Modelling and Interpretation of Architecture from Several Images

A. R. Dick[1*] P. H. S. Torr[2] R. Cipolla[1]

[1] Department of Engineering, University of Cambridge, Cambridge, CB2 1PZ, UK

[2] Department of Computing, Oxford Brookes University, Wheatley, Oxford, OX33 1HX, UK

ard@cs.adelaide.edu.au philiptorr@brookes.ac.uk cipolla@eng.cam.ac.uk

Abstract

This paper describes the automatic acquisition of three dimensional architectural models from short image sequences. The approach is Bayesian and model based. Bayesian methods necessitate the formulation of a prior distribution; however designing a generative model for buildings is a difficult task. In order to overcome this a building is described as a set of walls together with a 'Lego' kit of parameterised primitives, such as doors or windows. A prior on wall layout, and a prior on the parameters of each primitive can then be defined. Part of this prior is learnt from training data and part comes from expert architects. The validity of the prior is tested by generating example buildings using MCMC and verifying that plausible buildings are generated under varying conditions. The same MCMC machinery can also be used for optimising the structure recovery, this time generating a range of possible solutions from the posterior. The fact that a range of solutions can be presented allows the user to select the best when the structure recovery is ambiguous.

Keywords: Architectural modelling, structure and motion, object recognition

*Current address: School of Computer Science, University of Adelaide, Adelaide SA 5005, Australia

1 Introduction

This paper describes a system for processing several (2–6) images of an architectural scene to produce a 3D model of the scene in which various architectural components have been identified. An example set of input images is shown in Figure 1, along with the labelled 3D model which is our goal.

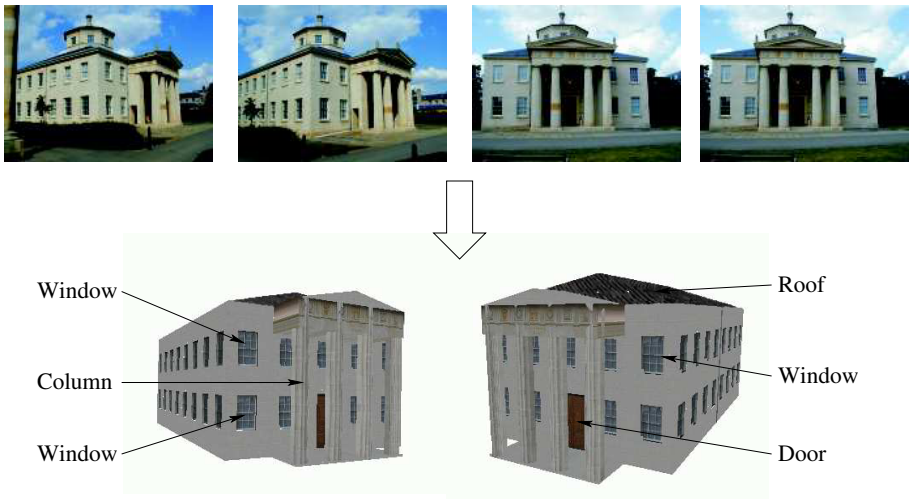


Figure 1: *The problem addressed in this paper. A labelled 3D model is generated from several images of an architectural scene, in this case the Downing College Library, Cambridge.*

This problem straddles two classic problems of computer vision: structure from motion and object recognition. Although they have different goals, these two problems are clearly related: the structure of an object is a strong clue to its identity, and conversely the identity of an object can inform an estimate of its structure. As neither problem has been completely solved, there is a case for techniques which use the relationship between them to improve upon methods which address only structure from motion or object recognition.

A probabilistic Bayesian framework is used for combining information about structure and identity. A prior distribution is defined for the parameters of the

model, and its validity is tested by simulating draws from it and verifying that it does indeed generate plausible buildings under varying conditions. A practical algorithm is then developed to find the maximum a posteriori (MAP) model parameters based on data likelihood and prior distributions. The algorithm is tested on a variety of architectural image sequences.

In Section 2 we review some relevant previous work before proceeding to lay out our own framework in Section 3. Following this an algorithm based on our framework is described in Section 4 along with results and analysis in Section 5.

1.1 Notation

The model developed in this paper requires a substantial amount of notation, summarised below:

Probabilistic symbols

$\mathbf{D}$	Data
$\mathbf{M}$	Model to be estimated
$\boldsymbol{\theta}$	Vector of parameters belonging to $\mathbf{M}$
$\mathbf{I}$	Prior information
$\Pr(\boldsymbol{\theta})$	Probability of parameter value(s) $\boldsymbol{\theta}$
$\Pr(\boldsymbol{\theta}\|\mathbf{I})$	Conditional probability of $\boldsymbol{\theta}$ given $\mathbf{I}$
$G(\mu, \sigma)$	Gaussian distribution with mean μ and standard deviation σ.
$U(a, b)$	Uniform distribution bounded by a and b.

Architectural model parameters

$\boldsymbol{\theta}_G$	Global style parameters
n_W	Number of walls
$\boldsymbol{\theta}_P$	Wall plane parameters $= \{w_P,\ h_P,\ \mathbf{n},\ \mathbf{p}\}$
w_P	Wall plane width
h_P	Wall plane height
$\mathbf{n}$	Wall plane normal
$\mathbf{p}$	Wall plane position
$\boldsymbol{\theta}_W$	Wall parameters $= \{n_P,\ \boldsymbol{\theta}_L,\ \boldsymbol{\theta}_S,\ \boldsymbol{\theta}_T\}$
n_P	Number of primitives
$\boldsymbol{\theta}_L$	Label parameters
$\boldsymbol{\theta}_S$	Shape parameters
$\boldsymbol{\theta}_T$	Texture parameters

2 Related work

There is a wealth of literature on both structure and motion estimation (review appears in [28] for example) and object recognition (for example [31]).

2.1 Structure and motion systems

Structure and motion systems often represent structure using simple primitives such as points and lines. One such algorithm is the *factorisation* method of Tomasi and Kanade [38]. The advantage of this method is that it is linear and thus is guaranteed to find the optimal estimate of structure and motion without initialisation.

In the case of non-linear camera models, a non-linear optimisation routine is used to minimise the reprojection error of the reconstructed points. Originally formulated by photogrammetrists, bundle adjustment [35, 42] is an optimisation framework in which the total distance between the detected image features and the projections of the reconstructed points is minimised. This optimisation can be performed simultaneously over the unknown 3D point positions and camera calibration parameters.

The advantage of representing structure using 3D points and/or lines is their generality, and the fact that they can be detected automatically, reliably and quickly in a wide variety of images [7, 20]. However in real world scenes, and particularly urban scenes, structure often appears in the form of smooth surfaces and solid objects, rather than as a disparate collection of points and lines. One way to encode this is to explicitly model structure as a set of bounded surfaces, or *layers* [43, 3, 40].

However layers are not an ideal representation for architecture. Although they are based on a parameterised 3D surface, their shape, and depth and texture maps if they exist, are not generally parameterised but rather represented

explicitly as a set of boundary/depth/texture values defined over the entire surface. This ignores a vast amount of information about the simplicity and regularity of shape in architectural scenes.

2.1.1 Architectural modelling systems

Recently, there has been a great deal of interest in structure and motion systems which are designed specifically for architectural scenes. Because these systems are designed for architectural modelling, they can use a more specialised representation of structure which is particularly suited to architecture. For example, a system developed as part of the IMPACT project for reconstruction of urban scenes from aerial images [1, 33] uses a polygonal structure representation. This is more constrained than the layer representation because it imposes a polygonal model of plane shape—all planes are bounded by straight lines.

On a larger scale, the City Scanning project at MIT[1] aims to automatically model extensive urban environments (such as the MIT campus) using several thousand calibrated images. The algorithm used by this system uses many architectural heuristics, such as the prominence of parallelism, and horizontal and vertical lines, but because of the vagueness of these heuristics requires an enormous amount of image data (570 mosaics, each consisting of 47 1012×1524 images) to model about 10 fairly simple buildings over an area of the MIT campus, according to their website. The general approach taken by this system to overcome image ambiguity is to use a combination of structural heuristics, a large amount of redundant data, and information about camera pose and calibration, whereas we advocate the use of more specific prior knowledge about architecture.

Façade [37] is an interactive system which generates highly realistic models of architectural scenes from a small number of images. Model structure is defined

[1] http://city.lcs.mit.edu

in terms of volumetric primitives such as rectangular and triangular prisms, which can be described with only a few parameters, such as their height, width and depth. Using volumetric primitives means that a building can be modelled using a relatively small number of parameters, which enables a complete model to be obtained from a small number of images. However such primitives are difficult to detect automatically, and hence Façade requires that a human operator specifies each of the blocks in the model and their spatial relationships, and manually registers them in each image. The models produced are extremely convincing, indicating that such specialised or "high-level" primitives can model architectural scenes very well.

2.2 Model-based object recognition

Despite using high-level primitives to represent architecture, the models produced by structure and motion systems have no semantic content. A model obtained using Façade, for instance, may contain a rectangular block of certain dimensions. What this block actually represents, whether it is a door, a window or an entire building, is not specified in the model. This sort of information, i.e. an *interpretation* of the scene, is valuable as it allows reasoning or inference about the scene based on the type of objects which are *recognised* in it. This enables parts of the scene which are unseen in any image, or whose interpretation from image data is ambiguous, to be inferred from those parts of the model which can be interpreted with more certainty.

2.2.1 Modelling and recognising shape

The representation of shape, which was considered in Section 2.1 in the context of structure and motion estimation, also affects object recognition. To be effective for object recognition, a shape representation should have the following properties [25, 31]: it should be flexible enough to represent all required shapes;

it should uniquely identify each shape which needs to be distinguished; it should be stable, so that a small change to the shape gives rise to only a small change in its description; and it should permit efficient comparison with other shapes and with input data. These properties suggest that the ideal shape representation depends largely on the application domain: what sort of shapes need to be modelled, and what type of shapes need to be distinguished from one another. This suggests that a high level representation of shape, a representation which is specifically tuned for modelling architecture, will be more effective than a more general representation for the purpose of object recognition.

Shape can be represented in a 3D coordinate system [5, 15, 16, 24], as a set of characteristic 2D projected shapes (a.k.a. aspects) [23], or as a combination of the two [30]. The aspect representation, although it avoids the need for a 3D model of the shape, is not suited to modelling a complex object which has a large number of distinct aspects, thus increasing drastically the number of models to be considered for recognition [14].

Geons [5, 12] are an attempt at a general 3D shape representation. A geon is a volume defined by the qualitative properties of its axis (e.g. straight or curved) and a generalised cylinder about this axis (e.g. tapered or straight). These properties are designed so that each geon can be identified from a wide range of viewpoints. However recognising a general shape as a collection of geons remains a challenging problem; experiments in this field have been reported only for line drawings, 3D range data [6] and uncluttered images [30]. Other 3D shape representations, such as CSG, CAD wireframe models, and triangulated meshes, have also been investigated as a basis for recognition [15, 24]. Regardless of the representation of shape, there remains a fundamental tradeoff between the generality of objects that can be recognised and the ability to discriminate between objects. The best compromise is generally domain dependent.

2.2.2 Modelling and recognising texture

Texture is used to model parts of a scene which are too detailed to be modelled geometrically, and is therefore inevitably more complex and not so easily parameterised as shape. Rather than using a parametric model of texture appearance, a more popular approach is to learn a model from patches of similar texture in other images [29, 34, 36]. This is generally done by accumulating texture characteristics in one or more histograms which then approximate a prior model for unseen texture. A Laplacian pyramid [36] or wavelet transform [34] of the texture can be used to provide tolerance to changes in scale and global illumination. Having obtained a model for texture, unseen texture can be classified by assigning it the texture type whose model (histogram) it most resembles.

3 The architectural model

3.1 Definition of the model

An architectural scene is modelled as a collection of walls. Each wall is modelled as a rectangle, and contains a set of volumetric primitives corresponding to common architectural features such as doors and windows. In addition there is a set of global parameters $\boldsymbol{\theta}_G$ which contains features of the model pertaining to all walls, such as the style of the building. Thus an architectural model $\mathbf{M}$ contains parameters $\boldsymbol{\theta} = \{n_W, \boldsymbol{\theta}_W, \boldsymbol{\theta}_P, \boldsymbol{\theta}_G\}$ where n_W is the number of walls in the model, $\boldsymbol{\theta}_P = \cup_{i=1}^{n_W} \boldsymbol{\theta}_P^i$ defines the orientation, position and boundary of each wall plane, $\boldsymbol{\theta}_W = \cup_{i=1}^{n_W} \boldsymbol{\theta}_W^i$ are parameters specific to each wall in the model, and $\boldsymbol{\theta}_G$ are global parameters which apply to the entire model.

Each wall plane has width w_P, height h_P, normal $\mathbf{n}$ and position $\mathbf{p}$, thus $\boldsymbol{\theta}_P^i = \{w_P^i, h_P^i, \mathbf{n}^i, \mathbf{p}^i\}$. Each parameter vector $\boldsymbol{\theta}_W^i$ is further subdivided into type, shape and texture parameters: $\boldsymbol{\theta}_W^i = \{n_P, \boldsymbol{\theta}_L^i, \boldsymbol{\theta}_S^i, \boldsymbol{\theta}_T^i\}$, where n_P is the number of volumetric primitives contained in the wall. $\boldsymbol{\theta}_L = \cup_{j=1}^{n_P} \theta_L^j$ is an

identifier for the type of each primitive in the wall. These range from $\mathcal{M}_1$ to $\mathcal{M}_9$ and are listed in Table 1. $\boldsymbol{\theta}_S = \cup_{j=1}^{n_P} \boldsymbol{\theta}_S^j$ defines the shape of each primitive. The shape parameters used for each primitive are also listed in Table 1, and an example window primitive is shown in Figure 2.

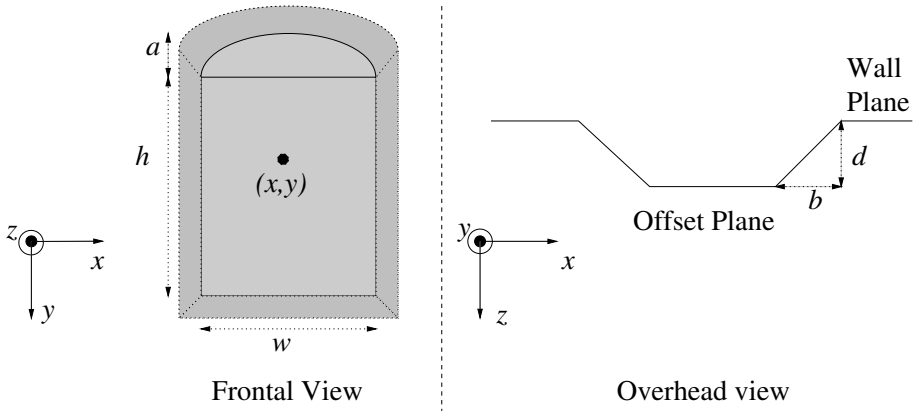


Figure 2: *Frontal and overhead views of a window or door primitive. The parameters are: position relative to the wall plane* (x, y)*, height* h*, width* w*, depth relative to the wall* d*, arch height* a*, bevel (slope of edges)* b*.*

The texture parameters $\boldsymbol{\theta}_T = \cup_{j=0}^{n_P} \theta_T^j$ are intensity variables $i(\mathbf{g})$ (between 0 and 255) defined at each point $\mathbf{g}$ on a regular 2D grid covering the model surface. θ_T^0 is the set of texture parameters belonging to the wall plane rather than a shape primitive attached to it.

The global parameter vector $\boldsymbol{\theta}_G$ contains a single style parameter which is set to be either Classical or Gothic. Examples of Classical and Gothic architecture are shown in Figure 3. In practice it is useful to classify Classical and Gothic styles according to the types of material used in their construction, and the styles of some of their components. This information is also included in the global parameter vector, as shown in Table 2.

The model is designed to be used with photographs of architecture taken from ground level. Therefore it models a level of detail consistent with these

Table 1: *Some primitives available for modelling architecture. Parameters in brackets are optional; a mechanism for deciding automatically whether they are used is given in Section 4.3.2. The parameters are defined as follows: x: x position; y: y position; w: width; h: height; d: depth; a: arch height; b: bevel (sloped edge); dw: taper of pillars, buttresses. The NULL model is simply a collection of sparse triangulated 3D points.*

θ_L^i	Description	θ_S^i
$\mathcal{M}_1$	Window	$x, y, w, h, d, (b), (a)$
$\mathcal{M}_2$	Door	$x, y, w, h, d, (b), (a)$
$\mathcal{M}_3$	Pediment	x, y, w, h, d
$\mathcal{M}_4$	Pedestal	x, y, w, h, d
$\mathcal{M}_5$	Entablature	x, y, w, h, d
$\mathcal{M}_6$	Column	x, y, w, h, d, dw
$\mathcal{M}_7$	Buttress	x, y, w, h, d, dw
$\mathcal{M}_8$	Drainpipe	x, w, h
$\mathcal{M}_9$	NULL	$x_1 ... x_n, y_1 ... y_n, z_1 ... z_n$

Figure 3: *Example of some primitives used to construct Gothic and Classical architecture.*

Table 2: *Values for the global parameter vector $\boldsymbol{\theta}_G$. These values are specified manually.*

Style	**Material**	**Window style**	**Column style**
Classical	Stone	Paned	Fluted
Gothic	Brick	Plain	Plain

viewpoints; for instance doors and windows are modelled in addition to the walls of each building. However finer levels of detail, such as the location of individual bricks, door handles and fine ornamentation are not modelled. Similarly little attention is paid to modelling roofs (in fact they are only modelled as a simple ridged structure), as most images taken from ground level include very little if any information about the roof structure.

3.2 Formulation of the problem

Having defined the parameters $\boldsymbol{\theta}$ comprising an architectural model $\mathbf{M}$, the problem of structure and motion, and interpretation of architectural scenes can now be defined. To solve these problems both the model $\mathbf{M}$ which best models the scene and its optimal parameters $\boldsymbol{\theta}$ are required. Thus we want to maximise the posterior probability $\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI})$, which can be decomposed via repeated applications of the product rule [22]:

$$\begin{aligned}
\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI}) \;&\propto\; \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\mathbf{M}\boldsymbol{\theta}\mathbf{I}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\boldsymbol{\theta}|\mathbf{MI})\Pr(\mathbf{MI}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\boldsymbol{\theta}_G\boldsymbol{\theta}_P\boldsymbol{\theta}_W|\mathbf{MI})\Pr(\mathbf{MI}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\boldsymbol{\theta}_W|\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_P\boldsymbol{\theta}_G|\mathbf{MI})\Pr(\mathbf{MI}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\boldsymbol{\theta}_W|\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_P|\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_G|\mathbf{MI})\Pr(\mathbf{MI})
\end{aligned} \tag{1}$$

It is assumed that the parameters of the primitives belonging to each wall are independent, and independent of the parameters of other wall planes. Thus $\Pr(\boldsymbol{\theta}_W|\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI}) = \prod_{i=1}^{n_W}\Pr(\boldsymbol{\theta}_W^i|\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})$. Decomposing the parameters $\boldsymbol{\theta}_W^i$ of each wall into their constituent shape, texture and layer parameters:

$$\begin{aligned}
\Pr(\boldsymbol{\theta}_W^i|\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI}) &= \Pr(\boldsymbol{\theta}_L^i\boldsymbol{\theta}_S^i\boldsymbol{\theta}_T^i|\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI}) \\
&= \Pr(\boldsymbol{\theta}_T^i|\boldsymbol{\theta}_L^i\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_S^i|\boldsymbol{\theta}_L^i\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_L^i|\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})
\end{aligned} \tag{2}$$

Note that the probabilities of $\boldsymbol{\theta}_T^i$ and $\boldsymbol{\theta}_S^i$ are independent as the probability distribution for the texture of a primitive is unaffected by its shape, and vice versa. Thus the fully expanded expression for the posterior is:

$$\begin{aligned}\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI}) \;\propto\; & [\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})\Pr(\boldsymbol{\theta}_P|\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_G|\mathbf{MI})\Pr(\mathbf{MI})] \\ \times \quad & \left[\prod_{i=1}^{n_W}\Pr(\boldsymbol{\theta}_T^i|\boldsymbol{\theta}_L^i\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_S^i|\boldsymbol{\theta}_L^i\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})\Pr(\boldsymbol{\theta}_L^i|\boldsymbol{\theta}_P^i\boldsymbol{\theta}_G\mathbf{MI})\right] \quad (3)\end{aligned}$$

Each term in (3) has an intuitive interpretation:

- $\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})$ is the likelihood of the images given a complete specification of the model. This is determined by the deviation of image intensities from the projection of the texture parameters, as described in Section 3.3.

- $\Pr(\boldsymbol{\theta}_P|\boldsymbol{\theta}_G\mathbf{MI})$ is the prior probability of the shape, position and orientation of each wall given global parameters such as the style of the building. A classical style building may for example specify a high prior probability for walls which are perpendicular to their neighbours and whose height to width ratio is near the "golden ratio". See Section 3.4 for more details.

- $\Pr(\boldsymbol{\theta}_G|\mathbf{MI})$ is the probability of the global parameters (Table 2), given a particular model. In general global parameters are set manually and this distribution is set to 1 for the preset value, and 0 elsewhere.

- $\Pr(\mathbf{MI})$ can express a prior preference for one model over another. This is not used and is uniform for all possible architectural models $\mathbf{M}$.

- $\Pr(\boldsymbol{\theta}_T|\boldsymbol{\theta}_L\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$ is a prior on the texture parameters. This is evaluated using learnt models of appearance, such as the fact that windows are often dark with intersecting mullions (vertical bars) and transoms (horizontal bars), or that columns contain vertical fluting. This is described in more detail in Section 3.6.

- $\Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$ is a prior on shape. This encodes prior knowledge of architectural style, for instance that windows in Gothic architecture are narrow and arched; it can also encode practical constraints, such as that doors generally appear at ground level. These priors are discussed in Section 3.5.

- $\Pr(\boldsymbol{\theta}_L|\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$ is the prior probability of each type of primitive. It may be used to specify the relative frequency with which primitive types occur, e.g. that windows are more common than doors, and is manually set to reflect the style of building being modelled (e.g. a Gothic building will have a high probability of buttresses where for a classical building columns are more likely).

Having briefly examined the function of each of these probability distributions, our next task is to consider how the likelihood, plane prior, shape prior and texture prior may be evaluated.

3.3 Evaluation of the likelihood

The likelihood $\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I})$ depends directly on the texture parameters $\boldsymbol{\theta}_T$. Recall that $\boldsymbol{\theta}_T$ is defined as a set of intensities $i(\mathbf{g})$ at regularly sampled points on the model surface; the estimation of the intensities is detailed in Section 4.3.4. The likelihood is evaluated by assuming that the projection of each texture parameter $i(\mathbf{g})$ into each image is corrupted by normally distributed noise $\epsilon \in G(0, \sigma_\epsilon)$, thus $i(\mathbf{g}^j) = i(\mathbf{g}) + \epsilon$, where $i(\mathbf{g}^j)$ is the projection of $i(\mathbf{g})$ into image j. Assuming that the error ϵ at each pixel is independent, the likelihood over all points $\mathbf{g}$ is written:

$$\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) = \prod_{\mathbf{g}} \prod_{j} \frac{1}{\sqrt{2\pi}\sigma_\epsilon} \exp -\frac{1}{2} \left(\frac{i(\mathbf{g}^j) - i(\mathbf{g})}{\sigma_\epsilon} \right)^2 \tag{4}$$

This formulation of the likelihood is simple to use, but it has a number of shortcomings. It assumes that all surfaces in the scene are Lambertian; that is, that

they emit the same colour and intensity of light regardless of the viewing angle. This is plausible for materials such as stone and plaster, but clearly not true for shiny materials such as glass. In practice it was found that although the Lambertian model is incorrect for windows, it is often acceptable because windows are often set into walls and therefore in shadow, appearing dark and non-reflective from any viewing angle. However there will inevitably be points which are outliers to this model, and to account for these the likelihood is represented as a mixture of the Gaussian distribution with a uniform distribution $U(0, 255)$:

$$\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) = \lambda \Pr*(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) + (1-\lambda)\frac{1}{256} \tag{5}$$

The mixture constant λ is typically set to around 0.9, as most points are inliers to the Gaussian model.

3.4 Evaluation of the wall plane prior

The prior on wall plane shape and position, $\Pr(\boldsymbol{\theta}_P|\boldsymbol{\theta}_G\mathbf{MI})$, is based on architectural heuristics such as the fact that walls are likely to intersect at right angles, and on practical constraints concerning the height to width ratio of walls. The plane parameters $\boldsymbol{\theta}_P$ can be divided into component parts whose prior distributions are independent:

$$\Pr(\boldsymbol{\theta}_P|\boldsymbol{\theta}_G\mathbf{MI}) = \Pr(\mathbf{n}|\boldsymbol{\theta}_G\mathbf{MI}) \Pr(\mathbf{p}|\boldsymbol{\theta}_G\mathbf{MI}) \Pr(\{h_P, w_P\}|\boldsymbol{\theta}_G\mathbf{MI}) \tag{6}$$

Each of these terms is represented by a simple distribution. A prior for $\mathbf{n}$ is based on the assumption that all planes should be perpendicular to their neighbours, and perpendicular to a common ground plane. First, a ground plane is estimated as the plane which best fits all wall plane normals. Then a prior is applied to the projection of each normal onto the ground plane:

$$\Pr(\mathbf{n}|\boldsymbol{\theta}_G\mathbf{MI}) \alpha \sum_{i=1}^{n-1} \left(\frac{\phi_i - \pi/2}{\sigma_\phi} \right)^2 \tag{7}$$

where ϕ_i is the interior angle between the projections of the normals to planes i and $i+1$ onto the ground plane. [2]

A uniform prior is applied to the plane position parameters $\mathbf{p}$, expressing no prior knowledge of the spatial location of the planes other than that they occur in the field of view of the cameras.

The prior on height and width for each plane is given by

$$\begin{aligned} \Pr(\{h_P, w_P\}|\boldsymbol{\theta}_G\mathbf{MI}) &= 1, \text{if } 0.25 \leq h_P/w_P \leq 4 \\ &= 0, \text{otherwise} \end{aligned} \tag{8}$$

Note that because a reconstruction is generally defined only up to scale, a prior is applied to the ratio of the height and width values rather than to their absolute values. The purpose of this prior is simply to eliminate unrealistically elongated wall planes from consideration.

3.5 Evaluation of the shape prior

Having formulated a prior for the wall planes, we now move on to priors for the shape primitives which they contain. In this section an architectural shape prior for primitives, $\Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$, is defined and assessed using a Markov Chain simulation [17, 18]. This prior encodes information about:

- The scale of each primitive. For instance a door should be tall enough for a person to comfortably walk through. Scale priors can only be used when the absolute scale of the model is known.
- The shape of each primitive. For instance columns are likely to be long and thin, while pedestals are more broad and flat.
- The alignment of primitives. For instance windows are likely to occur in rows corresponding to the floors of a building.

[2] The walls are projected onto the ground plane before their interior angle is calculated as this is more robust to errors in the estimate of the vertical orientation of each wall plane than computing the interior angle directly in 3D.

- Other spatial relations such as symmetry about a vertical axis.

A prior which expresses all of these types of information will inevitably be difficult to formulate and to sample. When it is not feasible to directly draw samples from a distribution $\Pr(\boldsymbol{\theta})$, a common technique is to simulate the drawing of samples using a Markov Chain Monte Carlo technique defined on the parameters $\boldsymbol{\theta}$ [18, 27].

Because it is required to sample from parameter spaces of varying dimension, the Reversible Jump MCMC algorithm [19] is used. The behaviour of this algorithm is largely dependent on the choice of scoring function used to accept or reject jumps, and the jumping distribution from which candidate jumps are drawn. These are the subjects of the following sections.

3.5.1 The scoring function

The scoring function contains terms relating to the scale, shape and alignment of primitives:

$$f_{prior}(\boldsymbol{\theta}) = f_{scale}(\boldsymbol{\theta}) + f_{shape}(\boldsymbol{\theta}) + f_{align}(\boldsymbol{\theta}) + f_{sym}(\boldsymbol{\theta}) \tag{9}$$

The shape and scale terms apply only to individual primitives, and are defined by simple distributions, which are fully listed in [10]. The purpose of these terms is mainly to disqualify implausible primitives such as doors which are too thin or short to be practical, windows which extend between floors of the building, or buttresses which do not reach the ground.

The component of the scoring function for the alignment of shapes into rows and columns computes the deviation of the shapes from an aligned grid containing R rows and C columns. Initially, primitives that overlap horizontally are grouped into rows, and those which overlap vertically are grouped into columns. The alignment score is then defined as

$$f_{align}(\boldsymbol{\theta}) = \sum_{r=1}^{R} \left[\mathrm{Var}(\mathbf{t}_r) + \mathrm{Var}(\mathbf{b}_r) + \mathrm{Var}(\mathbf{r}_r - \mathbf{l}_r)\right] \tag{10}$$

$$+\sum_{c=1}^{C}\left[\mathrm{Var}(\mathbf{l}_c)+\mathrm{Var}(\mathbf{r}_c)+\mathrm{Var}(\mathbf{t}_c-\mathbf{b}_c)\right] \tag{11}$$

where $\mathbf{t}_r, \mathbf{b}_r, \mathbf{l}_r, \mathbf{r}_r$ are the top, bottom, left and right coordinates of the shape primitives belonging to row r and $\mathbf{t}_c, \mathbf{b}_c, \mathbf{l}_c, \mathbf{r}_c$ are similarly defined for primitives belonging to column c. The function $\mathrm{Var}(\mathbf{x})$ gives the variance of the elements of $\mathbf{x}$. When a wall contains 0 or 1 primitives, it is assigned a fixed score of high variance (the height or width of the wall on which it occurs) which encodes a preference for more than one window per wall if the wall is large enough to accommodate it.

The symmetry component of the scoring function is minimised when a row is exactly centred on a wall, and increases quadratically:

$$f_{sym}(\boldsymbol{\theta})=\sum_{r=1}^{R}\left[(l_r-\mathbf{l})-(r_r-\mathbf{r})\right]^2 \tag{12}$$

where l_r is the leftmost point of row r, r_r is the rightmost point of row r, and $\mathbf{l}$ and $\mathbf{r}$ are the left and right coordinates of the wall on which the row appears. The symmetry function is applied only to rows as it was found that columns of shapes are not generally vertically centred on a wall.

3.5.2 The jumping distribution

As well as a scoring function, an MCMC algorithm requires the specification of a jumping distribution $J_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$. The jumping distribution we use is a mixture of several types of jump, which are listed in Table 3.

There are a number of issues bearing on the choice of jump types to use:

- A building should always be a closed structure with walls which intersect at near right angles. Therefore the add/remove/modify wall jumps actually add, remove or modify a closed set of perpendicular walls to the model, effectively adding or removing a room from the model while maintaining closure.

- For efficiency, it should be easy to sample from the jumping distributions. The process will converge more rapidly if each jump traverses a significant distance in parameter space, and has a high acceptance rate. Therefore simple jump types which are likely to generate a more probable model should be used.

- All jumps used in the algorithm must be reversible. This poses difficulties when considering jumps such as aligning a group of shapes into a regular column. To maintain reversibility, a related jump must be included which can take a column of shapes and perturb each shape so that if the column-aligning jump is applied again, the same column configuration would result (this is jump $\mathcal{J}_{12}$ in Table 3).

Table 3: *Jump types available to MCMC algorithm. The parameter n identifies a single primitive to which the jump is applied. Jump $\mathcal{J}_{10}$ "regularises" a row or column of shapes by aligning all shapes, making them the same size, and evenly spaced. Jump $\mathcal{J}_{11}$ is similar but also positions the row so that it is centred in the wall.*

Jump type	Description	Parameters
$\mathcal{J}_1$	Add shape	$\mathcal{M}_i, x, y, w, h$
$\mathcal{J}_2$	Remove shape	n
$\mathcal{J}_3$	Modify shape	n, x, y, w, h
$\mathcal{J}_4$	Add wall	n, w, h
$\mathcal{J}_5$	Remove wall	n
$\mathcal{J}_6$	Modify wall	w, h
$\mathcal{J}_7$	Add window row/col	n
$\mathcal{J}_8$	Remove window row/col	n
$\mathcal{J}_9$	Modify window row/col	n
$\mathcal{J}_{10}$	Regularise window row/col	n
$\mathcal{J}_{11}$	Symmetrise window row	n
$\mathcal{J}_{12}$	Perturb window row/col	n, x, y, w, h

A full list of the distributions associated with each jump type appears in [10].

3.5.3 Verifying the shape prior

Having specified a scoring function and jumping distribution, the Reversible Jump MCMC algorithm can be used to simulate drawing independent samples from the shape prior that they define. It is important to sample from the shape prior to verify that it generates plausible buildings, and that is not too restrictive, in which case it would produce very similar buildings.

In the following experiment, a total of 9 Markov processes are seeded from one of 3 starting points, shown in Figure 4: a square "hut", a tower, or a bungalow shape. Each of the seed models has one wall containing a door at ground level, and each wall contains 0 or 1 windows. Each Markov chain

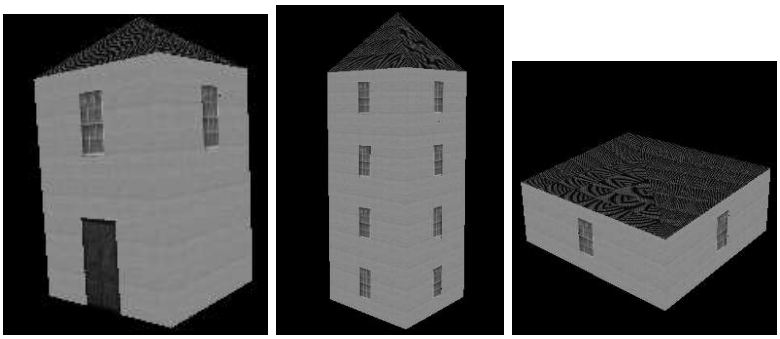


Figure 4: *"Hut", "tower" and "bungalow" seed points for the MCMC algorithm.*

is iterated for 2000 jumps. After this period, samples are drawn at random from each chain and displayed in Figure 5. Samples are drawn at random intervals rather than from consecutive iterations; consecutive iterations tend to be correlated as most jumps entail only a minor change to the model. The samples are displayed as a city of buildings to illustrate both the inter-building variation of the results and the plausibility of each individual structure.

Figure 5: *Collection of Classical style buildings generated from the shape prior.*

3.6 Evaluation of the texture prior

3.6.1 Learnt texture prior

The texture prior $\Pr(\boldsymbol{\theta}_T|\boldsymbol{\theta}_L\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$ is learnt from a training set of over 50 frontal images of architecture collected from both local sites and websites. The training set includes images from the example image sets and results used throughout this paper. In each training image, textures belonging to primitives and to the wall are manually marked and labelled. Textures from the same primitive type vary in appearance due to a number of factors, such as lighting and scale. To reduce the variation induced by these factors, a wavelet decomposition is applied to the interior of each texture. The 5/3 biorthogonal filter bank[3] (also used in [34]) is chosen to effect the decomposition, as it has several desirable properties: (a) compact support (which allows accurate localisation of features); (b) linear phase (so that the orientation of filters at different scales remains constant); (c) its low pass filter has a zero-order vanishing moment, which cancels effects of global illumination changes. An example of two texture patches and their wavelet decompositions is given in Figure 6.

The wavelet transform is applied horizontally and vertically to the texture. This decomposes it into 4 subbands at each scale level: the HL (horizontal high-pass, vertical low-pass) subband, the LH ((horizontal low-pass, vertical

[3]Lowpass filter coefficients: {0.1768, 0.3535, 1.0607, 0.3535, 0.1768}. Highpass filter coefficients: {0.3535, -0.7071, 0.3535}.

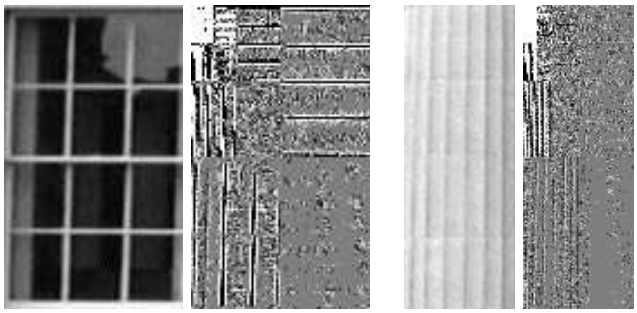


Figure 6: *Window and column texture. Beside each texture is its corresponding (quantised) wavelet transform. Output is shown for 3 levels of the wavelet transform. At each level, the HL (horizontal high-pass, vertical low-pass) component is shown at the bottom left, the HH component at the bottom right, the LH component at the top right and the LL component at the top left. The HL component responds strongly to vertical edges, while the LH component detects horizontal edges. The LL component is a smoothed, subsampled version of the original texture.*

high-pass) subband, the HH (horizontal high-pass, vertical high-pass) subband and the LL (horizontal low-pass, vertical low-pass) subband. The transform is applied recursively to the LL subband to obtain the same 4 subbands at a coarser level of scale.

As in [32, 34, 36], a texture model is represented by a set of histograms. The histograms are formed by counting the number of occurrences of every 3×3 pattern in the HL, LH and HH subband of each level of the texture's wavelet decomposition. The decomposition output is quantised to 3 levels to limit the size of each histogram to $3^9 = 19683$ bins. The prior probability of a patch of wavelet coefficients is then given by its frequency in the corresponding normalised histogram. These histograms are usually sparse, so the frequency of each bin is preset to 1 so that no 3×3 patch of texture has zero probability. Because texture is sampled regularly from the model surface and not the image, distortion due to the camera not being front on to the model surface is automatically nullified.

4 A model acquisition algorithm

4.1 Overview of the algorithm

The MAP parameter estimation algorithm, summarised in Algorithm 1, comprises 3 main stages. In the first stage the plane parameters $\boldsymbol{\theta}_P$ are estimated from a sparse reconstruction of the scene. In the second stage, likely shape primitives are located for each plane using a combination of single and multiple image likelihood measures. Finally these hypothesised primitives and planes are used as seed points for an MCMC sampling algorithm which simulates the posterior distribution $\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI})$.

Algorithm 1 Obtaining a MAP model estimate.

Stage 1: Plane initialisation (Section 4.2)
Detect and sequentially match corner and line features, and self-calibrate cameras [26] to obtain a sparse reconstruction
Recursively segment planes from the reconstruction using RANSAC, and heuristics applicable to architecture [11].

for each segmented plane **do**
  Stage 2.1: In a single image: (Section 4.3.1)
  for each primitive type **do**
    Set parameters d, a, b, dw to 0.
    Regularly sample remaining parameters x, y, w, h
    Rank model according to likelihood ratio $\mathcal{L}_S$ (14).
  end for

  Stage 2.2: In multiple images: (Section 4.3.2)
  for N_R highest ranked primitives **do**
    Draw parameter d from $\Pr(d|\theta_L\{xywh\}\mathbf{MI})$
    Search for maximum likelihood shape parameters $\{x, y, w, h, d\}$.
    Use model selection measure to determine value of remaining primitives $\{a, b, dw\}$ (Section 4.3.3)
    Re-rank the primitive based on likelihood ratio $\mathcal{L}_M$ (15).
  end for
end for

Stage 3: MCMC simulation of the posterior (Section 4.4)
Use combinations of likely primitives as seed points for MCMC algorithm. Use MCMC algorithm to simulate sampling of the posterior and retain the most probable (MAP) visited model.

4.2 Estimation of the plane parameters

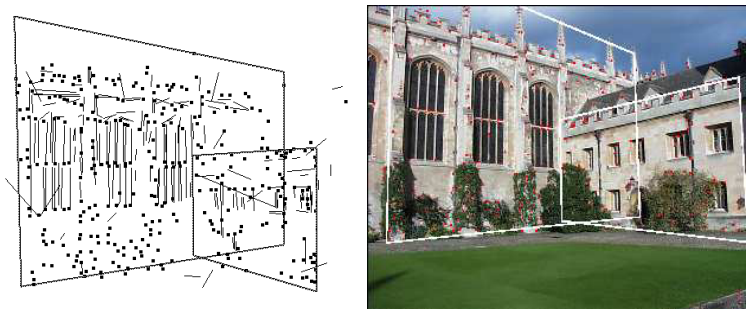
An initial estimate of the plane parameters $\boldsymbol{\theta}_P$ is obtained from a feature-based 3D reconstruction. The reconstruction is generated automatically using a sequential structure and motion algorithm similar to that described in [4]. This method is described in fuller detail in [11].

A plane plus parallax decomposition [9, 8, 41] was found to be particularly useful for architectural scenes with repeated structure. By initially estimating a planar homography $\mathbf{H}$ relating correspondences which lie on a plane in the scene, matches are constrained to appear at a point rather than on a line. Subsequent matches are still found by searching along epipolar lines, but in general repeated structure which occurs on planes has already been correctly matched and is therefore unambiguous. Camera calibration assumes that all internal parameters of the camera except focal length are known and uses the technique of Mendonca et al. [26].

4.2.1 Extracting planes from the reconstruction

Dominant planes are robustly extracted from the 3D points using a RANSAC technique (see [11]). The two most commonly occurring orientations of lines belonging to each plane are assumed to be the horizontal and vertical axes of the plane, as in architectural scenes most lines occur at horizontal and vertical orientations in the world. The sides of the wall are then aligned with these two dominant line directions, and the height and width of the wall are chosen to minimally span all horizontal and vertical lines and points belonging to the plane (see Figure 7). To extract subsequent planes, this process is recursively applied to the set of points which are outliers to all previously extracted planes. Further details are given in [11].

Thus a simple initial planar model is obtained in which all planes have rectangular shape and are perpendicular to a ground plane. This model is next



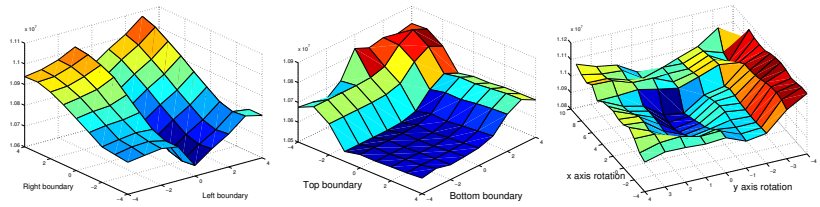
(a) (b)

Figure 7: *(a) Initial plane estimates based on including all inlying points and lines from the point based reconstruction. (b) Superposition on original image.*

used to seed a search for an improved model estimate.

4.2.2 Optimising the wall planes

The two main sources of error in the initialisation of the plane parameters $\boldsymbol{\theta}_P$ are the estimation of the ground plane normal, and the approximation of each plane boundary by a rectangle. Hence the model is optimised by gradient descent search on the components of the ground plane normal, and the height and width of each wall. The cost function used for the minimisation is the negative log likelihood of the model, some values of which are shown as the plane parameters are varied about their initial estimates in Figure 8. In Figure 8(c) the cost surface as the model is rotated about its x and y axes is quite irregular. This is typical of most scenes to which this algorithm has been applied. In practice it has been found that the search must be seeded with a model whose orientation is within about 0.1 radians rotation about each axis to converge to the correct solution. Another difficult situation for the search is illustrated in Figure 8(b). The flat region of the cost surface corresponds to a change to the top boundary which includes or excludes sky from the wall



(a) (b) (c)

Figure 8: *Cost surface as plane parameters are varied about their initial position. The parameters being varied belong to the leftmost plane in Figure 7(b), representing the side wall of a chapel. (a) Varying left and right boundary position. The left boundary minimum is sharply defined, the right less so as it is partly obscured by the other plane in the scene. Each axis unit is a 200 unit translation (the chapel plane is approximately 1600 units wide). (b) Top and bottom boundary height. At heights above the top of the wall, the error is fairly uniform as the sky is quite featureless. Similarly for heights below ground level, as the ground is quite featureless. (c) Rotation about x and y axes. Each axis unit corresponds to a rotation angle of* 0.1 *radians.*

plane. Because the sky is relatively featureless, its inclusion in the wall plane contributes very little to the likelihood. A similar situation occurs when the bottom boundary includes or excludes regions of the grassy area in front of the building. For this scene the minimum of the cost function does still occur at the correct position of the top and bottom boundary, but to deal with this instability the boundaries are not extended unless there is a significant change in the model likelihood.

4.3 Initial estimation of primitives

Having initialised and optimised estimates of plane position and orientation, the wall plane parameters are now fixed, and the remaining steps of the algorithm focus entirely on the detection and estimation of volumetric shape primitives offset from each plane. This simplification can be a problem when the estimates of plane parameters are inaccurate, as discussed in Section 5.4. However, in

general if the estimate of the underlying wall planes is inaccurate, then the subsequent estimation of the primitives belonging to those planes is unlikely to be correct, and thus cannot be used to recover from the initial plane estimation error.

The problem of estimating the number of primitives n_P which best model the scene, along with their type $\boldsymbol{\theta}_L$, shape $\boldsymbol{\theta}_S$ and texture $\boldsymbol{\theta}_T$ is very difficult due to the number of parameters which are required. Thus we proceed by factoring the problem into smaller parts, obtaining a set of approximate solutions first and refining them using increasingly accurate approximations to the posterior cost function $\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI})$.

As each primitive may contain thousands of texture parameters $i(\mathbf{g})$, the parameter space $\boldsymbol{\theta}_T$ is not searched. Instead the texture parameters are assigned a value based on the model state and image data as described in Section 4.3.4. The shape and label parameters, $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_L$, are optimised in two steps, to further reduce the number of parameters involved in each search. An initial search based on an approximate likelihood function (Section 4.3.1) locates likely values for a subset of the shape parameters, which are then used to seed searches in the full parameter space using the complete likelihood function (Section 4.3.2).

4.3.1 Single image shape proposal

The search for MAP model parameters is initialised by sampling the shape parameters $\boldsymbol{\theta}_{S^1} = \{x, y, w, h\}$ at regular intervals, for each primitive type $\boldsymbol{\theta}_L$. The values of each parameter are sampled only from intervals in which their prior probability is above some threshold, typically set to be 1/1000 of the peak value of the prior. The remaining shape parameters d, a, b, dw are fixed at 0. Because estimating the texture parameters (Section 4.3.4) requires all of the shape parameters to be known, $\boldsymbol{\theta}_T$ is approximated as $\boldsymbol{\theta}_{T^1}$ which is the projection of each wall plane onto a single near frontal image $\mathbf{D}_1$. This single

image likelihood $\Pr(\mathbf{D}_1|\boldsymbol{\theta}_{T^1}\boldsymbol{\theta}_{S^1}\boldsymbol{\theta}_L\mathbf{MI})$, is evaluated using the texture likelihood histograms defined in Section 3.6.

More precisely, it is computed as

$$\prod_{i=0}^{n_W}\prod_{\mathbf{x}\in\boldsymbol{\theta}_T^i}\prod_k f_k^{\theta_L^i}(\mathbf{W}_k(\mathbf{x})) \tag{13}$$

where $\mathbf{W}_k(\mathbf{x})$ denotes a 3×3 patch of subband k of the wavelet decomposition centred at $\mathbf{x}$ and $f_k^{\theta_L^i}$ denotes frequency in the normalised histogram for primitive type θ_L^i of the subband k wavelet decomposition. Usually 3 levels of the wavelet transform are computed, and thus there are 9 subbands: the HL (horizontal high-pass, vertical low pass), LH (horizontal low-pass, vertical high-pass) and HH (horizontal high-pass, vertical high-pass) subbands for each level. The final LL (horizontal low-pass, vertical low-pass) subband is not used. In practice the histograms recording the HL, LH and HH subbands at each level of scale are merged to ensure the texture likelihood measure is tolerant to global changes in scale. Another technique which has proved useful in practice is to partition the texture parameters into those neighbouring the boundary of a shape primitive, and those which do not. Texture on the boundary of each primitive generally has a steep intensity gradient and can be used to locate primitives whose texture is not easily distinguished.

The single image likelihood is a good indicator of the full likelihood function $\Pr(\mathbf{D}|\boldsymbol{\theta}_T\boldsymbol{\theta}_S\boldsymbol{\theta}_L\mathbf{MI})$ because it is insensitive to errors in the fixed shape parameters d, a, b, dw: the projection of a primitive to a frontal view is affected only slightly by changes in its depth, and the a, b, dw parameters are generally small compared to x, y, w and h.

The single image likelihood function is used to highlight regions of the model which are likely to be occupied by a shape primitive. This is approached as a problem of hypothesis testing [22], in which the hypothesis to be tested is that there is a primitive of type θ_L^i and shape θ_S^i. Likely values of θ_L^i and $\theta_{S^1}^i$

for primitive i are recorded by sampling and ranking primitives of each type according to their single image likelihood ratio:

$$\mathcal{L}_S^i = \frac{\Pr(\mathbf{D}_1|\theta_{T^1}^i \theta_{S^1}^i \theta_L^i \mathbf{MI})}{\Pr(\mathbf{D}_1|\theta_{T^1}^i \theta_{S^1}^i \overline{\theta_L^i} \mathbf{MI})} \tag{14}$$

where $\overline{\theta_L^i}$ denotes every primitive type except θ_L.

(a) (b)

Figure 9: *Collection of shapes whose likelihood ratio of being a window to not being a window is high. The zig-zag shapes correspond to columns.*

This likelihood approximation is useful because it can be computed using a subset of the shape parameters, and because it can distinguish primitives of a similar shape based on their texture. However texture also has drawbacks as a means of distinguishing primitives. A wall may exhibit several different types of texture, be it stone, brick, wood or some other material. Similarly windows may have many different patterns of glass and framework. It is therefore unrealistic to have a single texture histogram for each type of primitive. Instead separate texture histograms are learnt for stone and brick walls, for windows with vertical as opposed to horizontal and vertical pane dividers, for columns with and without fluting, and so on. A separate histogram is learnt for each of the global parameter settings in Table 2.

The prior $\Pr(\boldsymbol{\theta}_L|\boldsymbol{\theta}_P\boldsymbol{\theta}_G\mathbf{MI})$ is used to limit the number of primitive types which must be rated using the single image likelihood ratio, in the same way that shape priors limit the range of shape parameters for each primitive. A full definition of this prior is given in [10]. In practice the Downing library model (Figure 1) is the scene with the greatest number of competing primitive types (it has 6) for which this algorithm has succeeded.

A list is maintained of the N_R primitives with the highest ratio (14). N_R is chosen to exceed the maximum number of primitives which are expected to appear on each wall plane—in the Downing Library scene, N_R is set to 50. Some hypotheses are shown in Figure 9.

4.3.2 Multiple image shape verification

In multiple views, maximum likelihood shape parameters $\boldsymbol{\theta}_S^{ML}$ are found for each primitive proposed from a single image. First, the depth parameter d for each hypothesised primitive is found by sampling over a range of depths for which the shape prior $\Pr(d|\theta_L\{x, y, w, h\}\mathbf{MI})$ is significant (again usually 1/1000 of its peak value). An example of the variation of the likelihood over a range of depth values is given in Figure 10. Note that where an offset window exists, there is a sharp peak in likelihood at the correct depth value, whereas in regions which lie on the wall plane the likelihood value is more uniform as depth is varied.

4.3.3 Estimation of remaining shape parameters

Having maximised the likelihood of each primitive over the shape parameters $\{x, y, w, h, d\}$, a model selection criterion is used to decide whether additional shape parameters a, b, dw are required.

Model selection is based on the *evidence* [22] which can be approximated as the product of the maximum likelihood for a model and an Occam Factor

(a) (b)

Figure 10: *(a) Three candidate windows whose depth is to be varied. (b) The log likelihood of each window over a range of depths. The log likelihood of the window which is positioned over the actual window (the solid line) is strongly peaked, while the candidate which belongs to the wall (dashed line) is relatively uniform. A region which spans both window and wall (dot-dashed line) peaks at the depth of the window, but not as strongly as the true window candidate. Although changing the depth of the window affects a minority of its texture parameters, the change in likelihood is still significant (note that (b) plots* log *likelihood).*

which penalises model complexity. The maximum likelihood parameters for each model are found using a gradient descent search, initialised at the shape estimate based on only the $\{x, y, w, h, d\}$ parameters. Further details and results for the Occam Factor and other model selection criteria are given in [39]. It turns out that the likelihood is the dominant term in the selection process, and the main requirement of the model selection step is to prevent over-fitting; that is using an unnecessarily complex model to the data.

4.3.4 Texture parameter estimation

Having estimated both $\boldsymbol{\theta}_L$ and $\boldsymbol{\theta}_S$, only the texture parameters $\boldsymbol{\theta}_T = \{i(\mathbf{g})\}$ remain to be found. It is assumed that each texture parameter is observed with noise $i(\mathbf{g}) + \epsilon$, where ϵ has a Gaussian distribution mean zero and standard deviation σ_ϵ. Each parameter $i(\mathbf{g})$ can then be found such that over m views it minimizes the sum of squares $\sum_{j=1}^{j=m} \left(i(\mathbf{g}^j) - i(\mathbf{g})\right)^2$ where $i(\mathbf{g}^j)$ is the intensity at $\mathbf{g}^j$, and $\mathbf{g}^j$ is the projection of $\mathbf{g}$ into the jth image.

4.3.5 Maximum likelihood model estimation

At this stage, a complete set of shape, label and texture parameters has been estimated for a collection of hypothesised shape primitives. Once again, the problem of choosing which primitives to include in the model is treated as a series of hypothesis tests. For each primitive, the hypothesis to be tested is whether it should be a part of the model, but this time the results are ranked based on the full likelihood ratio:

$$\mathcal{L}_M^i = \frac{\Pr(\mathbf{D}|\theta^i \mathbf{M}\mathbf{I})}{\Pr(\mathbf{D}|\theta_0 \mathbf{M}_0 \mathbf{I})} \tag{15}$$

where $\mathbf{M}_0$ is the model containing no primitives and $\mathbf{M}$ is the model containing only primitive i, with parameters θ^i estimated as described in previous sections. This produces a list of potential shapes, ranked according to their likelihood. A maximum likelihood model would include all non-overlapping shapes whose

likelihood ratio is greater than 1. However there are many situations in which the likelihood function can be misleading, such as parts of the model which are occluded or subject to unmodelled lighting effects, or which have uniform texture.

The fallibility of both the single and multiple image likelihood functions suggests that better use should be made of the prior to resolve cases where they fail due to ambiguous or misleading image data. The maximum likelihood list of shapes is therefore treated as a set of possible starting points for an MCMC algorithm which incorporates the shape prior with the likelihood function to simulate sampling from the posterior density function, rather than simply the likelihood. This is the subject of the next section.

4.4 Finding the MAP model

In this section, the Reversible Jump MCMC algorithm is applied to the posterior probability function $\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI})$ using the same jump types that were applied to the shape prior in Section 3.5.3. The scoring function is altered to include image information by adding a log likelihood term, so that the complete score is now given by

$$f(\boldsymbol{\theta}) = \lambda f_{prior}(\boldsymbol{\theta}) + \sum_{i(\mathbf{g})} \sum_{j=1}^{m} \left(\frac{i(\mathbf{g}^j) - i(\mathbf{g})}{\sigma} \right)^2 \tag{16}$$

where $i(\mathbf{g}^j)$ is the projection of the texture parameter $i(\mathbf{g})$ onto the jth image, σ is an image variance parameter and λ is a relative scale factor.

Starting points for the MCMC algorithm are generated from the earlier steps of the MAP estimation algorithm, described in previous sections. These "seed" models are generated by setting an acceptance threshold on the likelihood of primitives. It is first tested on the image sequence of the Downing College Library shown in Figure 1, with acceptance thresholds chosen to include between 6 and 15 shapes. The original image resolution is 1600×1200. In each case,

after running the MCMC algorithm for 2000 iterations, it was run for a further 3000 iterations with $\lambda = 10^4$ and $\sigma = 10$, and the model and parameter values with the maximum posterior probability were retained.

The MAP model is shown in Figure 11. This reconstruction is obtained using MCMC jumping distributions which do not include the addition, modification or deletion of wall planes, so that only the parts of the model which are seen in the images are reconstructed. This model has some nice properties:

- The wireframe is constructed from a collection of simple primitives which have been assigned an interpretation, rather than as an unstructured cloud of 3D points. This makes it very simple to render and manipulate the model.

- The columns at the entranceway are correctly reconstructed despite only their front face being visible in the images. This is only possible due to the use of strong prior models for shape.

- In addition to the two base planes, there are 25 extra primitives, occurring in 5 rows, containing a total of 70 parameters, i.e. fewer parameters than the 72 required for a set of 24 3D points in general position.

- Although not shown in the reconstruction, each part of the model is *labelled*. This semantic information can be used to reason about the model and improve its appearance and accuracy. For instance, each window in the model is automatically rendered as a shiny and partially transparent object.

When jump types involving wall plane parameters are included in the MCMC algorithm, closure of the building is enforced and the reconstruction converges to a symmetric model such as that shown in Figure 12. The texture for this model is cut and pasted from areas of the image identified as a wall, window,

columns and so on, and the same texture sample is used for every instance of a type of primitive.

The operation of this algorithm is shown in Figures 13 for the Trinity Chapel sequence. Note that the entire model is obtained from only 3 images. The use of prior information and a representation of shape tuned to architecture results in a model which is more accurate, more complete, and more realistic than one based solely on image data [10]. Although the model is not completely accurate in areas which are not visible in the images, it is a plausible structure, and is obtained automatically except for the prior specification of the structure as being Gothic, and the restriction of the variety and shape of primitives this entails. [4]

5 Evaluation

5.1 Comparison with ground truth

For verification purposes, some ground truth measurements were taken from the Downing College library, reconstructed in Figure 12.

The height, width and depth of a set of windows belonging to this building were measured with a tape measure. The resulting lengths are shown in Figure 14. Because the absolute scale of the model is unknown, only ratios of lengths are compared to ground truth values. It is assumed that there is a $\pm$1cm error in each measurement. In Table 4, a comparison of the corresponding model values and ground truth measurements is given. The uncertainty in the model values is based on the resolution of the grid of texture parameters on each plane. It can be seen that the ratios of window height to width, and width

[4] The width of each part of the building is obtained from the average size of the window or door primitives on visible walls—each segment of the building is made wide enough to accommodate one window of height and width equal to the average height and width of the visible windows, with spacing to either side equal to half the window width. In the absence of image information, this seems a reasonable assumption to make and produces generally plausible architectural models.

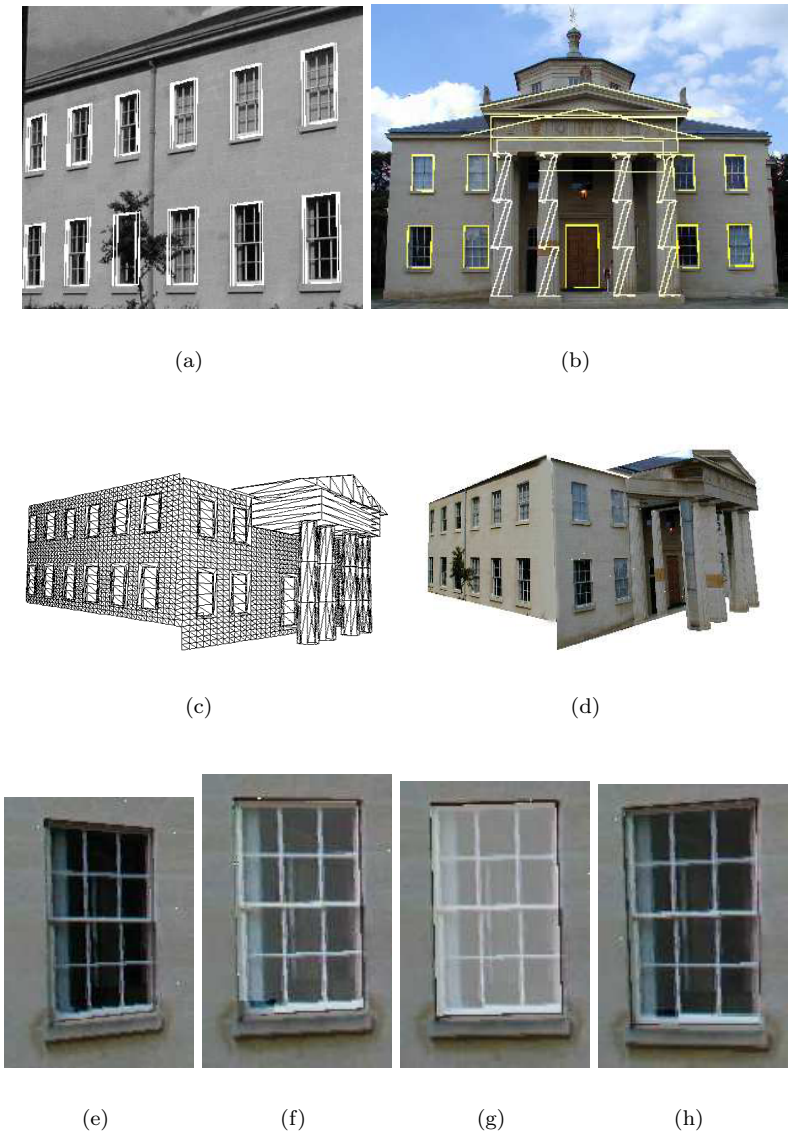
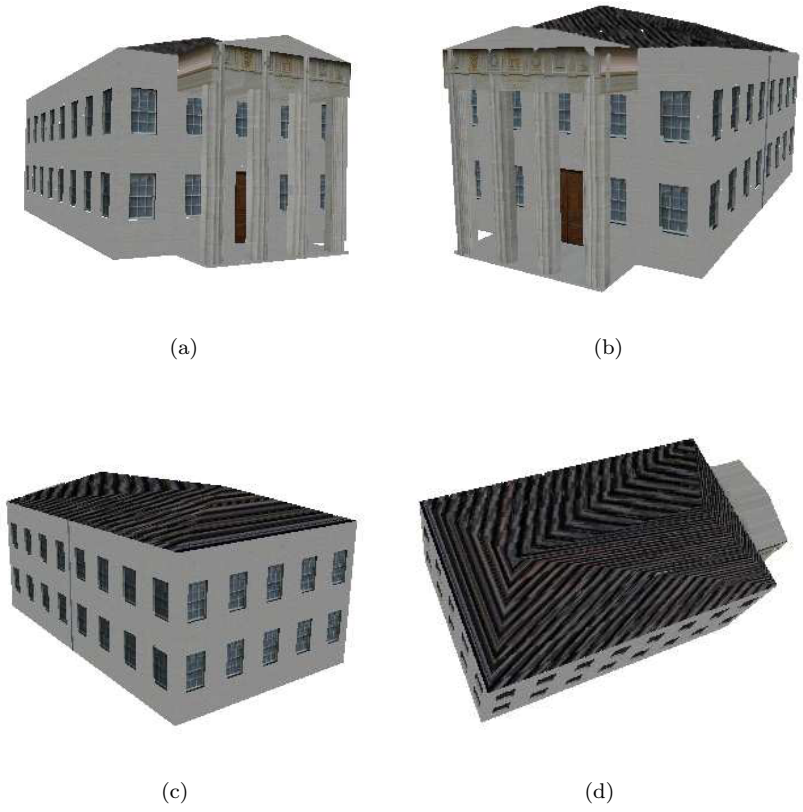


Figure 11: *(a) MAP model of side wall of Downing library, after 2000 MCMC iterations using the Classical prior. (b) Front wall. Both front and back faces of primitives are drawn, hence the pair of triangles and rectangles for the pediment and entablature. (c)-(d) 3D rendering of MAP Downing model, obtained without using add/remove/delete wall jumps. The textures shown on the model are automatically extracted from the images which are most front-on to each plane. (e)-(h) Window detail.*



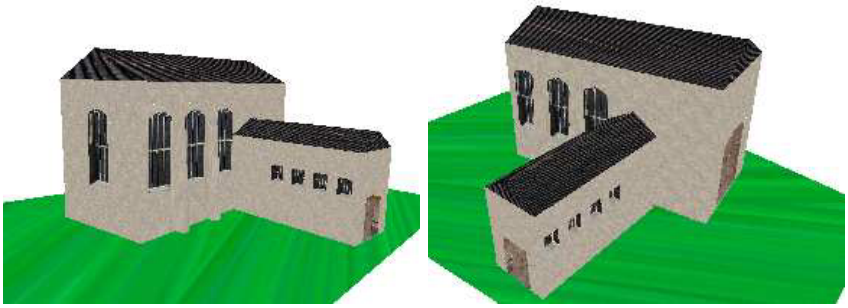
(a) (b)

(c) (d)

Figure 12: *Four views of the completed model of Downing library, with extra walls added. Even though only two walls are visible, a complete building has been modelled using symmetry. Wall, window, roof and column textures are sampled from the images and applied to the appropriate primitives.*

(a) (b) (c)

(d) (e)



(f) (g)

Figure 13: *(a)-(c) 3 original images of Trinity college courtyard. (d) MAP model primitives, superimposed on image. (e) Wireframe MAP model. (f)-(g) 3D model with texture taken from images.*

Figure 14: *Some measurements made of the Downing Library scene. The other windows in the scene are the same size as the one shown (to an accuracy of* $\pm 1cm$*).*

Ratio	Ground truth		Model	
	Lower	Upper	Lower	Upper
h/w	1.48	1.52	1.50	1.64
w/d	5.67	6.37	5.00	7.40
d_2/w	2.22	2.24	2.18	2.28
c/w	2.56	2.77	2.74	3.00

Table 4: *Comparison of ratios of window height (h) to width (w), width to depth (d) wall–column separation* (d_2) *to window width and the circumference of a column (c) to window width. The upper and lower bounds are based on* $\pm 1cm$ *accuracy for ground truth measurements, except for the circumference of the base of the column, which is measured to* $\pm 10cm$*. The accuracy of the model measurements is limited to the resolution of the texture parameters.*

to depth, are recovered to within the accuracy bounds of the ground truth measurements. The error margins are generally greater for model measurements, which are constrained by the resolution of the images. Although high resolution images (1600×1200 pixels) were used for this model, it seems that even more resolution is required to precisely recover fine details such as the depth of each window. A super-resolution method [21, 2], in which pixels from multiple images are combined to produce one very high resolution image, could possibly be used to obtain the required detail. The distance from the column to the wall is also identified accurately, but the circumference of the column is slightly underestimated (although the error margins just overlap). The circumference of the column is quite difficult to measure precisely, due to the stonework on its outer rim. Therefore there is an uncertainty of $\pm$10cm associated with its measurement—this is derived from the fact that there are 20 partitions in the fluting around the base of the column, each of which can be measured to a precision of approximately $\pm$0.5cm.

5.2 Reconstruction ambiguities

One feature of using an MCMC algorithm to sample the posterior is that as well as having a MAP model estimate, other probable samples can also be examined. This is useful for identifying ambiguities in the reconstruction. Four of the more marked ambiguities present in this model are shown in Figure 15.

5.3 The shape prior

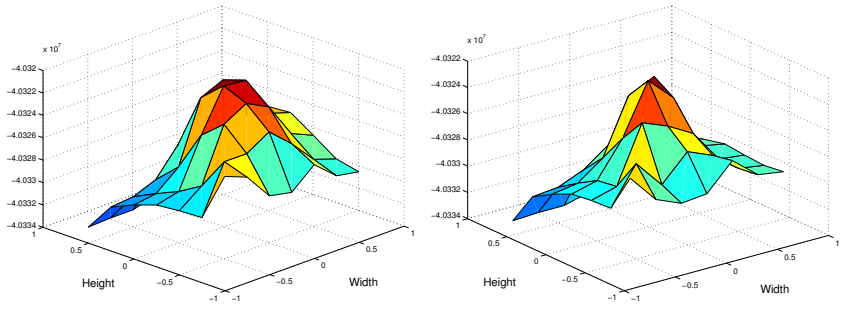
In Section 3.5.3, a city of models was generated to demonstrate that the structure of models sampled from the Classical shape prior varies substantially, but not beyond the bounds of plausibility. To test the effect of the choice of shape prior on the resultant model, the Classical and Gothic priors were swapped. Thus the Classical prior was used to reconstruct the Gothic Trinity chapel

(a) (b)

(c) (d)

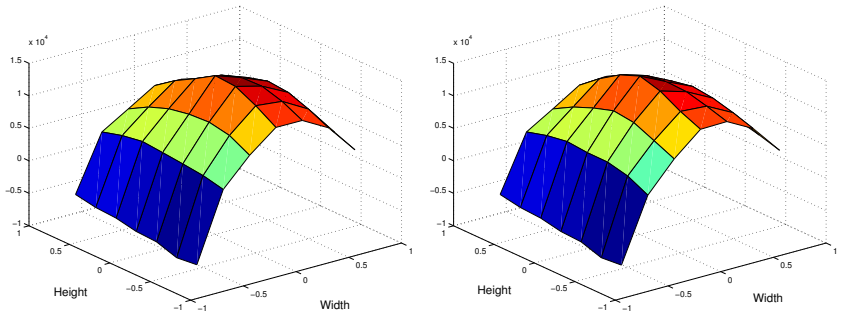
Figure 15: *Some ambiguities in the Downing model, chosen from the 20 most probable models visited by the MCMC process. (a) Window sills are included in the window primitives. (b) Windows are represented using two primitives each. (c) The door is omitted. (d) Extra columns are added in between the existing ones.*

model, and the Gothic prior was used for the Classical Downing Library model. It was found that using a Gothic prior had little effect on the reconstruction of the Downing Library. Although the shape of the posterior was slightly altered, the location of its maximum was unchanged (see Figure 16). Similarly the use of the Classical shape prior had little effect on the estimated shape of the Trinity chapel model, as once again the maximum a posteriori estimate is unchanged (Figure 17). However the choice of prior does affect the model in another way: in Gothic architecture, buttresses are quite common and are therefore assigned a high prior probability, whereas columns are extremely unlikely to occur. The opposite is true for Classical scenes, in which columns have a high prior probability and buttresses do not. Hence when a Classical prior is applied to the Trinity Chapel, the estimated model contains columns in place the buttresses reconstructed in Figure 13. A more detailed analysis of the effect of the shape prior, and a complete specification of the Gothic and Classical priors, is given in [10].



(a) (b)

Figure 16: *Posterior as height and width of window are varied, for a window in the Downing library scene. (a) Classical prior. (b) Gothic prior. The position of the peak value is unchanged.*



(a) (b)

Figure 17: *Posterior as height and width of window are varied, for a window in the Trinity Chapel scene. (a) Classical prior. (b) Gothic prior. The position of the peak value is unchanged.*

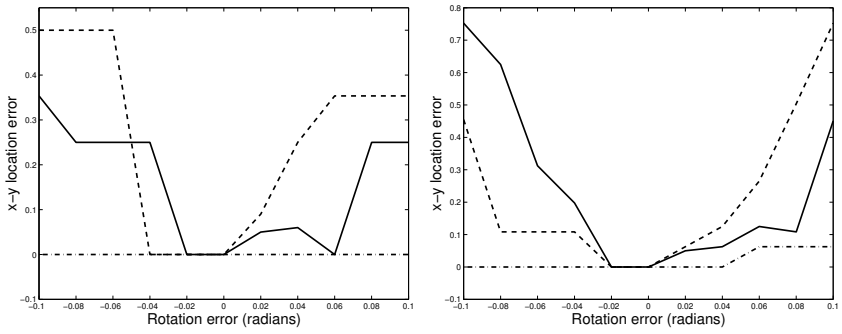
5.4 The likelihood measures

The estimation of architectural primitives, both their number and parameters, is based on the single and multiple image likelihood ratios, described in Section 4.3. In this section the behaviour of both these measures is examined when the underlying estimate of the wall planes is slightly inaccurate. This is important because the wall planes are estimated prior to the primitives, and are fixed while the primitives they contain are located. It is therefore useful to know the approximate error bounds on the estimation of the planes for which the estimation of the primitives is still likely to succeed.

Windows from the Trinity Chapel and the Downing Library scene, shown in Figure 18 (a) and (b), are chosen demonstrate the behaviour of these measures. A primitive is hypothesised in a range of x-y positions around its true location, and the single and multiple image likelihood ratios are recorded. The x-y positions are spaced evenly over a grid ranging from plus to minus three quarters of the width of the primitive in each direction. This is repeated for models in

which the wall plane has been deliberately offset by a small rotation about the x, y or z axis.

(a) Original shape (b) Original shape



(c) (d)

Figure 18: *(a)-(b) Primitives used to test the single image likelihood ratio. (c) Errors in x-y location of the primitive in (a) using single image likelihood ratio, as model is rotated about the x axis (solid line), y axis (dash-dotted line, equal to 0) and z axis (dashed line). (d) Same graph, for the primitive shown in (b). x-y location error is scaled such that the width of the primitive is 1.0.*

Figure 18 (c) and (d) show the error in the location of the maximum single image likelihood ratio primitive as the ground plane normal is rotated about the x, y and z axes. The position of the maximum likelihood primitive is quite sensitive to errors in the orientation of the wall plane. This is because as the

orientation of the plane changes, the boundary of each candidate primitive is no longer aligned with the boundary in the image. Thus there will inevitably be some overlap with the texture surrounding the window in the image. Because the edges of the hypothesised primitives are no longer aligned with edges in the image, the likelihood of the texture at the edges is also no longer sharply peaked. These factors make the result quite unstable. Similar issues arise for the multiple image likelihood ratio—a more detailed analysis of this instability is presented in [10].

These results show that if the wall planes are not accurately estimated, subsequent estimation of the primitives becomes unreliable. Often the effects of an inaccurate wall estimate, such as fragmentation of the primitives, can be fixed with some post-processing "hack", but this is hardly an ideal solution. One alternative is an iterative scheme which adjusts wall and primitive estimates in turn to obtain an optimal solution for both, but it is unclear how this could be carried out reliably.

5.5 Further reconstructions

Figure 19 shows a Classical style model, generated from 4 images of a pair of walls belonging to the Fitzwilliam Museum in Cambridge. The completed model is plausible, and accurate where it is visible in the images. Note that each of the 3 windows near the junction of the wall is recovered accurately, as are the pillars separating them. This kind of detail is possible from so few images due to the very specialised representation of shape and specification of prior knowledge. As the completed model uses generic Classical textures it lacks much of the detail visible in the model which uses texture parameters estimated from images of the scene. To prevent all completed models from looking too similar, a more elaborate technique may be needed for generating synthetic texture [13] which can combine image and prior information to retain the detail in the images but

compensate for lighting, occlusion and so on.

A second Gothic scene is reconstructed in Figure 20. The model is enhanced by automatically identifying texture parameters belonging to the most visible window, and copying them onto windows with less visibility. In the completed model, the rows of visible windows and buttresses are automatically extended to the end of each wall, and extra walls are added to form a closed model. Again however the texture in the completed model lacks distinctiveness.

Reconstructions of a wider range of scenes are reported in [10].

6 Conclusion

This paper has described a framework and an algorithm for automatically determining the structure and identifying a piece of architecture from a small number of images. The algorithm has been implemented and shown to work on a selection of architecture, and some evaluation of its current limitations has also been presented.

There is a fair amount of future work to be done on improving the reliability of the algorithm in the face of varying lighting conditions and architectural styles. A key step in this will be the integration of plane and primitive estimation. Ideally the estimate of both planes and primitives should be updated iteratively, as each is dependent on the other. The dependence of the algorithm on a feature based reconstruction limits its applicability to closely spaced image sequences with favourable lighting conditions. In the future we would like to remove this dependency.

References

[1] C. Baillard, C. Schmid, A. Zisserman, and A.W. Fitzgibbon. Automatic line matching and 3d reconstruction of buildings from multiple views. In *ISPRS Congress*, pages 69–80, 1999.

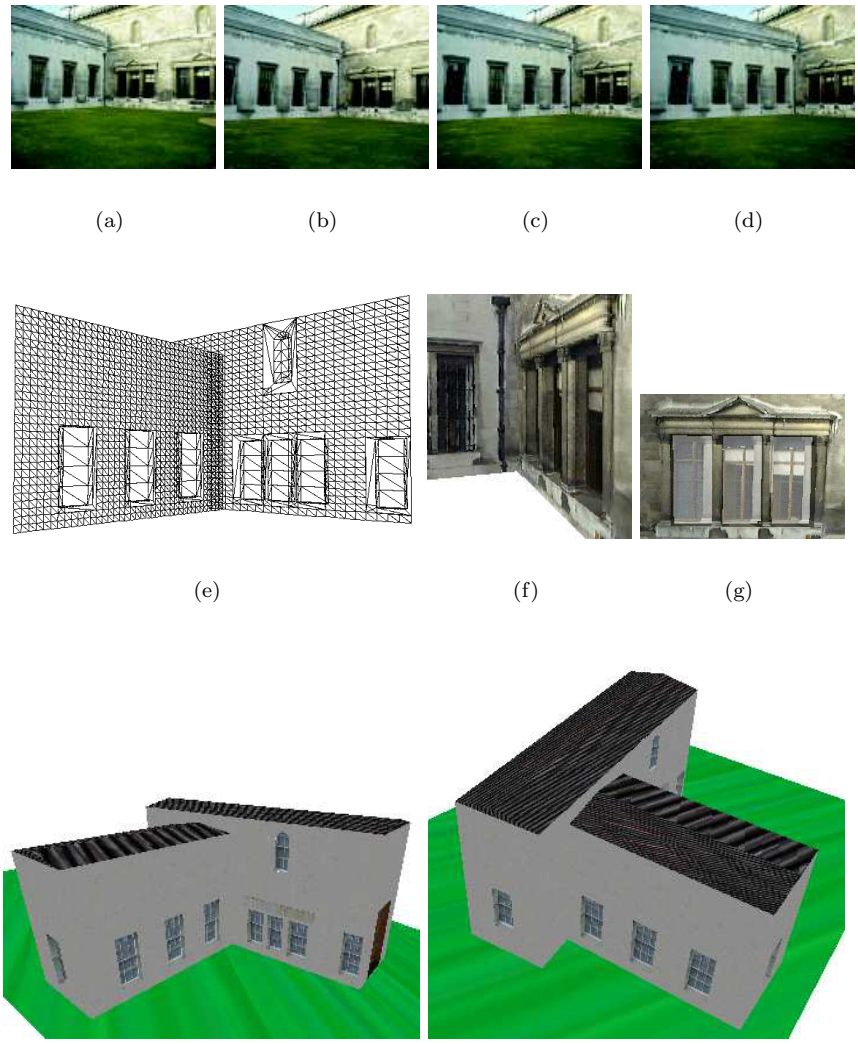


Figure 19: *(a)-(d) Original images of Fitzwilliam Museum. (e) Wireframe model. (f)-(g) Details of textured model, showing offset primitives. (h)-(i) Completed model.*

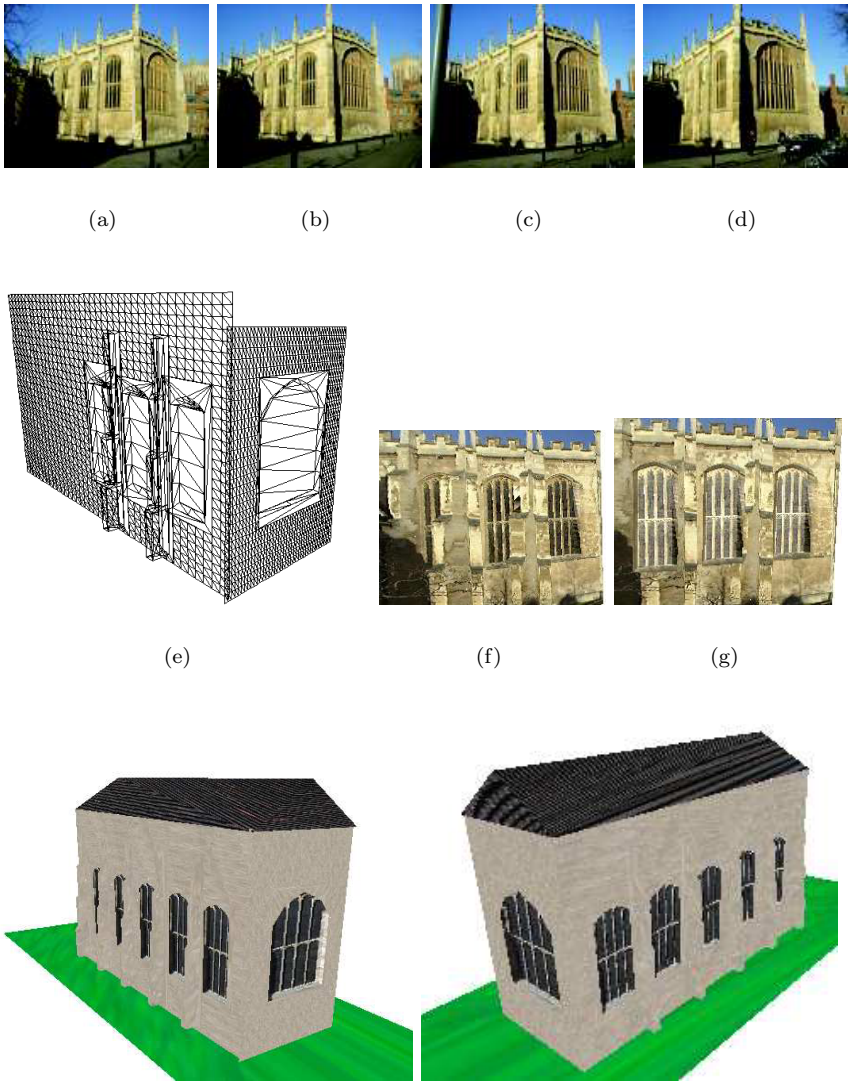


Figure 20: *(a)-(d): Four images of Trinity Chapel, taken from Trinity Street. (e) Wireframe model. (f) Original texture pasted from one view. Note that the texture is inaccurate in areas obscured by the buttresses. (g) Texture from most visible window is copied to less visible windows, improving the appearance of the model. (h)-(i) Completed model, with generic Gothic textures.*

[2] S. Baker and T. Kanade. Super-resolution: Reconstruction or recognition? In *Proc. IEEE-EURASIP Workshop on Nonlinear Signal and Image Processing*, Baltimore, Maryland, 2001.

[3] S. Baker, R. Szeliski, and P. Anandan. A layered approach to stereo reconstruction. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 434–441, 1998.

[4] P.A. Beardsley, A. Zisserman, and D.W. Murray. Sequential updating of projective and affine structure from motion. *International Journal of Computer Vision*, 23(3):235–259, 1997.

[5] I. Biederman. Human image understanding: Recent research and a theory. *Computer Vision Graphics and Image Processing*, 32(1):29–73, 1985.

[6] D.L. Borges and R.B. Fisher. Class-based recognition of 3d objects represented by volumetric primitives. *Image and Vision Computing*, 15(8):655–664, 1997.

[7] J. Canny. A computational approach to edge detection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 8(6):679–698, 1986.

[8] R. Cipolla, Y. Okamoto, and Y. Kuno. Robust structure from motion using motion parallax. In *Proc. IEEE International Conference on Computer Vision*, pages 374–382, 1993.

[9] R. Collins. Projective reconstruction of approximately planar scenes. In *Interdisciplinary Computer Vision: An Exploration of Diverse Applications*, pages 174–185, 1992.

[10] A.R. Dick. *Modelling and Interpretation of Architecture from Several Images*. PhD thesis, University of Cambridge, 2001.

[11] A.R. Dick, P. Torr, and R. Cipolla. Automatic 3d modelling of architecture. In *Proc. 11th British Machine Vision Conference (BMVC'00)*, pages 372–381, Bristol, 2000.

[12] S.J. Dickinson, R. Bergevin, I. Biederman, J.O. Eklundh, R. Munck-Fairwood, A.K. Jain, and A.P. Pentland. Panel report: The potential of geons for generic 3-d object recognition. *Image and Vision Computing*, 15(4):277–292, 1997.

[13] A. Efros and T. Leung. Texture synthesis by non-parametric sampling. In *Proc. IEEE International Conference on Computer Vision*, pages 1033–1038, 1999.

[14] O.D. Faugeras, J.L. Mundy, N. Ahuja, C.R. Dyer, A.P. Pentland, R. Jain, K. Ikeuchi, and K.W. Bowyer. Why aspect graphs are not (yet) practical for computer vision. *Computer Vision Graphics and Image Processing*, 55(2):212–218, 1992.

[15] J. M. Ferryman, A. D. Worrall, G. D. Sullivan, and K. D. Baker. A generic deformable model for vehicle recognition. In *Proceedings British Machine Vision Conference*, pages 127–136, 1995.

[16] R. Fisher. *From Surfaces to Objects: Computer vision and three dimensional scene analysis*. John Wiley and Sons, 1989.

[17] A. Gelman, J. Carlin, H. Stern, and D. Rubin. *Bayesian Data Analysis*. Chapman and Hall, Boston, 1995.

[18] W. Gilks, S. Richardson, and D. Spiegelhalter (editors). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London., 1996.

[19] P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.

[20] C. Harris and M. Stephens. A combined corner and edge detector. In *Proc. 4th Alvey Conference*, pages 147–152, 1988.

[21] M. Irani and S. Peleg. Using motion analysis for image enhancement. *Journal of Visual Communication and Image Representation*, 4(4):324–335, 1993.

[22] E. T. Jaynes. *Probability Theory: The Logic of Science*. Unpublished but available online at http://bayes.wustl.edu/etj/prob.html, 1996.

[23] J.J. Koenderink and A.J. van Doorn. The internal representation of solid shape with respect to vision. *Biological Cybernetics*, 32:211–216, 1979.

[24] D.G. Lowe. Fitting parameterized three-dimensional models to images. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13(5):441–450, 1991.

[25] D. Marr. *Vision: A Computational Investigation into the Human Representation and Processing of Visual Information*. W. H. Freeman, 1982.

[26] P.R.D.S. Mendonca and R. Cipolla. A simple technique for self-calibration. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages I:500–505, 1999.

[27] R. M. Neal. Probabilistic inference using monte carlo markov chains. Technical Report CRG-TR-93-1, University of Toronto, 1993.

[28] J. Oliensis. A critique of structure-from-motion algorithms. *Computer Vision and Image Understanding*, 80(2):172–214, 2000.

[29] C. Papageorgiou and T. Poggio. A trainable system for object detection. *International Journal of Computer Vision*, 38(1):15–33, 2000.

[30] M. Pilu and R.B. Fisher. Recognition of geons by parametric deformable contour models. In *Proc. 4th European Conference on Computer Vision*, Lecture Notes in Computer Science 1064, pages I:71–82. Springer-Verlag, 1996.

[31] A.R. Pope. Model-based object recognition: A survey of recent research. Technical Report 94-04, University of British Columbia, 1994.

[32] J. Portilla and E.P. Simoncelli. A parametric texture model based on joint statistics of complex wavelet coefficients. *International Journal of Computer Vision*, 40(1):49–70, 2000.

[33] C. Schmid and A. Zisserman. The geometry and matching of lines and curves over multiple views. *International Journal of Computer Vision*, 40(3):199–233, 2000.

[34] H. Schneiderman and T. Kanade. A statistical method for 3d object detection applied to faces and cars. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages I:746–751, 2000.

[35] C.C. Slama. *Manual of Photogrammetry, 4th ed.* American Society of Photogrammetry, 1980.

[36] J. Sullivan, A. Blake, M. Isard, and J.P. MacCormick. Object localization by bayesian correlation. In *Proc. IEEE International Conference on Computer Vision*, pages 1068–1075, 1999.

[37] 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.

[38] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: A factorization method. *International Journal of Computer Vision*, 9(2):137–154, 1992.

[39] P. Torr, A. Dick, and R. Cipolla. Layer extraction with a bayesian model of shapes. In *Proc. 6th European Conference on Computer Vision*, Lecture Notes in Computer Science 1843, pages II:273–289. Springer-Verlag, 2000.

[40] P. Torr, R. Szeliski, and P. Anandan. An integrated bayesian approach to layer extraction from image sequences. In *Proc. IEEE International Conference on Computer Vision*, pages 983–990, 1999.

[41] B. Triggs. Plane + parallax, tensors and factorization. In *Proc. 6th European Conference on Computer Vision*, Lecture Notes in Computer Science 1842, pages I:522–538. Springer-Verlag, 2000.

[42] B. Triggs, P. McLauchlan, R. Hartley, and A. Fitzgibbon. Bundle adjustment – A modern synthesis. In W. Triggs, A. Zisserman, and R. Szeliski, editors, *Vision Algorithms: Theory and Practice*, LNCS 1883, pages 298–375. Springer Verlag, 2000.

[43] J. Wang and E. H. Adelson. Layered representation for motion analysis. In *Proc. IEEE Computer Vision and Pattern Recognition*, pages 361–366, 1993.