

# Mesh Optimisation Using Edge Information In Feature-Based Surface Reconstruction

Jun Liu and Roger Hubbard

Department of Computer Science, University of Manchester  
Manchester, M13 9PL, United Kingdom  
`{jun, roger}@cs.man.ac.uk`

**Abstract.** One of the most challenging and fundamental problems in computer vision is to reconstruct a surface model given a set of uncalibrated 2D images. Well-established Structure from Motion (SfM) algorithms often result in a sparse set of 3D surface points, but surface modelling based on sparse 3D points is not easy. In this paper, we present a new method to refine and optimise surface meshes using edge information in the 2D images. We design a meshing – edge point detection – re-meshing scheme that can gradually refine the surface mesh until it best fits the true physical surface of the object being modelled. Our method is tested on real images and satisfactory results are obtained.

## 1 Introduction

Much attention has been paid to the task of obtaining a surface representation from 3D data points on an unknown surface. Early work has been focused on the data acquired with a laser range scanner, with the characteristics that the obtained 3D point cloud is dense and well-distributed [1, 2, 3, 4]. Although these methods have been reported to be successful, the nature of the range scanning technique greatly limits its usefulness in real-world applications: it is an “invasive” technique in that the ray emitted by the scanner may damage the object being scanned; the scanning device is often very expensive; the scanning process is very slow even for a moderately sized object, and thus not suitable for modelling large scenes such as buildings.

The advances in computer vision technologies provide an exciting alternative for surface reconstruction. One of the most challenging and fundamental problems in computer vision is reconstruction of a surface model given a set of uncalibrated 2D images captured by a hand-held camera. Although this is not a solved problem, progress has been made in the last decade and a few working systems have been built. The first steps usually involve Structure from Motion (SfM) [5] and camera auto-calibration [6], delivering camera pose information as well as a sparse 3D point reconstruction based on image feature points. Pollefeys et al. [7] then applied dense stereo matching techniques on the images which results in a per-pixel density reconstruction of the scene. Lhuillier and Quan [8, 9] adopted a different approach by propagating points on the images to obtain a “quasi-dense” reconstruction.

Even with “dense” stereo matching, the 3D data from multiple images are sparse, noisy, and irregularly distributed compared to those from range scanners, therefore traditional 3D surface reconstruction techniques used with range data cannot be used with the data from passive computer vision systems. Fortunately, the extra 2D image information is still available which can be used to facilitate surface reconstruction. Lhuillier and Quan [8] proposed a variational approach integrating 3D stereo data with 2D image information; Solem and Heyden [10] addressed the problem based on methods for region tracking on surfaces and moving implicit curves; Paris et al. [11] used global graph cut optimisation to find optimal surface patches.

An interesting divergence from dense-stereo-data-based surface reconstruction is “feature-based surface reconstruction” [12, 13]: the sparse 3D points obtained with SfM techniques are directly fed into the surface reconstruction algorithm. This is based on the observation that the human vision system tends to recognise objects by salient features such as edges and points. Feature-based surface reconstruction has the advantage that it represents the scene models in a more efficient way: meshes are greatly simplified, thus the cost of computation and visualisation is greatly reduced. It also has the advantage over dense-stereo-based methods in that feature based surface reconstruction allows for much wider base lines in the input images, which often prevents the dense stereo matching algorithms from working. However, finding meshes that correspond to the true shape of scenes being modelled based on sparse 3D points is very difficult. This is probably the reason why feature-based surface reconstruction remains largely unpopular.

This paper addresses the mesh optimisation problem in feature-based surface reconstruction. The remainder of the paper is organised as follows: Section 2 discusses the limitations of previous work and briefly states the advantages of our method; Section 3 formally specifies the problem; Section 4 presents a robust method to detect points lying on the image edges; Section 5 presents a method to search for the correct 3D positions for the identified edge points; Section 6 shows some experimental results and Section 7 concludes our work with suggestions for future work.

## 2 Related Work

The problem of mesh optimisation in feature-based surface reconstruction was first addressed by Morris and Kanade [14]. In their pioneering work an initial mesh based on Delaunay triangulation is first obtained. An edge swapping technique is then applied to traverse possible topologies of the 3D feature points. They use a greedy algorithm to search for the best triangulation corresponding to the most consistent topology across multiple views. Vogiatzis et al. [15] extended this work by using simulated annealing instead of a greedy algorithm to search for the best triangulation, making the optimisation less susceptible to local minima. Most recently Nakatsuji et al. [16] detect texture discontinuities in the images and swap the edges to minimise the overall discontinuity of the triangulation.

In general, previous methods suffer from several problems:

1. They are heavily dependent on a judicious selection of feature points on edges and corners. If no feature points are found (or cannot be reliably tracked) on edges and corners, then edge swapping will not return a triangulation that reflects the true surface of the object.
2. Finding the optimal triangulation by edge swapping can easily get stuck in local minima. Statistical methods such as simulated annealing are computationally expensive, and their convergence cannot be guaranteed. Therefore, the previous methods can only work with a small set of 3D points.

**Our Contribution.** We present a new method for mesh optimisation in feature-based surface reconstruction by directly incorporating edge information in the triangulation process. Our method is different from previous work in that we do not rely on an edge swapping technique, whose complexity is exponential to the number of input 3D points. Instead, we adopt a meshing – edge point detection – re-meshing scheme. Our method has several advantages over the previous methods:

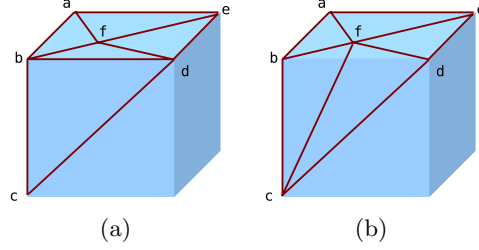
1. No edge swapping is involved. We only rely on the well-established 2D Delaunay triangulation algorithm. Our algorithm is readily extensible to large sets of 3D feature points.
2. Contrary to previous methods, triangle splitting is permitted (and is essential) in our method, allowing for a more general distribution of features on the object surface, i.e. features do not need to lie strictly on the corners.
3. Our method can be iterated multiple times to further refine the quality of the resultant surface mesh.

### 3 Problem Statement

Given a set of 3D points lying on the surface of an object, there exist many possible surface triangulations passing through all these points. A simple example is shown in Figure 1. Although both triangulations in Figure 1 are valid configurations in 2D as well as in 3D, only triangulation (a) is consistent with the true object surface. In practice we do not know the true surface but instead have a set of images of the object. The goal is to resolve this ambiguity by selecting a triangulation based on its consistency with this set of images of the object.

The problem can be mathematically formulated as follows: we have as input a set of  $n$  images  $\mathbf{I} = \{\mathbf{I}_1, \dots, \mathbf{I}_n\}$ , a set of  $m$  3D points  $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_m\}$  obtained with a SfM algorithm, and  $n$  projection matrices  $\mathbf{P} = \{\mathbf{P}_1, \dots, \mathbf{P}_n\}$  which define the transformation from 3D points to 2D points for each image. We define our triangular mesh model as  $\mathbf{M} = \{\mathbf{V}, \mathbf{E}\}$  where  $\mathbf{V}$  is a set of 3D points and  $\mathbf{E}$  is a set of edges connecting members of  $\mathbf{V}$ . Our goal is to find a mesh model  $\mathbf{M}$  that maximises the conditional likelihood  $\Pr$

$$\underset{\mathbf{M}}{\operatorname{argmax}} \Pr(\mathbf{M} \mid \mathbf{I}, \mathbf{X}, \mathbf{P}) \quad (1)$$



**Fig. 1.** Two of the many possible triangulation from 3D points on a cube. (a) Triangulation that corresponds to the true physical surface; (b) Triangulation that does not corresponds to the true physical surface: points from different planar surfaces are triangulated.

Note that  $\mathbf{V}$  and  $\mathbf{X}$  are different in our formulation. Unlike previous methods [14, 15, 16], where no new 3D points are added and only edges are swapped, we add new 3D points in  $\mathbf{X}$  to refine the mesh topology. Hence,  $\mathbf{X}$  is a subset of  $\mathbf{V}$  in our formulation.

## 4 Edge Point Detection

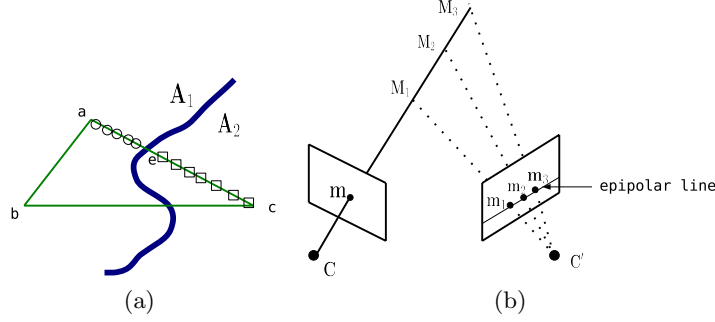
One of the methods to create an initial mesh model is to perform a Delaunay triangulation on the 2D feature points from one of the images and project it into 3D space [17, 7]. This method works well for dense stereo reconstruction but for feature-based surface reconstruction, it often triangulates points from different planar surfaces (as shown in Figure 1(b)), leading to artifacts when the object is viewed from different angles.

Since edges in the images are natural indicators of surface discontinuities, it is advisable to include edge information in the triangulation process. However, points on edges are difficult to track across the images and hence many feature detection algorithms such as SIFT [18, 19] deliberately discard points lying on edges. Furthermore, traditional edge detectors such as the Canny edge detector [20] and other gradient-based techniques, although successful in many application areas, tend to give false positive responses in the presence of highly textured objects. All these factors pose difficulties in applying the edge information in guiding the triangulation process.

Fortunately, in this particular problem, we are only interested in finding the *intersection* of images edge with edges in mesh triangles rather than the *integral* edges as a whole. This observation leads us to designing a specific “edge point detection” algorithm.

### 4.1 Problem Re-formulation

Consider that after Delaunay triangulation on the feature points on a reference image, as shown in Figure 2(a), the goal is to find a point  $e$  on the triangle edge



**Fig. 2.** (a) Naive Delaunay triangulation inevitably connects points from different areas (areas  $A_1$  and  $A_2$ ) together. The goal is to find the intersection point  $e$  lying on the discontinuity along the edge  $a \rightarrow c$ . (b) Multiple view geometry. The 3D point corresponding to image point  $m$  lies in the ray passing through  $m$  from the image centre  $C$ . When only one view is available, there is not enough information to determine the 3D location: any point lying on the ray is a possible candidate. However, when more views are available, the location can be identified by computing the reprojection on another image by camera  $C'$ : if  $M_k$  is the correct 3D point, then its reprojected 2D point  $m_k$  should have similar appearance with  $m$ .

$a \rightarrow c$  that lies on the discontinuity (image edge separating areas  $A_1$  and  $A_2$ ). If we travel along the edge  $a \rightarrow c$  and record the pixels in a vector  $\mathbf{S} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ , then we need to find a pixel  $\mathbf{p}_k$  where  $1 < k < n$  such that

$$\operatorname{argmax}_k \Pr(\mathbf{S}_1^{k-1} \mid \mathbf{A}_1) \Pr(\mathbf{S}_{k+1}^n \mid \mathbf{A}_2) \quad (2)$$

where  $\mathbf{S}_1^{k-1} = \{\mathbf{p}_1, \dots, \mathbf{p}_{k-1}\}$  and  $\mathbf{S}_{k+1}^n = \{\mathbf{p}_{k+1}, \dots, \mathbf{p}_n\}$ .

#### 4.2 Maximum-Likelihood Estimation and KL Divergence

In practice the distribution models of  $A_1$  and  $A_2$  are not known a priori unless some texture segmentation techniques are used. Therefore the formulation in Equation 2 is not readily applicable to our problem. However, we do know a priori that  $A_1$  and  $A_2$  are different from each other. If we can define an appropriate distance function  $D(\mathbf{S}_i, \mathbf{S}_j)$  to measure the (dis-)similarity of the two segments  $\mathbf{S}_i$  and  $\mathbf{S}_j$ , then we can formulate our problem in a maximum-likelihood estimation (MLE) framework

$$\operatorname{argmax}_k D(\mathbf{S}_1^{k-1}, \mathbf{S}_{k+1}^n) \quad (3)$$

In other words, we want to find a  $k$  such that  $\mathbf{S}_1^{k-1}$  and  $\mathbf{S}_{k+1}^n$  are most different from each other.

It is safe to assume that the pixel set sampled from area  $A$  follows a Gaussian distribution  $N$  with mean  $\mu$  and standard deviation  $\sigma$ . If the pixels are sampled from different areas, then they follow a mixture of Gaussian distribution  $\mathcal{N}$  with

mean  $\mu$  and covariance matrix  $\Sigma$ . Kullback-Leibler (KL) divergence can be used to measure the cross-entropy (i.e. dis-similarity) between two Gaussian mixtures  $f$  and  $g$ :

$$D_{\text{KL}}(f \parallel g) = \int f \ln \frac{f}{g} \quad (4)$$

where  $f \sim \mathcal{N}(\mu_f, \Sigma_f)$  and  $g \sim \mathcal{N}(\mu_g, \Sigma_g)$ .

Since there is no closed-form expression for KL-divergence between two mixture of Gaussians, computing this distance measure is usually done using Monte-Carlo simulation, which causes a significant increase in computational complexity. An approximation is proposed by Goldberger et al. [21] which leads to a closed-form solution:

$$D_{\text{KL}}(f \parallel g) = \frac{1}{2} \left( \ln \frac{|\Sigma_f|}{|\Sigma_g|} - d + \text{tr}(\Sigma_f^{-1} \Sigma_g) + (\mu_g - \mu_f)^\top \Sigma_f^{-1} (\mu_g - \mu_f) \right) \quad (5)$$

where  $d$  is the dimensionality of the Gaussian mixtures. In our problem  $d = 3$  because we model the RGB plane of each image pixel separately.

### 4.3 Algorithm Description

It is sensible to combine gradient-based edge detection with maximum-likelihood estimation. For each edge in the triangle, we travel through the pixels and find intensity discontinuities by computing the first-order derivative in the travelling direction. Suppose we record  $n$  pixels in one triangle edge  $\mathbf{S} = \{\mathbf{p}_1, \dots, \mathbf{p}_n\}$ , a good approximation to measure discontinuity  $d$  for pixel  $\mathbf{p}_k$  is

$$d(\mathbf{p}_k) = |\mathbf{p}_k - \mathbf{p}_{k-1}| + |\mathbf{p}_k - \mathbf{p}_{k+1}| \quad (6)$$

Gradient-based edge detection doesn't work well for highly textured areas. Imagine that in Figure 2(a),  $\mathbf{A}_1$  is highly textured and has repetitive pattern while  $\mathbf{A}_2$  is texture-less, then gradient-based edge detection will return many positive responses in  $\mathbf{A}_1$ , which is not desirable for finding true discontinuities. In this case, maximum-likelihood estimation can be used to discard false positive responses: the positive responses from gradient-based detection can be used as candidates in maximum-likelihood estimation framework, and the true discontinuity corresponds to the pixel  $\mathbf{p}_k$  that maximises the KL-divergence  $D_{\text{KL}}$  between  $\mathbf{S}_1^{k-1}$  and  $\mathbf{S}_{k+1}^n$ .

## 5 3D Position Identification

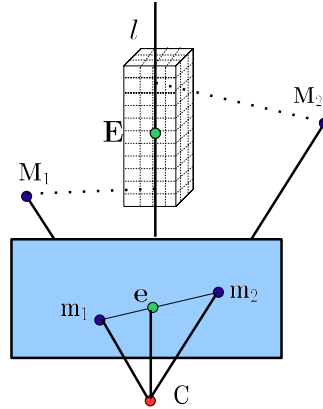
The detected edge points described in Section 4 do not provide any extra information to facilitate surface reconstruction unless their corresponding positions in 3D space are identified. Estimating 3D positions of feature points lying on edges in the SfM stage is not easy, as points along the edge usually have similar appearance, and hence they are difficult to identify and track across the images. However, camera poses and sparse reconstructed features are available after SfM.

This information can help constrain the search-space of the 3D positions of the detected 2D edge points.

Ideally, the corresponding 3D point of a 2D image point lies on the back-projected ray from the camera centre passing through the 2D image point (see Figure 2(b)). If we can identify the true location of the 3D point, then its reprojected point in each image should have similar appearances as they correspond to the same feature. Therefore, the 3D point identification problem can be viewed as an inverse problem of SfM.

In practice, however, inevitably there will be noise in SfM and camera auto-calibration processes. Therefore, the position of 3D point may show slight deviation from the ray. We propose to propagate the search-space such that it can take noise into account, as shown in Figure 3.  $\mathbf{m}_1$  and  $\mathbf{m}_2$  are two vertices of a triangle (obtained from Delaunay triangulation on the feature points) and  $\mathbf{M}_1$  and  $\mathbf{M}_2$  are the corresponding 3D points respectively (as after SfM their 3D locations are already known). An edge point  $\mathbf{e}$  is detected with the method described in Section 4. The search-space for its 3D location is a uniformly sampled 3D grid centred at  $\mathbf{E}$ .  $\mathbf{E}$  is the mid-point of projections of  $\mathbf{M}_1$  and  $\mathbf{M}_2$  on the viewing ray  $l$  which passes through  $\mathbf{e}$  from  $\mathbf{C}$ . The 3D grid is positioned in such a way that it has larger search-space along the ray  $l$  and smaller search-space in the direction perpendicular to  $l$ . Note that when the search-space perpendicular to  $l$  is zero, it reduces to search-space along the ray, ignoring the deviation caused by noise.

We evaluate each 3D point sample  $\mathbf{E}_k$  from the grid by computing its reprojected 2D point  $\mathbf{e}_i$  on each image  $\mathbf{I}_i$  and compare their similarity to the edge point  $\mathbf{e}$ . If  $\mathbf{E}_k$  is the correct 3D point, then its projection in other images should



**Fig. 3.** Search-space propagation.  $\mathbf{m}_1$  and  $\mathbf{m}_2$  are two vertices of a triangle and  $\mathbf{M}_1$  and  $\mathbf{M}_2$  are the corresponding 3D points respectively. An edge point  $\mathbf{e}$  is detected. The search-space for its 3D location is a uniformly sampled 3D grid centred at  $\mathbf{E}$ .  $\mathbf{E}$  is the mid-point of projections of  $\mathbf{M}_1$  and  $\mathbf{M}_2$  on the viewing ray  $l$  which passes through  $\mathbf{e}$  from  $\mathbf{C}$ . The 3D grid is positioned in such a way that it has larger search-space along the ray  $l$  and smaller search-space in the direction perpendicular to  $l$ .

have similar appearance to  $\mathbf{e}$ . The problem can be formulated as finding an  $\mathbf{E}_k$  that minimises the cost function

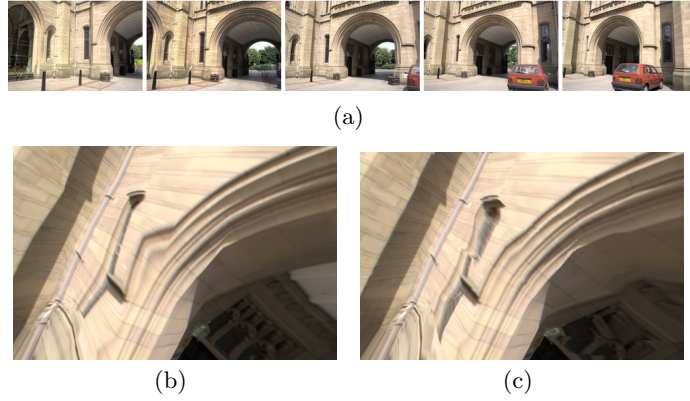
$$\min_{\mathbf{E}_k} \sum_i \mathcal{C}(\mathbf{e}, \mathbf{e}_i) \quad (7)$$

where  $\mathcal{C}(\mathbf{e}, \mathbf{e}_i)$  is the cost function measuring the dis-similarity between  $\mathbf{e}$  and  $\mathbf{e}_i$ . Two common criteria are Sum of the Squared Difference (SSD) and Normalised Cross-Correlation (NCC) between pixels in a small window centred at  $\mathbf{e}_i$  in each image. In our experiment we use NCC because it is less sensitive to illumination change between the views. We select the window size to be  $7 \times 7$ , as a trade-off between performance and speed.

One problem still remains. The visibility property  $\mathbf{V}_{\mathbf{E}_k}$  of the 3D point  $\mathbf{E}_k$  is not known beforehand: we have no knowledge about in which images the 3D point  $\mathbf{E}_k$  is seen. Fortunately, the neighbouring feature points can provide a reasonable approximation for  $\mathbf{V}_{\mathbf{E}_k}$ . As is shown in Figure 3, if  $\mathbf{e}$  is closer to  $\mathbf{m}_1$  than to  $\mathbf{m}_2$ , then we assign the visibility property  $\mathbf{V}_{\mathbf{M}_1}$  to  $\mathbf{E}_k$ ; otherwise we assign the visibility property  $\mathbf{V}_{\mathbf{M}_2}$  to  $\mathbf{E}_k$ .

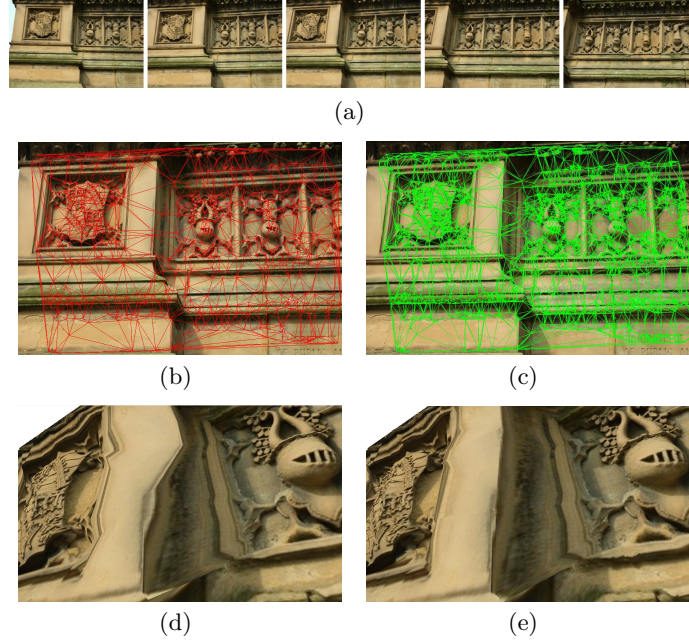
## 6 Experimental Results

We begin by performing a Delaunay triangulation on the feature points of a reference image. Edge points are detected as described in Section 4 and their 3D locations are identified as described in Section 5. The surface mesh can be optimised by re-meshing the new feature set on the image. Note that this process can be iterated multiple times until no triangle edge cuts through an image edge.



**Fig. 4.** Test case 1: Arch sequence. (a) 5 of the input images; (b) Close-up view of the original surface model from a very different angle from where the input images are captured. Notice that the artifacts in the area of arch are caused by poor meshing; (c) Enhanced surface model with our method. The curved surface in the arch is correctly modelled.





**Fig. 5.** Test case 2 : Monument sequence. (a) 5 of the input images; (b) Original mesh super-imposed onto the reference image: notice that triangles cut through image edges; (c) Refined mesh, triangulation is well-conditioned and very few triangles cut through image edges; (d) Original surface model viewed from very different angle from where the image is captured: the artifacts along the edge are caused by triangular meshes connecting points from different surfaces; (e) Refined surface model: edge points are detected and their 3D locations are correctly determined. The surface model refined by our method is very consistent with the true physical surface of the object.

3D feature points and camera information are obtained based on our previous work [13]. The reconstruction process is fully automatic and requires no information other than the images alone. Figure 4 shows the reconstructed surface model of an arch. Notice that the fine details of the curved surface are correctly reconstructed, which would be impossible if we use edge swapping techniques. Figure 5 shows another example demonstrating the result of our method. Our method is very efficient: both test cases involve detection and 3D position identification of around 10,000 edge points and it finishes within 10 seconds on a 2GHz processor.

## 7 Conclusion and Future Work

We presented a new method for mesh optimisation in feature-based surface reconstruction by directly incorporating edge information in the triangulation process. Our method is different from previous ones in that we do not rely on an

edge swapping technique. Instead, we adopt a meshing – edge point detection – re-meshing scheme. Experiments on real images show satisfactory results.

Our method still has limitations: it assumes that edges do not change across the images. Although it is true for most occasions, it may not hold when the object has a smoothly curved surface, in which case the edges change according to the viewpoint. Moreover, our method works less well when the edges lie on the epipolar line (See Figure 2(b)): re-projected 2D points are more difficult to distinguish and hence the identified 3D positions are less reliable. In our future work, we plan to solve the above problems by combining our method with the edge swapping method and use a more robust descriptor (such as a SIFT-like descriptor [18, 19]) rather than NCC to match reprojected 2D points.

## References

- [1] Hoppe, H., DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W.: Surface reconstruction from unorganized points. In: SIGGRAPH. (1992) 71–78
- [2] Szeliski, R., Tonnesen, D., Terzopoulos, D.: Modeling surfaces of arbitrary topology with dynamic particles. In: CVPR. (1993) 82–87
- [3] Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH. (1996) 303–312
- [4] Zhao, H., Osher, S., Merriman, B., Kang, M.: Implicit and nonparametric shape reconstruction from unorganized data using a variational level set method. *CVIU* **80**(3) (2000) 295–314
- [5] Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Second edn. Cambridge University Press (2003)
- [6] Triggs, B.: Autocalibration and the absolute quadric. In: CVPR. (1997) 609
- [7] Pollefeys, M., Gool, L.V., Vergauwen, M., Verbiest, F., Cornelis, K., Tops, J., Koch, R.: Visual modeling with a hand-held camera. *IJCV* **59**(3) (2004) 207–232
- [8] Lhuillier, M., Quan, L.: Surface reconstruction by integrating 3D and 2D data of multiple views. In: ICCV. Volume 02. (2003) 1313
- [9] Lhuillier, M., Quan, L.: A quasi-dense approach to surface reconstruction from uncalibrated images. *TPAMI* **27**(3) (2005) 418–433
- [10] Solem, J.E., Heyden, A.: Reconstructing open surfaces from unorganized data points. In: CVPR. Volume 02. (2004) 653–660
- [11] Paris, S., Sillion, F., Quan, L.: A surface reconstruction method using global graph cut optimization. *IJCV* **66**(2) (2006) 141–161
- [12] Taylor, C.J.: Surface reconstruction from feature based stereo. In: ICCV. Volume 01. (2003) 184
- [13] Liu, J., Hubbold, R.: Automatic camera calibration and scene reconstruction with scale-invariant features. In: ISVC. (2006) to appear
- [14] Morris, D., Kanade, T.: Image-consistent surface triangulation. In: CVPR. Volume 1. (2000) 332–338
- [15] Vogiatzis, G., Torr, P., Cipolla, R.: Bayesian stochastic mesh optimisation for 3D reconstruction. In: BMVC. Volume 2. (2003) 711–718
- [16] Nakatani, A., Sugaya, Y., Kanatani, K.: Mesh optimization using an inconsistency detection template. In: ICCV. (2005) 1148–1153

- [17] Pollefeys, M., Koch, R., Gool, L.V.: Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In: ICCV. (1998) 90
- [18] Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. (1999) 1150
- [19] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
- [20] Canny, J.: A computational approach to edge detection. TPAMI **8** (1986) 679–714
- [21] Goldberger, J., Gordon, S., Greenspan, H.: An efficient image similarity measure based on approximations of KL-divergence between two Gaussian mixtures. In: ICCV. Volume 1. (2003) 487