A Bayesian Estimation of Building Shape Using MCMC

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

[1] Engineering Department
Cambridge University, Cambridge, UK
{ard28,cipolla}@cam.ac.uk
http://www-svr.eng.cam.ac.uk/ ard28/
[2] Microsoft Research, 7 JJ Thomson Avenue,
Cambridge, CB3 0FB, UK
philtorr@microsoft.com
http://research.microsoft.com/ philtorr/

Abstract. This paper investigates the use of an implicit prior in Bayesian model-based 3D reconstruction of architecture from image sequences. In our previous work architecture is represented as a combination of basic primitives such as windows and doors etc, each with their own prior. The contribution of this work is to provide a global prior for the spatial organization of the basic primitives. However, it is difficult to explicitly formulate the prior on spatial organization. Instead we define an implicit representation that favours global regularities prevalent in architecture (e.g. windows lie in rows etc.). Specifying exact parameter values for this prior is problematic at best, however it is demonstrated that for a broad range of values the prior provides reasonable results. The validity of the prior is tested visually by generating synthetic buildings as draws from the prior simulated using MCMC. The result is a fully Bayesian method for structure from motion in the domain of architecture.

1 Introduction

Many algorithms (e.g. [1,3,16,14]) have been developed for inferring 3D structure from a set of 2D images. A review of the state of the art in this area can be found at [17]. However there are often cases in which image information is ambiguous or misleading, such as in areas of homogeneous or repeated texture. In such cases extra information is required to obtain a model of the scene.

In the past, dense stereo algorithms have used heuristics favouring "likely" scenarios such as regularization or smoothing in an attempt to resolve these ambiguities (e.g. [5,16]), but in general these are unsatisfactory (for instance, the smooth surface assumption is violated at occlusion boundaries). It is our belief that maximum likelihood estimates (even regularized) of structure have progressed as much as they are able, and that further research in this area will yield negligible or arguable benefit. Our approach to structure from motion is to develop generic methods to exploit domain-specific knowledge to overcome these ambiguities. This has been successfully done for other 3D reconstruction domains, e.g. heads [9,18], bodies [15].

Within this paper we explore the reconstruction of generic buildings from images, using strong prior knowledge of building form provided by architects, this is most naturally done in a Bayesian framework. The Bayesian framework provides a rational

method for incorporating prior information into the estimation process [12]. However in complicated scenarios such as the modelling of architecture, there still remains two problems to be resolved. Whilst Bayes provides the basic laws for manipulating probabilities, we still need to resolve the problem of parameterization, and once the problem is parameterized choose the best algorithm to optimize the parameters.

Structure is represented as a collection of planes (corresponding to walls) and primitives (representing windows, doors and so on). Each primitive is defined by several parameters, as listed in Table 1. The advantages of this model-based approach are that it enables the inference of scene structure and geometry where evidence from the images is weak, such as in occluded regions or areas of homogeneous texture, and that it provides an interpretation of the scene as well as its geometry and texture [8]. The representation of the scene as a set of planes and primitives is useful for reasoning about the scene during reconstruction and for subsequent rendering and manipulation of the model. The compactness of the representation also makes recovery of structure and motion more reliable, as demonstrated in [19].

In previous work [8] a framework was defined for model based structure from motion for buildings. In this framework an algorithm for estimating a maximum a posteriori (MAP) estimate of the model based on priors and image likelihood measures was proposed (this is summarized in Section 2). However the spatial prior used in this work applied only to the parameters of each individual primitive, thus ignoring information about their spatial juxtaposition (for instance, that windows are likely to occur in rows and columns). In this paper the spatial prior is expanded to include this sort of information.

The form which the spatial prior should take is far from obvious. Ideally it should admit all plausible buildings while excluding those which are for practical or aesthetic reasons implausible. However the plausibility of a structure can in general only be verified by manual inspection. Thus a crucial step in the formulation of the prior is to test it by drawing sample buildings from it and checking that they appear reasonable. However even with expert knowledge, it is very difficult to explicitly represent the probability density function (pdf) of a suitable prior. What is somewhat easier to do is to express the prior as a scoring function that favours particular configurations, such as windows in rows. One approach is to use a scoring function suggested by an expert and then draw samples from the implicitly defined pdf using an MCMC algorithm. If the samples drawn look like reasonable buildings then the prior must be close to the true prior.

This raises the question of just how close to the true prior our estimate must be to generate reasonable looking models. To answer this empirically the scoring function is varied both on a small scale and a large scale, and the effect on models generated from the prior, and reconstructions obtained using the prior and an image sequence, is observed.

The paper is organized as follows: Section 2 defines an architectural model as a collection of wall planes containing parameterized shapes, and establishes a framework for optimizing it. In Section 3 the spatial prior is discussed and the MCMC algorithm is used to simulate samples from it. Section 4 then presents some reconstructions based

on this prior, and demonstrates the effect of varying the prior on reconstruction. The paper concludes with discussion and ideas for future work in Section 6.

2 Problem formulation

This section briefly recapitulates previous work [8] on the definition of a model for architecture and an algorithm for optimizing it. The projection matrices are initially recovered using point matching and robust methods as described in [2,20]. With the projection matrices recovered the 3D structure of the scene must be parameterized. An architectural model is formulated as a number of base planes (generally walls), each of which contains a number of offset primitives such as doors and windows. Also modelled are the height of the apex of the roof, and the number of floors in the building, as these will affect the window layout. The model M therefore contains parameters $\boldsymbol{\theta} = \{n, \boldsymbol{\theta}_L, \boldsymbol{\theta}_S, \boldsymbol{\theta}_T, \boldsymbol{\theta}_G\}$, where n is the number of primitives in the model, $\boldsymbol{\theta}_L$ identifies the type of each primitive, $\boldsymbol{\theta}_S$ are structure parameters which define its shape, $\boldsymbol{\theta}_T$ are texture parameters describing its appearance, and $\boldsymbol{\theta}_G$ are global parameters describing such things as the number of floors and the style of the building (e.g. Classical, Gothic). The types of primitive available and their shape parameters are given in Table 1. The texture parameters are intensity variables $i(\mathbf{x})$ (between 0 and 255) defined at each point $\mathbf{x}$ on a regular 2D grid covering the model surface [1].

Our choice of primitive reflects the scale of detail in the model. 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 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 pyramid), as most images taken from ground level include very little if any information about the roof structure.

To recover the architectural model we want to maximize

$$\begin{aligned}
&\Pr(\mathbf{M}\boldsymbol{\theta}|\mathbf{DI}) \propto \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) \Pr(\mathbf{M}\boldsymbol{\theta}\mathbf{I}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}\mathbf{I}) \Pr(\boldsymbol{\theta}|\mathbf{MI}) \Pr(\mathbf{MI}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}_L\boldsymbol{\theta}_S\boldsymbol{\theta}_T\boldsymbol{\theta}_G\mathbf{I}) \Pr(\boldsymbol{\theta}_L\boldsymbol{\theta}_S\boldsymbol{\theta}_T\boldsymbol{\theta}_G|\mathbf{MI}) \Pr(\mathbf{MI}) \\
&= \Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}_L\boldsymbol{\theta}_S\boldsymbol{\theta}_T\boldsymbol{\theta}_G\mathbf{I}) \Pr(\boldsymbol{\theta}_T|\boldsymbol{\theta}_L\boldsymbol{\theta}_G\mathbf{MI}) \\
&\quad \Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L\boldsymbol{\theta}_G\mathbf{MI}) \Pr(\boldsymbol{\theta}_L|\boldsymbol{\theta}_G\mathbf{MI}) \Pr(\boldsymbol{\theta}_G|\mathbf{MI}) \qquad (1)
\end{aligned}$$

where D is the available data (the images), I denotes prior information (the camera calibration and the estimated wall planes), and

- $\Pr(\mathbf{D}|\mathbf{M}\boldsymbol{\theta}_L\boldsymbol{\theta}_S\boldsymbol{\theta}_T\boldsymbol{\theta}_G\mathbf{I})$ is the likelihood of the images given a complete specification of the model. This is determined by the deviation of image intensities from those predicted by the texture parameters.

[1] This allows us to specify the model to super resolution; however this aspect is not dealt with in this paper.

Table 1. Some primitives available for modelling classical architecture. Parameters in brackets are optional. 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. $\mathcal{M}_0$ is reserved as the background model (generally a wall).

θ_L^i	Description	Parameters
$\mathcal{M}_1$	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
$\mathcal{M}_8$	Drain pipe	x, w, h
$\mathcal{M}_9$	Floor	y
$\mathcal{M}_{10}$	Roof	h
$\mathcal{M}_{11}$	NULL	$x_1 \ldots x_n, y_1 \ldots y_n, z_1 \ldots z_n$

- $\Pr(\boldsymbol{\theta}_T|\boldsymbol{\theta}_L\boldsymbol{\theta}_G\mathrm{MI})$ is the probability of 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.
- $\Pr(\boldsymbol{\theta}_L|\boldsymbol{\theta}_G\mathrm{MI})$ is the prior probability for each type of primitive. It is used to specify the relative frequency with which primitive types occur, e.g. that windows are more common than doors, or that buttresses appear frequently in Gothic architecture.
- $\Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L\boldsymbol{\theta}_G\mathrm{MI})$ is a prior on shape. The formulation and validation of this prior is described in more detail in Section 3.

The global parameters $\boldsymbol{\theta}_G$ are generally given rather than estimated (for instance, the style of the model is specified manually) and hence the probability $\Pr(\boldsymbol{\theta}_G|\mathrm{MI})$ is fixed at 1 for the given parameters, and 0 elsewhere.

2.1 Obtaining an initial model estimate

The input to our system is an uncalibrated sequence of 3–6 images, in which corner and line features are automatically detected and matched as in [3] to estimate the structure of the building and the motion of the cameras. This reconstruction is then segmented into planes to obtain an initial estimate of the position and orientation of each wall in the building [7].

Ideally the combined $\boldsymbol{\theta}_T$, $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_L$ parameter space would then be searched for MAP parameter values. However each primitive may contain thousands of texture parameters, so only the $\boldsymbol{\theta}_S$ and $\boldsymbol{\theta}_L$ parameter spaces are searched. This is carried out in two steps: an initial search based on an approximate single image likelihood function locates likely values for a subset $\boldsymbol{\theta}_{S1} = \{x, y, w, h\}$ of the shape parameters, while

the remaining shape parameters d, a, b, dw are set to zero. These are then used to seed searches in the full parameter space using the complete likelihood function [8]. Models found using this method are subsequently used as seed points for the MCMC algorithm described in this paper.

3 The shape prior

In this section an architectural shape prior $\Pr(\boldsymbol{\theta}_S|\boldsymbol{\theta}_L, \mathrm{MI})$ is defined and assessed using a Markov Chain simulation. Ideally this prior should encode 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.
- Other spatial relations such as symmetry about a vertical axis.

It is extremely difficult to explicitly formulate a prior pdf to meet this set of desiderata. When it is required to draw samples from a pdf $\Pr(\boldsymbol{\theta})$ which cannot be explicitly defined, a common technique is to simulate the drawing of samples using a Markov process defined on the parameters $\boldsymbol{\theta}$. The Markov process is chosen so that over time (as $t \to \infty$) its transition probability $\Pr(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ converges to the desired distribution $\Pr(\boldsymbol{\theta})$. Therefore as the number of iterations increases, the values of $\boldsymbol{\theta}$ visited by the Markov process mimics independent draws from $\Pr(\boldsymbol{\theta})$ with increasing accuracy. The group of algorithms which operate on this principle are known as Markov Chain Monte Carlo (MCMC) algorithms [10] (Monte Carlo refers to the fact that Markov processes are seeded at many points in parameter space).

There are a number of ways to generate a Markov process with the desired convergence properties. These include the Metropolis-Hastings class of algorithms, in which transitions or jumps are drawn from a user-defined jumping distribution $J_t(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$ and the updated model is accepted or rejected based on a scoring function $f(\boldsymbol{\theta})$. In particular the Reversible Jump [11] Metropolis-Hastings algorithm is used in this paper, as it allows jumps between parameter spaces of varying dimension, as required when primitives are added or removed from the model. This algorithm is summarized in Algorithm 1.

The behaviour of the Reversible Jump MCMC algorithm depends largely on the choice of both the scoring function and the jumping distribution. These are the subjects of the following two sections.

3.1 The scoring function

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

$$f_{prior}(\boldsymbol{\theta}) = f_{scale}(\boldsymbol{\theta}) + f_{shape}(\boldsymbol{\theta}) + f_{align}(\boldsymbol{\theta}) + f_{sym}(\boldsymbol{\theta}) \tag{3}$$

Algorithm 1 Reversible Jump MCMC algorithm.

Draw an initial point θ_0 from a starting distribution $\Pr_0(\theta)$.
for $t = 1..T$ **do**
  Draw candidate point θ_* from the jumping distribution $J_t(\theta_*|\theta_{t-1})$.
  Calculate the ratio

$$r = \frac{f(\theta_*)J_t(\theta_{t-1}|\theta_*)}{f(\theta_{t-1})J_t(\theta_*|\theta_{t-1})} \tag{2}$$

  Set $\theta_t = \theta_*$ with probability min(r,1), otherwise set $\theta_t = \theta_{t-1}$
end for

The shape and scale terms apply only to individual primitives. In this paper the shape and scale components of the scoring function are given by simple functions, some of which are listed in Table 3. The purpose of these components 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. It is defined as

$$f_{align}(\theta) = \sum_{r=1}^{R} [\mathrm{Var}(\mathbf{t}_r) + \mathrm{Var}(\mathbf{b}_r) + \mathrm{Var}(\mathbf{r}_r - \mathbf{l}_r)] \tag{4}$$

$$+ \sum_{c=1}^{C} [\mathrm{Var}(\mathbf{l}_c) + \mathrm{Var}(\mathbf{r}_c) + \mathrm{Var}(\mathbf{t}_c - \mathbf{b}_c)] \tag{5}$$

where $\mathbf{t}_r, \mathbf{b}_r, \mathbf{l}_r, \mathbf{r}_r$ are the top, bottom, left and right coordinates of the 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 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 maximized when a row is exactly centred on a wall, and decays quadratically:

$$f_{sym}(\theta) = \sum_{r=1}^{R} [(l_r - \mathbf{l}) - (r_r - \mathbf{r})]^2 \tag{6}$$

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 symmetric on a wall.

3.2 The jumping distribution

As well as a scoring function, an MCMC algorithm requires the specification of a jumping distribution $J(\boldsymbol{\theta}_t|\boldsymbol{\theta}_{t-1})$. The jumping distribution is a mixture of several types of jump, which are listed in Table 2.

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 reconstruction while maintaining closure.
- For efficiency, it should be easy to sample from the jumping distributions. Each jump should also traverse a significant distance in parameter space, and have a high acceptance rate. Therefore simple jump types which are likely to generate a more probable model should be used.
- Reversible jump MCMC requires that jumps be symmetric; that is, after jumping from $\{\mathbf{M}_1, \boldsymbol{\theta}_1\}$ to $\{\mathbf{M}_2, \boldsymbol{\theta}_2\}$ it should always be possible to return to $\{\mathbf{M}_1, \boldsymbol{\theta}_1\}$ in a single jump. More precisely, given a jump type $\mathcal{J}_{i1}$ which moves from the parameter space $\boldsymbol{\theta}_1$ to $\boldsymbol{\theta}_2$ based on the values of $\boldsymbol{\theta}_1$ and some extra random variables $\boldsymbol{\phi}_1$, there must be a reverse jump type $\mathcal{J}_{i2}$ which moves from $\boldsymbol{\theta}_2$ to $\boldsymbol{\theta}_1$ based on $\boldsymbol{\theta}_2$ and $\boldsymbol{\phi}_2$, where the dimension of $\boldsymbol{\theta}_1 \oplus \boldsymbol{\phi}_1$ equals that of $\boldsymbol{\theta}_2 \oplus \boldsymbol{\phi}_2$. 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.

Table 2. 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

3.3 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 1: 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 is iterated for 2000 jumps. After this period, samples

Fig. 1. *"Hut", "tower" and "bungalow" seed points for the MCMC algorithm.*

are drawn at random from each chain and displayed in Figure 2. 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.

Fig. 2. *Collection of Classical style buildings generated from the shape prior.*

4 Results for a single wall

Some results of the MCMC algorithm are now shown for a single wall. The scoring function is altered to include image information by adding a likelihood term, so that the

complete score is now given by

$$f(\theta) = \lambda f_{prior}(\theta) + \sum_{i(\mathbf{x})} \sum_{j=1}^{m} \left(\frac{i(\mathbf{x}^j) - i(\mathbf{x})}{\sigma} \right)^2 \tag{7}$$

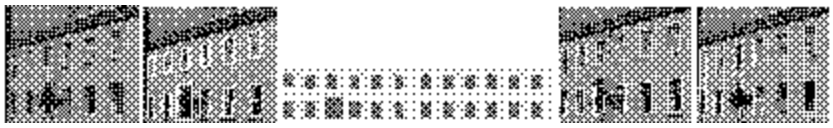
where $i(\mathbf{x}^j)$ is the projection of the texture parameter $i(\mathbf{x})$ onto the jth image, σ is an image variance parameter and λ is a relative scale factor. The global parameter set θ_G is reduced to contain simply a style parameter, as the number of floors can be deduced from the images.

Starting points for the algorithm are generated from the results of a previously developed algorithm for finding a MAP model estimate, described in Section 2.1. It is first tested on images of the Downing library, one of which is shown in Figure 3(a), which contains strong information and complies with our priors on regularity and shape. Because the our old algorithm incorporated priors only on individual primitives, the windows estimated, although approximately correct, are slightly misaligned. Furthermore the window obscured by the tree is incorrectly fitted. However the new global shape prior described in this paper significantly improves matters: After running the MCMC algorithm for 2000 iterations, with $\lambda = 10^4$ and $\sigma = 10$, the prior has overridden local likelihood maxima and the MAP model is one in which the windows are properly aligned, even where the window is occluded in some images by a tree (Figure 3(d)). To test the sensitivity of this result to the choice of prior, a Gothic shape prior is substituted for the original Classical shape prior. Although the Gothic prior favours narrower and more arched windows, the same result is obtained (Figure 3(e)). In Figure 3(f) a very different prior is proposed in which very tall narrow windows are strongly favoured (Table 3). This results in a different interpretation of the scene in which vertically aligned windows are merged.

Table 3. Relevant shape priors. H and W are the height and width of the wall containing the primitive. $G(\mu, \sigma)$ is a Gaussian distribution with mean μ and standard deviation σ.

Primitive	Param.	Score
Window (Classical)	h/w	$-(G(0.75, 0.25) + G(1.0, 0.25) + G(1.25, 0.25) + G(1.5, 0.25) + G(1.75, 0.25) + G(2.0, 0.25))$
Window (Gothic)	h/w	$-(G(2.0, 0.25) + G(2.5, 0.25))$
Window (Narrow)	h/w	$-(G(4.0, 1.0) + G(6.0, 1.0))$
Window	y/H	0, if $0.1 \leq y/H \leq 0.9$. Else LARGE.
Window	x/W	0, if $0.1 \leq x/W \leq 0.9$. Else LARGE.
Window	h/d	0, if $-0.2 \leq h/d \leq 0.2$. Else LARGE.

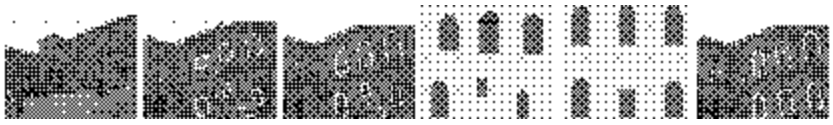
The next image sequence, of the Palazzo Pitti, one image of which is shown in Figure 4(a), is more challenging. The images were taken on a rainy day, and the window



(a) (b) (c) (d) (e) (f)

Fig. 3. (a) One image of the wall. (b) Model obtained using only Classical single primitive prior. (c) Frontal view of this model (windows are darker, background is light). (d) Model obtained using MCMC and Gothic shape prior. (e) Same model, projected onto image. (f) Incorrect model obtained using prior favouring very narrow windows.

texture is indistinct from wall texture—the windows and wall are a similar colour, and the brick texture of the wall is not easily distinguished from that of the windows. Due to the lack of information in the images, the reconstruction obtained is dependent on whether a Classical or Gothic prior is used. Using Classical priors (Figure 4(b)), only the interior parts of the top windows are detected, whereas a Gothic prior (Figure 4(c)) detects the surrounding arch structure. Applying the full set of shape priors with $\lambda = 10^4$, $\sigma = 10$ results in a MAP model where the windows are aligned despite this not being the case for the bottom row of windows. To prevent this from occurring, λ can be reduced to 1 in which case the prior has no significant effect on the model.



(a) (b) (c) (d) (e) (f)

Fig. 4. (a) One image of the Palazzo Pitti. (b) Model obtained using Classical priors. (c) Model obtained using Gothic priors. (d) Primitives before MCMC. (e) After MCMC. (f) Projected onto image.

5 Building Results

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 5. 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. Another 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 5 (i)-(l).

The operation of this algorithm is shown in Figure 6 for the Trinity Chapel sequence. Note that the entire model is obtained from only 3 images. 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. 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.

5.1 Comparison with ground truth

It is difficult to directly compare the results of this algorithm with previous methods, as no other method solves quite the same problem as this one. In the absence of experimental results with which to compare it, some ground truth measurements were taken from the Downing College library, reconstructed in Figure 5.

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 7. 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 ± 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 to depth, are recovered to within the accuracy bounds

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 ± 1cm accuracy for ground truth measurements, except for the circumference of the base of the column, which is measured to ± 10cm. The accuracy of the model measurements is limited to the resolution of the texture parameters.*

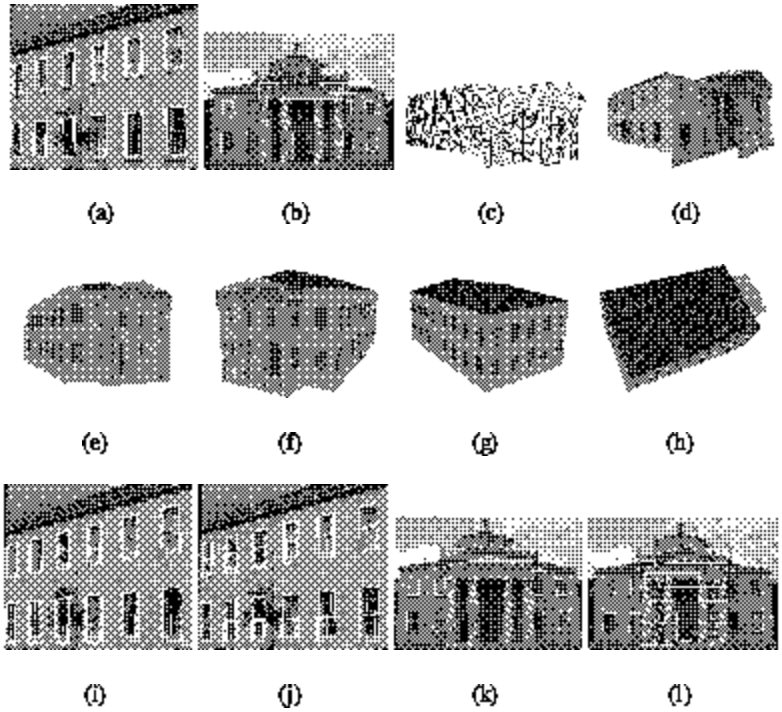


Fig. 5. *(a) MAP model of side wall of Downing library, after 2000 MCMC iterations using the Classical prior. (b) Front wall. Both front and back faces of primitives are drawn, hence the pair of triangles and rectangles for the pediment and entablature. (c)-(d) 3D rendering of MAP Downing model, obtained 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) 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. (i)-(l) Some ambiguities in the Downing model, chosen from the 20 most probable models visited by the MCMC process. (i) Window sills are included in the window primitives. (j) Windows are represented using two primitives each. (k) The door is omitted. (l) Extra columns are added in between the existing ones.*

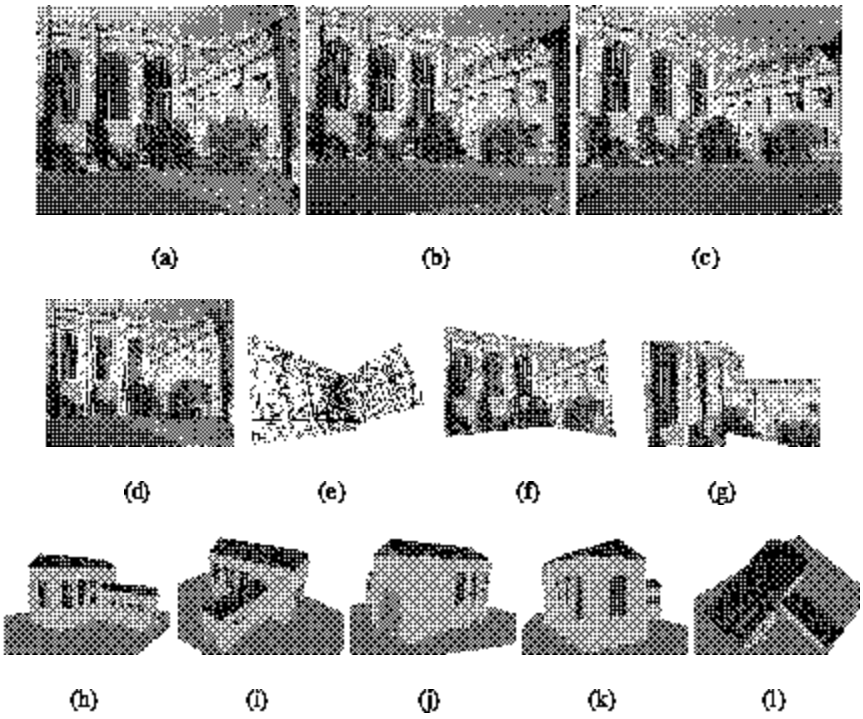


Fig. 6. *(a)-(c) 3 original images of Trinity college courtyard. (d) MAP model primitives, superimposed on image. (e) Wireframe MAP model. (f)-(g) 3D model with texture taken from images. (i)-(m) Five views of the completed model of the north-east corner of Great Court, Trinity College. Only two of the walls are visible in the images.*

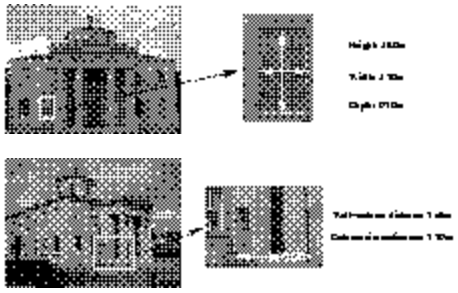


Fig. 7. *Some measurements made of the Downing Library scene. The other windows in the scene are the same size as the one shown (to an accuracy of ±1cm).*

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. 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 ± 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 ± 0.5cm.

6 Conclusion

If structure from motion algorithms are to progress they must find ways of incorporating more and higher level prior information concerning the nature of the world. To effectively use this information, recognition of what we are observing will play a crucial rôle. Recovering structure from visual input alone is highly ill conditioned, thus is it envisaged that a robot of the future might carry many prior models in its head and recognize which class of priors is appropriate to reduce the ambiguity in resolving a particular scene. Thus classic geometric structure from motion becomes a blend of learning, classification and geometry. This paper has presented a framework for representing one such spatial prior for the case of architecture. Samples are generated from the prior and shown to be reasonable instances of genuine buildings. A Bayesian framework is a natural way to effectively use the prior information to enhance 3D reconstruction in a variety of ways. Future work includes the parametrization of texture in a similar way to shape, i.e. using only a few texture parameters per primitive, which would also result in a super resolution of the images. As previously mentioned, the roof structure is currently modelled simply as a pyramid; however more complex roof models could improve the appearance of the model from elevated viewpoints.

The general philosophy underlying this paper is that the state of the art has been reached with existing structure from motion methods, and that the best route for progress is to combine structure from motion with recognition. This allows the use of strong prior models of shape, stronger than the markov random fields traditionally used. To some extent this has been done in the past with simple shape models e.g. [4, 6, 13], but these models possess only limited variability. Within this paper we have attempted to present a generic framework which could be used to optimize classes of objects that possess a much greater variability of shape, but that can be decomposed into a 'lego kit' of parameterizable parts.

Software: It is intended to release a Matlab SFM toolkit to illustrate some of the methods described, please check http://research.microsoft.com/ philtorr/, as the release is aimed to coincide with ECCV.

References

1. M.E. Antone and S. Teller. Automatic recovery of relative camera rotations for urban scenes. In *CVPR00*, pages II:282–289, 2000. See also http://city.lcs.mit.edu.
2. P. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In B. Buxton and Cipolla R., editors, *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695. Springer–Verlag, 1996.
3. 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.
4. T.F. Cootes, G.J. Edwards, and C.J. Taylor. Active appearance models. *IEEE PAMI*, 23(6):681–685, 2001.
5. I.J. Cox, S.L. Hingorani, S.B. Rao, and B.M. Maggs. A maximum-likelihood stereo algorithm. *Computer Vision and Image Understanding*, 63(3):542–567, 1996.
6. D. Cremers, C. Schnoerr, and J. Weickert. Diffusion-snakes: Combining statistical shape knowledge and image information in a variational framework. In *1st IEEE Workshop on Variational and Level Set Methods in Computer Vision*, 2001.
7. A.R. Dick, P.H.S. Torr, and R. Cipolla. Automatic 3d modelling of architecture. In *Proc. 11th British Machine Vision Conference (BMVC'00)*, pages 372–381, Bristol, 2000.
8. A.R. Dick, P.H.S. Torr, S. Ruffle, and R. Cipolla. Combining single view recognition and multiple view stereo for architectural scenes. In *Proc. IEEE International Conference on Computer Vision*, pages 1:268–274, 2001.
9. P. Fua. Regularized Bundle-Adjustment to Model Heads from Image Sequences without Calibration Data. *International Journal of Computer Vision*, 38(2), July 1999.
10. W. Gilks, S. Richardson, and D. Spiegelhalter (editors). *Markov Chain Monte Carlo in Practice*. Chapman and Hall, London., 1996.
11. P. Green. Reversible jump markov chain monte carlo computation and bayesian model determination. *Biometrika*, 82:711–732, 1995.
12. E. T. Jaynes. *Probability Theory: The Logic of Science*. Unpublished but available online at http://bayes.wustl.edu/etj/prob.html, 1996.
13. M. Leventon, E. Grimson, and O. Faugeras. Statistical Shape Influence in Geodesic Active Contours. In *CVPR*, pages 316–323, 2000.
14. D. Nister. Frame decimation for structure and motion. In *2nd European Workshop on 3D Structure from Multiple Images of Large Scale Environments (SMILE 2000)*, pages 17–34, 2000.
15. R. Plänkers and P. Fua. Articulated Soft Objects for Video-based Body Modeling. In *International Conference on Computer Vision*, Vancouver, Canada, July 2001.
16. M. Pollefeys, R. Koch, M. Vergauwen, and L. van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In *1st European Workshop on 3D Structure from Multiple Images of Large Scale Environments (SMILE 1998)*, pages 139–155, 1998.
17. D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1):7–42, 2002. Evaluation page http://www.middlebury.edu/stereo/eval/.
18. Y. Shan, Z. Liu, and Z. Zhang. Model based bundle adjustment with applications to face modeling. In *ICCV Vol 2*, pages 644–651. IEEE, 2001.
19. 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.
20. P. H. S. Torr and D. W. Murray. The development and comparison of robust methods for estimating the fundamental matrix. *Int Journal of Computer Vision*, 24(3):271–300, 1997.