

Bayesian Stochastic Mesh Optimisation for 3D Reconstruction

George Vogiatzis¹ Philip Torr² Roberto Cipolla¹

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

² Microsoft Research, 1 Guildhall St, Cambridge CB2 3NH, UK

gv215@eng.cam.ac.uk philtorr@microsoft.com cipolla@eng.cam.ac.uk

Abstract

We describe a mesh based approach to the problem of structure from motion. The input to the algorithm is a small set of images, sparse noisy feature correspondences (such as those provided by a Harris corner detector and cross correlation) and the camera geometry plus calibration. The output is a 3D mesh, that when projected onto each view, is visually consistent with the images. There are two contributions in this paper. The first is a Bayesian formulation in which simplicity and smoothness assumptions are encoded in the prior distribution. The resulting posterior is optimized by simulated annealing. The second and more important contribution is a way to make this optimization scheme more efficient. Generic simulated annealing has been long studied in computer vision and is thought to be highly inefficient. This is often because the proposal distribution searches regions of space which are far from the modes. In order to improve the performance of simulated annealing it has long been acknowledged that choice of the correct proposal distribution is of paramount importance to convergence. Taking inspiration from RANSAC and importance sampling we craft a proposal distribution that is tailored to the problem of structure from motion. This makes our approach particularly robust to noise and ambiguity. We show results for an artificial object and an architectural scene.

1 Introduction

Dense stereo is a well studied problem in Computer Vision. A good survey of methods can be found in [11]. The output of most stereo algorithms is often a per pixel depth map obtained by using correlation-based methods together with simple first order Markovian assumptions on the pixels of one of the images. However, for true 3D reconstruction what is often required is information about surfaces in the scene. One way to obtain this would be to use correlation-based methods to identify features that have good correspondence and then triangulate between them to obtain a mesh based representation [7].

Polygonal meshes are one of the simplest models that have extensively been used in graphics and vision applications to represent surfaces and 3D objects. There is a growing volume of research on techniques for rendering, simplifying, compressing and transmitting meshes in the graphics community. Many of the algorithms necessary for rendering and handling texture mapped meshes have been implemented in extremely fast hardware which means that a mesh can be used to generate images of the scene under varying lighting and shading conditions. Furthermore mesh data structures can be used by many applications such as VRML. All this makes them a very appealing choice for 3D object representation.

However a problem with the approach of just triangulating between matched feature points is that information about smoothness is not included in the fitting cost. Often a small error in the location of a vertex will lead to a large change in the 3D position leading to ‘spikes’ appearing in the reconstruction. Furthermore the triangulation might ill fit the images in the planar facets formed between well matched points.

There is a substantial volume of work in the graphics literature on the topics of mesh simplification and optimisation. Simplification algorithms take a mesh with a few thousand vertices, usually the output of very accurate range measurements, and produce one with a few hundred vertices, which looks the same based on some particular metric [8, 9]. Mesh optimisation involves [4] optimising the vertices and connectivity of a mesh to achieve a better geometric fit to a 3D structure. However the work in the graphics literature, whilst giving some useful hints to the solution, can not be directly used for our problem, as in their case the 3D structure of the vertices is assumed known (e.g. from a laser range finder), whereas in our case we are case we know nothing about 3D.

In the vision literature, there is only one approach which considers unknown 3D which is that of Fua and Leclerc [2], who give a method for surface reconstruction based on a constrained mesh structure. The metric optimised is a sum of terms relating to stereo correspondence, shading and geometric smoothness information that is calculated at sample points on the mesh. The weights of the three terms vary across the surface, the stereo term being more significant where the surface is more textured while relying on shading in uniform areas. A gradient descent algorithm is used. As mentioned in that work, their method converges only when the mesh samples project a few pixels away from their true positions in the images. Morris and Kanade [1] keep vertices of the mesh fixed and only search for good triangulations using edge swaps. Their algorithm performs all possible edge swaps at each step and uses the one producing the maximum decrease in cost, terminating when there are no cost-reducing swaps. It relies heavily upon a carefully selected set of vertices on edges and corners and is unable to cope with local minima in the search space. Two more recent attempts using deformable meshes can be found in [15, 16]. Both these approaches proceed by iteratively shrinking and deforming a mesh until it is visually consistent with the images. Finally, although not using meshes, related to this paper is the work of Dick *et al* [5] in which architectural models are automatically reconstructed from multiple images. They use a Bayesian prior to encode strong geometric constraints inherent to architectural scenes.

The problem of finding the best mesh to represent a surface is unlikely to have a solution in polynomial time. The complexity and non-concavity of the search-space introduces local minima which make gradient based techniques difficult to use. The work of Hoppe [4] in the graphics literature suggests that stochastic algorithms that propose random perturbations of the mesh are the most successful, however he does not follow a formal statistical approach. Simulated annealing is a probabilistic technique that has been used successfully against similar problems because its random search approach can jump out of local minima. When the search is purely random however, the speed of convergence can be low. It is thus necessary to have a more intelligent strategy that takes account of the data to guide the search. Additionally, the ambiguity and noise present in the data suggests that a regularisation term in the form of a mesh prior would be useful.

This paper proposes a simulated annealing strategy and outlines a proposal distribution that is shown to accelerate the optimisation compared to a purely random search. Our algorithm assumes that epipolar geometry for the images has been previously recovered

by another process (e.g. [12]). We start by running a structure from motion algorithm to obtain a cloud of 3D points. We then project these points on one of the images and evaluate their 2D Delaunay triangulation. This becomes the starting point of the simulated annealing scheme which maximises a posterior probability measure of the mesh given the image data. In this measure, a prior term is included which assigns greater belief in largely planar surfaces with as few vertices as possible.

The rest of the paper is laid out as follows. Section 2 describes the probabilistic formulation of the posterior likelihood, the negative log of which is the cost for a particular configuration. Section 3 describes the simulated annealing algorithm used for finding the minimum cost configuration. Section 4 describes our suggested improvements to the proposal distribution, the benefits of which are shown in Section 5.

2 Bayesian Mesh Model

We are given a set of n images $\mathbf{I} = \{I_1, \dots, I_n\}$, and n projection mappings m_1, \dots, m_n , which define the transformation from points on the mesh to the coordinate system of each of the images. In order to define these mappings it is assumed that the camera positions and calibration have been recovered by some other method. Our mesh model is $\mathbf{M} = (\mathbf{T}, \mathbf{E}, \mathbf{V})$ where \mathbf{V} is the set of 3D vertices and \mathbf{E} is a set of edges connecting members of \mathbf{V} . We also let \mathbf{T} be a texture map mapping points on the surface of the mesh to intensity values. For a given mesh \mathbf{M} , $\mathbf{J} = \{J_1, \dots, J_n\}$ is the set of images of \mathbf{M} taken from the viewpoints of the images in \mathbf{I} .

We are interested in the posterior probability of a particular mesh structure given the image data. Using Bayes' theorem this can be expressed as:

$$p(\mathbf{T}, \mathbf{E}, \mathbf{V} | \mathbf{I}) \propto p(\mathbf{I} | \mathbf{T}, \mathbf{E}, \mathbf{V})p(\mathbf{T} | \mathbf{E}, \mathbf{V})p(\mathbf{E}, \mathbf{V}). \quad (1)$$

Equation (1) has two distinct parts, the likelihood $p(\mathbf{I} | \mathbf{T}, \mathbf{E}, \mathbf{V})$ and the prior $p(\mathbf{T} | \mathbf{E}, \mathbf{V})p(\mathbf{E}, \mathbf{V})$ which are now described in more detail.

2.1 Likelihood Term

We assume that each pixel intensity in each image I_k is normally and independently distributed about the corresponding image intensity in J_k . This gives us the following likelihood measure

$$p(\mathbf{I} | \mathbf{T}, \mathbf{E}, \mathbf{V}) \propto \exp \left(- \sum_{k=1}^n \sum_{x,y} \frac{1}{\sigma^2} (I_k(x,y) - J_k(x,y))^2 \right). \quad (2)$$

In this work we will be making no prior assumptions on the texture of the mesh so in fact $p(\mathbf{T} | \mathbf{E}, \mathbf{V})$ will be uniform. Following [1] we will treat the texture \mathbf{T} as a *nuisance* parameter and marginalise it out of (1). Because of the normality of the image intensities we can obtain $p(\mathbf{I} | \mathbf{E}, \mathbf{V})$ by substituting $J_k(x,y)$ in (2) by the weighted average:

$$\hat{J}_k(x,y) = \frac{\sum_{i=1}^n I_i(m_i m_k^{-1}(x,y, \mathbf{E}, \mathbf{V})) V_i(m_k^{-1}(x,y, \mathbf{E}, \mathbf{V}))}{\sum_{i=1}^n V_i(m_k^{-1}(x,y, \mathbf{E}, \mathbf{V}))}, \quad (3)$$

where $m_i^{-1}(x, y, \mathbf{E}, \mathbf{V})$ is the 3D point (X, Y, Z) on the surface of the mesh (\mathbf{E}, \mathbf{V}) such that $m_i(X, Y, Z) = (x, y)$. The function $V_i(X, Y, Z)$ is a visibility map that returns 1 when point (X, Y, Z) is unoccluded in image i and 0 otherwise. The simplified version of the posterior is:

$$p(\mathbf{E}, \mathbf{V} | \mathbf{I}) \propto p(\mathbf{I} | \mathbf{E}, \mathbf{V})p(\mathbf{E}, \mathbf{V}). \quad (4)$$

One of the benefits of our mesh based approach is that the projections $\hat{J}_1, \hat{J}_2, \dots, \hat{J}_n$ can be efficiently computed in graphics hardware using projective texture mapping. Conceptually this is similar to viewing the cameras that produced images I_1, I_2, \dots, I_n as slide-projectors, projecting these images onto the mesh. This lit mesh is now photographed from the viewpoint of image k with camera m_k , and the result is \hat{J}_k . We do this with a multipass rendering of the mesh from the k -th viewpoint. At each pass, each of the images I_i is used as a texture map, and m_i is set as the texture transformation. The multiple textures are then averaged using alpha-blending.

2.2 Shape Prior Term

The shape prior distribution encapsulates our knowledge about mesh surfaces and gives us a way to cope with ambiguity that is inherently present in the surface reconstruction problem. No matter how precise our measurements, 2D images always represent incomplete information that can usually be interpreted in several different ways. In order to be able to make reasonable inferences about what we see in them, we must regularise all the possible interpretations so that we keep just the plausible ones. In this preliminary work we have chosen a simple shape prior that qualitatively expresses our belief in smooth, well-behaved surfaces rather than rugged and spiky ones.

In particular we will assume that in the absence of other information, two neighbouring mesh faces will tend to be coplanar. Also we will assume that everything else being equal, the mesh with the fewest vertices is more plausible.

To quantify these assumptions we define $NB(\mathbf{E}, \mathbf{V}) \subseteq \mathbf{E}$ as the set of non-boundary edges of the mesh (i.e. the edges that belong to only one face). Now for $e \in NB(\mathbf{E}, \mathbf{V})$ we let $\vartheta(e)$ be the angle between the outward normals of the two faces that share e as a common edge. This implies that if n_1 and n_2 are the two normals then $\cos \vartheta(e) = n_1 \cdot n_2$. We can now define the prior by:

$$p(\mathbf{E}, \mathbf{V}) \propto \exp \left(-A \text{size}(\mathbf{V}) - B \sum_{e \in NB(\mathbf{E}, \mathbf{V})} (1 - \cos \vartheta(e)) \|e\| \right), \quad (5)$$

where $\|e\|$ denotes the length of edge e .

3 Algorithm for finding the MAP estimate

Having specified all the components of the posterior distribution $p(\mathbf{E}, \mathbf{V} | \mathbf{I})$ in (4) we use the simulated annealing scheme summarised in Alg. 1 to obtain a MAP estimate. The random search element of the algorithm provides a means of escaping local maxima. We set $(\mathbf{E}_0, \mathbf{V}_0)$ as the starting point of the mesh structure and then a series of moves or *jumps* that change the connectivity and vertex locations of the mesh are proposed and evaluated.

Algorithm 1 Simulated annealing

 $(\mathbf{E}, \mathbf{V}) := (\mathbf{E}_0, \mathbf{V}_0)$ $k := 1$ *Loop until convergence* $K := C/\log(1+k)$ *Sample from a proposal distribution; $J((\mathbf{E}', \mathbf{V}') | (\mathbf{E}, \mathbf{V}))$, that takes us from (\mathbf{E}, \mathbf{V}) to $(\mathbf{E}', \mathbf{V}')$* *Set the importance ratio $\eta := \left[\frac{p(\mathbf{E}', \mathbf{V}' | \mathbf{I})}{p(\mathbf{E}, \mathbf{V} | \mathbf{I})} \right]^{1/K}$.**With probability $\min\{1, \eta\}$ set $(\mathbf{E}, \mathbf{V}) := (\mathbf{E}', \mathbf{V}')$ and $k := k + 1$
 (\mathbf{E}, \mathbf{V}) is the MAP estimate.*

As some of the most successful of those proposals get accepted, the mesh gradually moves towards a MAP estimate.

The exit condition of Alg. 1 is met when the change of (\mathbf{E}, \mathbf{V}) has remained small for a number of iterations. The mesh converges to the equilibrium distribution which is concentrated around the mode of our posterior. The *annealing temperature* K is a measure of the variance of the equilibrium distribution. When it is large, the speed of convergence increases in expense of the accuracy of the final solution. When it is small, we get a more accurate solution but it will take much longer to reach it. K starts from a large value and is gradually decreased according to the *annealing schedule* $K = C/\log(1+k)$ at the k -th iteration [10], allowing the system to *cool* as it approaches the optimum. C is a constant.

The main limitation of the simulated annealing algorithm is that for a high-dimensional search space such as the one we are faced with (typically hundreds of vertices), the speed of convergence can be quite low. This is because the algorithm is essentially doing a blind search in this vast space and jumps that lead us closer to the optimum are rare. The way to remedy that is to improve the proposal distribution by letting it make more sensible proposals as we outline in the next section.

4 Proposal distribution

Gelman *et al* [13] defines the following properties necessary for a good proposal process: 1. For any (\mathbf{V}, \mathbf{E}) , it is easy to sample from $J((\mathbf{V}', \mathbf{E}') | (\mathbf{V}, \mathbf{E}))$. 2. It is easy to compute the importance ratio η . 3. Each jump moves a reasonable distance in parameter space (otherwise the random walk moves too slowly). 4. The jumps are not rejected too frequently (otherwise the random walk spends too much time standing still). 5. In addition it must also be possible to access any part of the parameter space by a series of proposals starting from any other.

We suggest an ensemble of proposals that satisfy these desiderata. They fall into several classes. First there are the vertex moving jumps (*Random jump*, *Gradient jump* and *Correlation jump*) and second the mesh connectivity jumps (*Edge swap*, *Edge removal* and *Vertex splitting*). All of the jumps are of the form that satisfies desiderata 1 and 2. The *Random jump* allows large movements across parameter space (condition 3). Noting that the key to the success of probabilistic algorithms like RANSAC [6] is the use of small samples of the data itself in the generation of the hypotheses, the *Gradient jump* and *Correlation jump* exploit this insight in order to achieve desiderata 4. The mesh con-

nectivity jumps (*Edge swap*, *Edge removal* and *Vertex splitting*) are present to ensure 5 is satisfied¹. They also ‘tunnel’ through large distances in parameter space in order to escape local minima (satisfying 3). All these jumps are now described in more detail.

Vertex Jumps: In the following three jumps (Fig. 1a) we pick a vertex $v \in \mathbf{V}$ at random and we generate a 3D displacement vector dx in one of three possible ways. We then replace that vertex by $v + dx$. We have tested and compared three types of vertex moving jumps. The first is completely random while the other two use the image data to form a proposal.

Random jumps. In a random jump, dx is a random vector normally distributed about 0. This jump takes no account of the data and provides a way of escaping from the basin of a local maximum.

Gradient jumps. For this type of jump we evaluate the derivatives of an estimate of the negative log likelihood. Partially following Fua [2] we define a set of sample points $S = \{x_1, x_2, \dots, x_m\}$ on the faces of the mesh that have v as a vertex. If we define $u_{kl} = I_k(m_k(x_l))$ then an approximation to the negative log likelihood sampled over S is (up to proportionality)

$$\varepsilon = \sum_{k=1}^n \sum_{l=1}^m (u_{kl} - \frac{1}{n} \sum_{i=1}^n u_{il})^2 \quad (6)$$

We then evaluate the derivatives of this expression with respect to the x , y and z coordinates of v . The derivatives can be written as functions of the image spatial derivatives $\frac{\partial I_k(u,v)}{\partial u}$ and $\frac{\partial I_k(u,v)}{\partial v}$ which are approximated as finite differences. We let dx be a small vector along the direction of $-\nabla \varepsilon$. This type of proposal leads to good convergence but only when we are already in the vicinity of a maximum. The reason is that when the sample points project more than a few pixels away from their true positions, the spatial image derivatives no longer provide significant information.

Correlation jumps. This type of jump uses correlation information as well as the epipolar constraint between the images to propose a new vertex position. It is well known that cross-correlation gives strong feature matches between images and one would expect it to produce sensible proposals for placing vertices. We select one of the images as the base image in which the projection of v will be kept fixed. We then take a $(2w+1) \times (2w+1)$ window centered at the projection of v in the base image and do a 1D search along the epipolar lines in the other images to find the point of maximum correlation which gives us a proposed new point v' . To avoid the effect of outliers we also limit the 1D search to a $2d$ pixel wide interval centred on the projection of v on the other images. The proposed displacement is just $dx = v' - v$. In feature matching stereo algorithms a cross-correlation search along the epipolar line is carried out giving a global maximum which sometimes leads to an erroneous correspondence. This is usually the cause of spikes along the line of sight frequently seen in correlation based reconstructions (Fig. 4 top left). Our strategy proposes a local cross-correlation maximum but the other jumps will ensure that other local maximums are also visited. The erroneous proposals will be discarded as the surface they generate will not be smooth or not consistent with the images.

Connectivity Jumps: The following three jumps, first used in [4] (Fig. 1b,c,d) tunnel through great distances in the search space, ensuring that many different configurations are visited and tested. They transform the mesh by locally changing the connectivity of vertices. The latter two are also the means of removing and adding vertices to the mesh.

¹For a proof see [14].

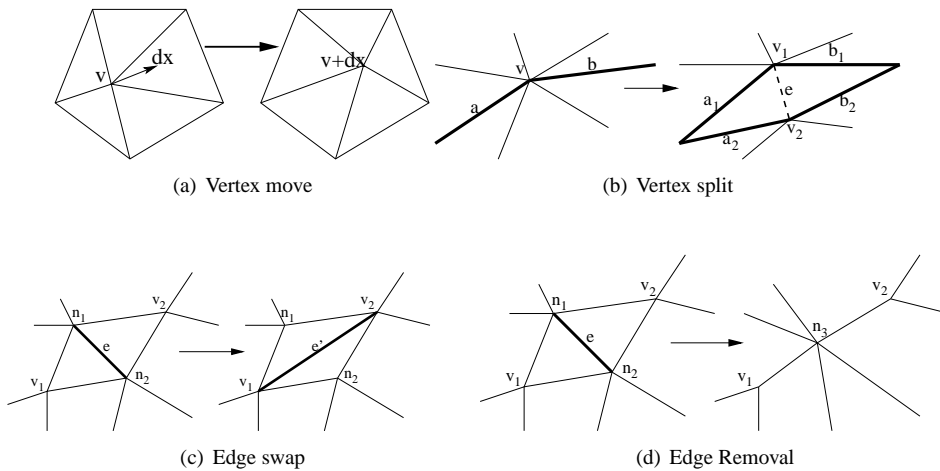


Figure 1: **Mesh Jumps.** Examples of the jumps proposed in the annealing scheme used by our algorithm.

Edge Swap. Edge e joining vertices n_1 and n_2 is replaced by e' joining v_1 and v_2 .

Edge Removal. Edge e is removed, and its vertices n_1 and n_2 are replaced by new vertex $n_3 = \frac{1}{2}(n_1 + n_2)$.

Vertex Split. Vertex v is replaced by vertices v_1 and v_2 and new edge e joining them is introduced. Also edges a and b connected to v are picked at random and split to a_1, a_2, b_1, b_2 . Having ordered the edges that come out of v clockwise, we determine the sets of vertices that must be connected to v_1 and v_2 respectively.

5 Results

The experiments described in this section were performed on a 2.4GHz Intel Pentium with GeForce4 Ti 4200 graphics and 512MB of memory. Each took approximately 10 minutes of running time. The parameters used were $\sigma = 1, A = 3, B = 1, C = 0.4, w = 2$ and $d = 5$.

The first experiment (Fig. 2) demonstrates the simplification aspect of the algorithm. We artificially generated two images of a scene of a texture mapped cube. As a starting point we randomly generate 300 vertices on the faces of the cube and add random Gaussian displacements to those points. The algorithm needs approximately 2500 iterations² to converge to a greatly simplified version of the cube with just 8 vertices, with 7 vertices being of course the minimum necessary number needed to represent the 3 faces of the cube.

In the second experiment we have used a real scene of the Great Court at Trinity College, Cambridge. Three images of the scene (Fig. 3) are used and the starting point of the algorithm is a set of 300 points obtained through a standard feature matching Structure

²We count an iteration when a proposal is accepted. For 3500 iterations, approximately 10000 proposals were needed.

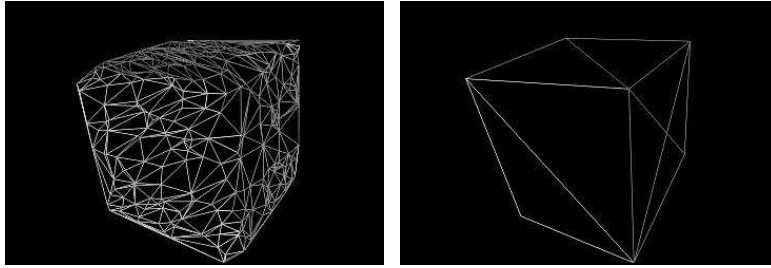


Figure 2: *Simplification of Artificial Scene (textured cube)* On the left is the initial mesh with 300 vertices and on the right is the simplified version after convergence with just 8 vertices.



Figure 3: *Trinity College Chapel and Porter's Lodge.* The three images used in the reconstruction presented in this paper.

from Motion Algorithm [7]. Fig. 4 shows both the initial coarse mesh as well as views of the final reconstructed model. The optimisation is particularly successful in highly textured areas of the scene. The buttresses on the side of the chapel have been captured quite well while areas such as the roof that are quite untextured and poorly described by the input images are less accurate.

Comparison of proposal strategies In order to test the performance of our proposal strategies against each other we ran the algorithm for the real scene three times, using each strategy alone. The gradient strategy does not perform significantly better than the random strategy taking approximately 2500 iterations to converge. The correlation strategy however performs significantly better (Fig. 5) as it only requires approximately 1200 iterations to reach the same level of convergence. We also measured the rate at which proposals of the three types are accepted and we found that the random proposal is accepted 20.2% of the times it is made, the gradient strategy 30.5% and the correlation strategy 47.4% which again demonstrates the significance of the correlation strategy with respect to desiderata 4.

6 Conclusions and future work

We have described an algorithm for optimising a triangular mesh model to fit a 3D scene. This is a relatively under studied problem in Computer Vision. The algorithm presents an improvement on existing similar techniques [1, 2] as it is a more general approach, it avoids getting trapped in local minima and performs well even when initialised well away from the real solution. We cast the problem in a Bayesian framework and use a simulated annealing strategy to perform the optimisation. The Bayesian framework lets us express properties of surfaces through a mesh prior which helps the system cope with ambiguity. Simulated annealing allows for the avoidance of local minima which a gradient descent system could not escape.

Our algorithm achieves good results under sparseness of input data and heavy presence of noise, both in the reconstruction of a scene from a coarse initial mesh, as well as in the simplification of a mesh model. We have demonstrated that an intelligent proposal distribution strategy such as the correlation proposal dramatically outperforms a random proposal mechanism. This suggests that a fruitful line of research is to find better proposal distributions for instance based on edge matching and plane fitting.

Another avenue for future work is to explore the effects of different types of prior and look for suitable priors for each 3D scene domain like architecture or sculpture. In this paper we have used a relatively simple prior that encodes local geometry features in the neighborhood of each vertex. It would be interesting to investigate the effect of more complex priors that model longer range interactions across different regions of the mesh. One could, for example, force areas of similar texture to have similar curvature or orientation in space. Our method in its current form also requires specifying a number of weight-parameters for the different parts of the prior. In the future we would like to explore the opportunity of learning these parameters by analysing models of real scenes.

References

- [1] D. Morris and T. Kanade. *Image-Consistent Surface Triangulation*. CVPR 2000, Vol. 1, pp. 332-338, June 2003
- [2] P. Fua and Y. G. Leclerc. *Object-centered surface reconstruction: Combining multi-image stereo and shading*. Int. J. Comp. Vision (1995), 16(1), 35-56
- [3] P. Lindstrom, G. Turk. *Image-Driven Mesh Optimisation*. Tech. rep. GIT-GVU-00-16, June 2000
- [4] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. *Mesh Optimisation*. Computer Graphics SIGGRAPH '93 Proceedings, pp. 19-26
- [5] A. Dick, P. Torr, S. Ruffle and R. Cipolla. *Combining single view recognition and multiple view stereo for architectural scenes*. ICCV 2001 pages I:268-274, 2001
- [6] M.A. Fischler and R.C. Bolles. *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. CACM, pages 24(6):381-395, June 1981.
- [7] P.A. Beardsley, P.H.S. Torr, and A. Zisserman. *3D model acquisition from extended image sequences*. ECCV 1996, pages II:683-695, 1996
- [8] M. Garland and P. Heckbert. *Simplifying surfaces with color and texture using quadric error metrics*. IEEE Visualization '98, pages 263-270. October 1998.
- [9] H. Hoppe. *New quadric metric for simplifying meshes with appearance attributes*. IEEE Visualization '99, pages 59-66. October 1999.

- [10] S. Geman and D. Geman. *Stochastic Relaxation, Gibbs distribution, and Bayesian restoration of images*. IEEE Transactions on Pattern Analysis and Machine Intelligence, November 1984
- [11] D. Scharstein and R. Szeliski. *A taxonomy and evaluation of dense two-frame stereo correspondence algorithms*. Int. J. Comp. Vision (2002) , pages 47(1-3), 7-42
- [12] R. Hartley and A. Zisserman. *Multiple View Geometry*. Cambridge University Press, 2000
- [13] Gelman, A. and Carlin, J. and Stern, H. and Rubin, D. *Bayesian Data Analysis*. Chapman and Hall, 1995
- [14] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald and W. Stuetzle. *Mesh Optimisation*. TR 93-01-01, Dept. of Computer Science and Engineering, University of Washington, January 1993
- [15] J. Isidoro, S. Sclaroff. *Stochastic Mesh-Based Multiview Reconstruction*. Presented in 3DPVT2002
- [16] L. Zhang, S. Seitz. *Image-Based Multiresolution Shape Recovery by Surface Deformation*. Proc. SPIE Vol. 4309, p. 51-61, Videometrics and Optical Methods for 3D Shape Measurement, Sabry F. El-Hakim, Armin Gruen, Editors

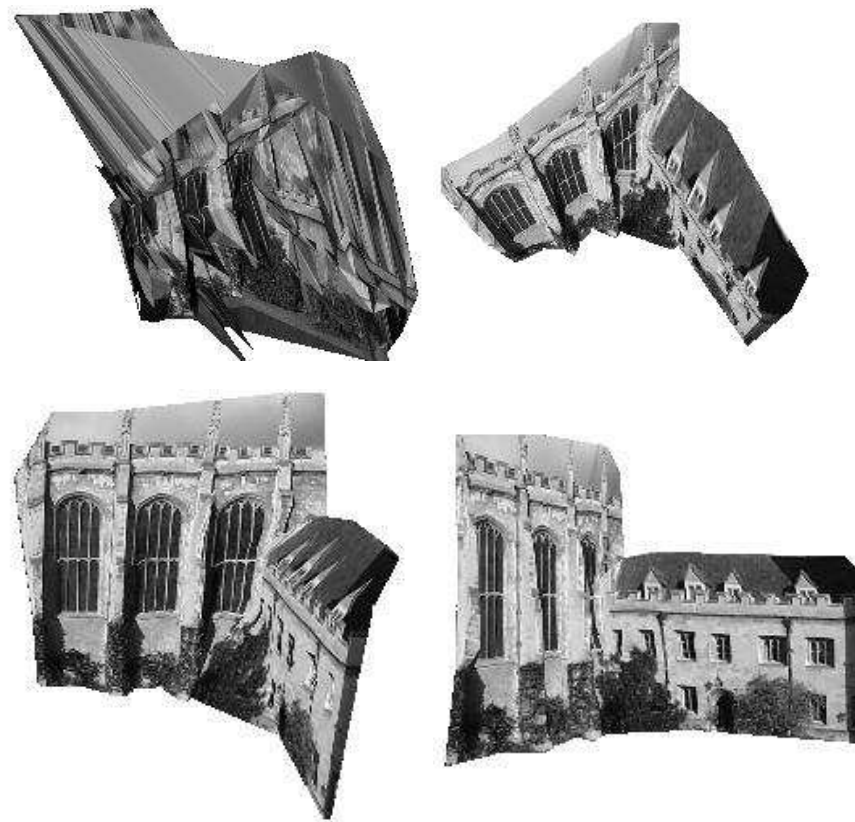


Figure 4: **Reconstruction.** Top left is the Delaunay triangulation of the original set of 300 vertices. Note the spikes along the line of sight caused by bad feature matches. The other three are views of the reconstructed model after convergence of our algorithm, that consists of 192 vertices.

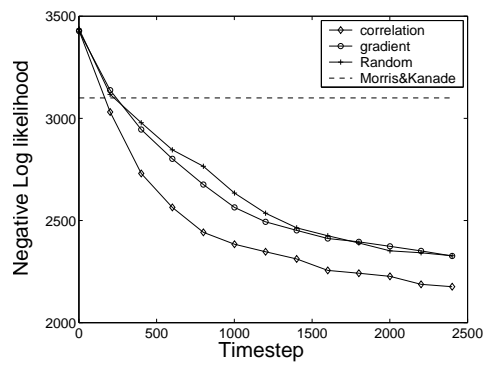


Figure 5: **Performance comparison.** The error (negative log of posterior) is shown for three different runs of the algorithm in which each of the three vertex moving jumps was used separately. The Correlation jump clearly outperforms the other two. The graph also shows the performance of Morris&Kanade [1] applied to the same scene. The reason it fails is that erroneous initial vertices cannot be removed and new areas of the surface cannot be added.