, M. I. 1999. Loopy belief propagation for approximate inference: An empirical study. In *Uncertainty in Artificial Intelligence (UAI), Proceedings of the Fifteenth Conference*, Morgan Kaufmann.

NISTÉR, D. 2003. Preemptive ransac for live structure and motion estimation. In *Proceedings IEEE International Conference on Computer Vision*.

POLLEFEYS, M., GOOL, L. V., VERGAUWEN, M., VERBIEST, F., CORNELIS, K., TOPS, J., AND KOCH, R. 2004. Visual modeling with a hand-held camera. *International Journal of Computer Vision 59*, 3, 207–232.

ROBERTSON, D., AND CIPOLLA, R. 2000. An interactive system for constraint-based modelling. In *Proc. 11th British Machine Vision Conference*, 536–545.

SHI, J., AND TOMASI, C. 1994. Good features to track. In *Proc. IEEE Computer Vision and Pattern Recognition*, 593–600.

SHUM, H., HAN, M., AND SZELISKI, R. 1998. Interactive construction of 3d models from panoramic mosaics. In *Proc. IEEE Computer Vision and Pattern Recognition*, 427–433.

SMITH, P., DRUMMOND, T., AND CIPOLLA, R. 2000. Motion segmentation by tracking edge information over multiple frames. In *ECCV (2)*, 396–410.

STURM, P., AND MAYBANK, S. 1999. A method for interactive 3d reconstruction of piercewise planar objects from single images. In *Proc. 10th British Machine Vision Conference*, 265–274.

TAYLOR, C., DEBEVEC, P., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. *ACM SIGGraph, Computer Graphics*, 11–20.

THE PIXEL FARM. PFTRACK: A commercial camera tracking and image based modelling product http://www.thepixelfarm.co.uk.

THORMÄHLEN, T. 2006. *Robust estimation of camera motion from image sequences*. PhD thesis, University of Hannover.

TORR, P. H. S. 1997. An assessment of information criteria for motion model selection. In *Proc. IEEE Computer Vision and Pattern Recognition*, 47–52.

TRIGGS, B. 1997. Autocalibration and the absolute quadric. In *Proc. IEEE Computer Vision and Pattern Recognition*, 609–614.

VAN DEN HENGEL, A., DICK, A., THORMAEHLEN, T., TORR, P. H. S., AND WARD, B. 2006. 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*.

VAN DEN HENGEL, A., DICK, A., THORMÄHLEN, T., WARD, B., AND TORR, P. 2007. Videotrace: Rapid interactive scene modelling from video. In *ACM SIGGraph 2007*.

VIOLA, P., AND JONES, M. 2001. Rapid object detection using a boosted cascade of simple features. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*.

WALTZ, D. 1971. Understanding line-drawings of scenes with shadows. *Artificial Intelligence 2*, 79–116.

WILCZKOWIAK, M., STURM, P., AND BOYER, E. 2005. Using geometric constraints through parallelepipeds for calibration and 3d modeling. *IEEE Trans. Pattern Analysis and Machine Intelligence 27*, 2 (February), 194–207.