Automatic Camera Calibration and Scene Reconstruction with Scale-Invariant Features

Jun Liu and Roger Hubbold

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

Abstract. The goal of our research is to robustly reconstruct general 3D scenes from 2D images, with application to automatic model generation in computer graphics and virtual reality. In this paper we aim at producing relatively dense and well-distributed 3D points which can subsequently be used to reconstruct the scene structure. We present novel camera calibration and scene reconstruction using scale-invariant feature points. A generic high-dimensional vector matching scheme is proposed to enhance the efficiency and reduce the computational cost while finding feature correspondences. A framework for structure and motion is also presented that better exploits the advantages of scale-invariant features. In this approach we solve the "phantom points" problem and this greatly reduces the possibility of error propagation. The whole process requires no information other than the input images. The results illustrate that our system is capable of producing accurate scene structure and realistic 3D models within a few minutes.

1 Introduction

The possibility of being able to acquire 3D information from 2D images has attracted considerable attention in recent years. It offers promising applications in such areas as archaeological conservation, scene-of-crime analysis, architectural design, movie post-processing, to name but a few. The idea of automatic reconstruction from images is intriguing because, unlike other techniques which usually require special devices to obtain the data (e.g. laser scanner, ultrasound), the digital images are readily available. Although reconstructing 3D models from 2D images is a very difficult problem, recent years have seen several theoretical breakthroughs, and a few systems have already been built. However, most of the systems only work under restrictive conditions, and are not readily applicable to more general cases.

One of the most important stages of scene reconstruction is structure from motion (SfM), which determines the camera poses and scene structure based on image information alone. Feature points are first extracted from each input image, and they are tracked to provide information about the relations between the images. Therefore, feature extraction and tracking act as a starting point for scene reconstruction, and their performance largely determines the overall reliability of the reconstruction algorithm. Two of the most popular feature tracking

G. Bebis et al. (Eds.): ISCV 2006, LNCS 4291, pp. 558–568, 2006.

algorithms are the Harris corner detector [1] followed by Sum of Squared Difference (SSD) matching [2, 3, 4, 5], and the Kanade-Lucas-Tomasi (KLT) tracker [6, 7]. These algorithms work well when the baseline (i.e. viewpoint change between images) is relatively small and the appearance of the features doesn't change much across subsequences. However, this condition does not hold when the input data is a sequence of "sparse" images instead of a "dense" video stream, or where the appearances of the features change significantly with respect to the viewpoint. Therefore, a more robust feature tracking method is desirable to form a good foundation for the scene reconstruction problem.

The Scale Invariant Feature Transformation (SIFT), first proposed by Lowe[8, 9], extracts distinctive features which act as descriptors of local image patches. These features are largely invariant to image scale and rotation, and partially invariant (i.e. robust) to affine distortion, change in 3D viewpoint, addition of noise and change in illumination. SIFT has become well-accepted by the computer vision community. A recent evaluation by Mikolajczyk and Schmid [10] suggested that the SIFT-based descriptors performed the best among many other local descriptors, in terms of distinctiveness and robustness to various changes in viewing conditions. Successful applications of the SIFT algorithm have been reported in the areas of object recognition [8, 9], panorama stitching [11] and augmented reality [12].

Due to the invariant properties of SIFT, it can potentially tackle the problem of wide baseline matching and matching between significantly changing features. There is, however, very little literature about the application of SIFT in such areas as camera calibration and scene reconstruction. A similar but different work to ours is by Gordon and Lowe [12], where SIFT features are extracted and matched to relate any two images from an image sequence. In this paper we propose a more complete algorithm for SIFT feature matching and SfM computation. Our system is different from others in that we not only use SIFT features for camera pose estimation, but also their reconstructed 3D positions for scene analysis.

The rest of the paper is organised as follows: Section 2 introduces a new feature matching algorithm based on SIFT, which improves the efficiency without compromising its accuracy. Section 3 discusses a novel framework for SfM, with the advantage that it can match features from non-adjacent images, thus solving the problem of "phantom points" and making the system less prone to error propagation. Section 4 shows some experimental results to validate our method and Section 5 concludes our work.

2 A New Approach for SIFT Feature Matching

2.1 Related Work

The SIFT algorithm computes, for each keypoint, its location in the image as well as a distinctive 128-dimension descriptor vector associated with it. Matching a keypoint to a database of keypoints is usually done by identifying its nearest neighbour in that database . The nearest neighbour is defined as the keypoint

with minimum Euclidean distance to the descriptor vector. To reduce the number of spurious matches, the ratio R of the distance of the closest neighbour D to that of the second closest neighbour D′ is computed. The matches with a ratio greater than a certain threshold (0.8 is suggested by Lowe) are discarded.

Due to the high dimensionality of the keypoints, the matching process is relatively expensive. The naive exhaustive search has a complexity of $\mathcal{O}(nmd)$ where n is the number of keypoints in the database, m is the number of keypoints to be matched, and d is the dimension of the descriptor vector. The best algorithms, such as a k-d tree, provide no speedup over exhaustive search for more than about 10 dimensional spaces [9]. Therefore, two approximate matching algorithms have been proposed, namely Best-Bin-First(BBF) [13] and PCA-SIFT [14]. The BBF algorithm is very similar to the k-d tree algorithm, except that the BBF algorithm restricts the search step so that it sets an absolute time limit on the search. As a result, the BBF algorithm returns a nearest neighbour at a high probability. However, our experiment shows that as the number of keypoints and the dimension increases, the BBF algorithm provides no significant speedup over the standard matching method. PCA-SIFT, on the other hand, reduces the dimensionality based on Principal Component Analysis. Both algorithms incur a certain amount of loss of correct matches. In our system SIFT is applied to act as a starting point for structure & motion and camera calibration, so it is desirable that the data is as noise-free as possible.

2.2 Problem Specification

A formal specification of the matching problem is first outlined. Suppose we have in the database n points $\mathbf{P} = \{\mathrm{P}_0, \mathrm{P}_1, ..., \mathrm{P}_{n-1}\}$, each of which comprises of a d dimensional descriptor vector $[V_0, V_1, ..., V_{d-1}]$. We want to match m points $\mathbf{P'} = \{\mathrm{P}'_0, \mathrm{P}'_1, ...\mathrm{P}'_{m-1}\}$, each with a descriptor vector of the same dimension $[V'_0, V'_1, ..., V'_{d-1}]$, to the database, in order to obtain a set of matched pairs $\mathbf{S} = \{(\mathrm{P}, \mathrm{P}') \mid \mathrm{P} \leftrightarrow \mathrm{P}', \mathrm{P} \in \mathbf{P}, \mathrm{P}' \in \mathbf{P'}\}$. A matched pair $(\mathrm{P}_i, \mathrm{P}'_j)$ has the property that P_i is the nearest neighbour of P'_j in the database $\mathbf{P}$, i.e. ,

$$\forall \mathrm{P}_k \in \mathbf{P} \qquad \mathrm{D}(\mathrm{P}_i, \mathrm{P}'_j) \leq \mathrm{D}(\mathrm{P}_k, \mathrm{P}'_j) \tag{1}$$

where $\mathrm{D}(\mathrm{P}_i, \mathrm{P}'_j)$ is the Euclidean distance between the two descriptor vectors associated with the two keypoints. Furthermore, if P_k is the second nearest neighbour to P'_j in the database, then another constraint should be satisfied that

$$\mathrm{D}(\mathrm{P}_i, \mathrm{P}'_j) \ / \ \mathrm{D}(\mathrm{P}_k, \mathrm{P}'_j) \leq thr \tag{2}$$

where thr is the threshold value, normally 0.8. This thresholding is designed to make sure that the match is distinctive enough from other possible matches, so as to discard many spurious matches.

2.3 The Algorithm

Here we present a new method which improves the efficiency without compromising its accuracy. Our algorithm first performs a Principal Component Analysis

(PCA) on the two sets of keypoints $\mathbf{P}$ and $\mathbf{P'}$, or more specifically, on the descriptor vectors associated with them. PCA is essentially a multivariate procedure which rotates the data such that maximum variabilities are projected onto the axes. In our case, the descriptor vector sets

$$\mathbf{V} = \{(V_0^i, V_1^i, ...V_{d-1}^i) \mid \mathrm{P}_i \in \mathbf{P}\} \tag{3}$$

$$\mathbf{V'} = \{(V'^j_0, V'^j_1, ...V'^j_{d-1}) \mid \mathrm{P}'_j \in \mathbf{P'}\} \tag{4}$$

are transformed into

$$\widehat{\mathbf{V}} = \{(\widehat{V}_0^i, \widehat{V}_1^i, ...\widehat{V}_{d-1}^i) \mid \mathrm{P}_i \in \mathbf{P}\} \tag{5}$$

$$\widehat{\mathbf{V'}} = \{(\widehat{V'}^j_0, \widehat{V'}^j_1, ...\widehat{V'}^j_{d-1}) \mid \mathrm{P}'_j \in \mathbf{P'}\} \tag{6}$$

with $\widehat{V}_0$ and $\widehat{V'}_0$ representing the dimension of the greatest amount of variation, $\widehat{V}_1$ and $\widehat{V'}_1$ representing the dimension of the second greatest amount of variation, and so on.

The next step is that for every keypoint P'_j in $\mathbf{P'}$, two initial full Euclidean distances between P'_j and the first two elements in $\mathbf{P}$, P_0 and P_1, are computed. These initial distances, $\mathrm{D}(\mathrm{P}'_j, \mathrm{P}_0)$ and $\mathrm{D}(\mathrm{P}'_j, \mathrm{P}_1)$, are compared, with the smaller one assigned to the nearest distance Nd, and the bigger one assigned to second nearest distance Snd.

After the initialisation, the comparison continues, but without the necessity to compute the full Euclidean distance for each keypoint in $\mathbf{P}$. Suppose now we want to test the keypoint P_i and see whether it is a nearer neighbour to P'_j than the current nearest one. We start by computing the difference of the vector in the first dimension $\mathrm{D}^2 \leftarrow (\widehat{V'}^j_0 - \widehat{V}^i_0)^2$, and compare it with the nearest distance squared Nd^2. If $\mathrm{D}^2 \geq Nd^2$, which indicates that P_i cannot become the nearer neighbour, then it is unnecessary to compute the rest of the dimensions. If $\mathrm{D}^2 < Nd^2$, then the process continues by adding the difference of the vector in the second dimension, $\mathrm{D}^2 \leftarrow \mathrm{D}^2 + (\widehat{V'}^j_1 - \widehat{V}^i_1)^2$. The aim of this method is to avoid unnecessary computations in the dimensional space, i.e., to more quickly discard any keypoint which is unlikely to be the nearest neighbour. If, after going though all the dimensions d, $\mathrm{D}^2 < Nd^2$ still holds, then we identify P_i as the new nearest neighbour by assigning Nd to Snd, and assigning D to Nd.

The process continues until it reaches the end of the list P_{n-1}. The final stage is to compute the ratio R of the distance of the nearest neighbour Nd to that of the second nearest neighbour Snd: $R = Nd/Snd$. If R is below a certain threshold thr, then the matched pair is added to the set $\mathbf{S}$, otherwise there is no reliable match. The role PCA plays here is to re-arrange the order of the dimensions so that the ones with larger variation come before the ones with smaller variation. This allows this algorithm to execute more quickly, i.e. to discard faster the keypoint which is not the nearest neighbour.

2.4 Experimental Results

We use an experimental setup where we match 2 sets of keypoints of same size. This assumption is valid when the SIFT algorithm is applied to camera calibration, in which case the number of keypoints detected for each frame does not vary very much. The number of keypoints is in a range from 250 to 4000. We use Pollefeys' Castle sequence [15] to illustrate the algorithm. Two images are randomly selected from the Castle Sequence, from which the SIFT algorithm can detect up to around 4200 keypoints for each image. Then a random subset of the detected keypoints is selected and matched using the standard exhaustive search algorithm as well as our new algorithm. In this experiment 8 different image pairs are tested, and each pair is matched 3 times with a certain number of keypoints selected. The average time spent on keypoint matching is compared and the result is shown in Figure 1. The result suggests that for the cases where

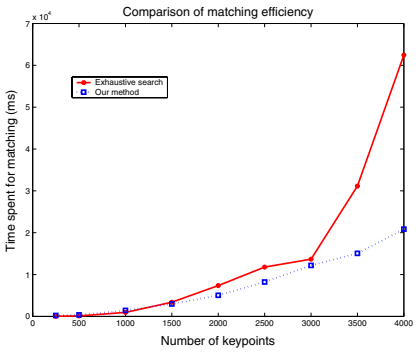


Fig. 1. Efficiency comparison between the standard exhaustive search algorithm and our improved algorithm

the number of keypoints is less than 1000, the performance of our algorithm is only slightly worse than the standard algorithm. This is because the PCA computation in our algorithm introduces an overhead, which offsets the speedup for modest point sets. However, our algorithm significantly outperforms the original one when the number of keypoints exceeds 3000. Therefore, our algorithm is ideal for a large keypoint database, as it greatly improves the efficiency while preserving the accuracy.

3 A Novel Framework for Structure from Motion

3.1 The "Phantom Points" Problem

The classical SfM only relates an image to the previous one. It is implicitly assumed that once a point is out of frame or occluded, it will not reappear. Although this is true for many sequences, the assumption does not always hold. Imagine a certain point becomes occluded for several frames in the middle of the sequence, but

becomes visible again for the rest of the sequence. The classical SfM method will generate two different 3D points although they are supposed to be the same 3D point. Here we coin the term *phantom points*, referring to points which the algorithm generates, but which do not really exist separately. The "phantom points" problem has so far not been properly addressed in the computer vision literature. Unfortunately, the problem often occurs in real image sequences, where there are foreground occlusions, or the camera moves back and forth.

3.2 A New Framework for SfM

We start by extracting the keypoints from the first image and inserting them into a list of vectors. Each keypoint P has three properties associated with it: its location in the image, `coord`, its feature vector, `fvec`, and the frame number of the image which it is from, `fnum`. After inserting the keypoints of the first image, the list is illustrated in Figure 2(a) (with `fnum` marked):

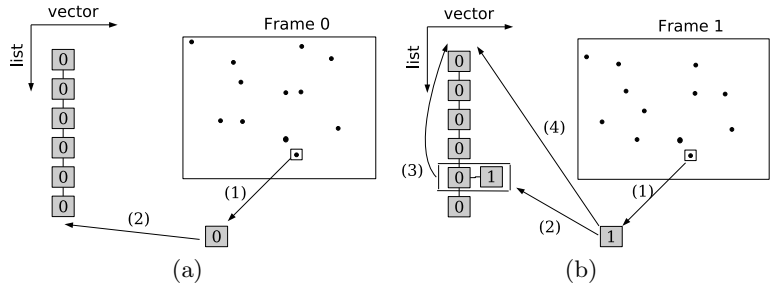


Fig. 2. (a): Adding the first frame: (1) Features are extracted with SIFT, each of which contains the information of its location in the image `coord`, its feature vector `fvec`, and the frame number of the image which it is from `fnum`. Here only `fnum` is illustrated; (2) The keypoints are inserted at the back of the list. (b): Adding the second frame: (1) Features are extracted, which are matched to the list; (2) For those which find a match, we extend the vector and move the pair (3) to the front of the list; (4) For those which cannot find a match, we insert them at the front of the list.

The keypoints from the second image are extracted and matched with the method described in Section 2.3. For those which do not match any keypoints in Frame 0, we simply insert them at the front of the list. For those which do match, we extend the vector and move the pair to the front of the list, which is illustrated in Figure 2(b).

From the matched pairs a fundamental matrix `F` is computed. Based on `F`, spurious matches (the ones which do not adhere to the epipolar constraint) are detected. The false matches are, however, not removed from the list as in traditional methods. Instead, false matches are split: we remove the last item from the vector and insert it at the front of the list (See Figure 3(a)). This way the keypoints that SIFT detects are utilised to the maximum: the falsely matched keypoints are given another chance to match the keypoints from later frames.

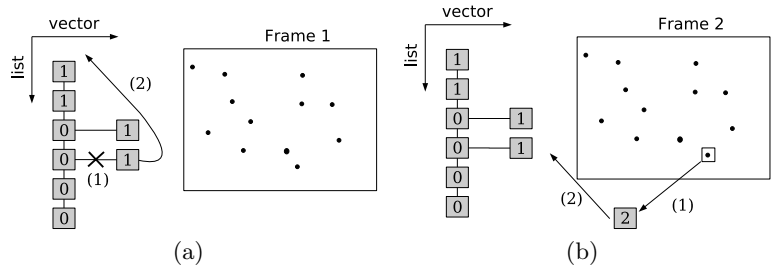


Fig. 3. (a): Rejecting outliers: outliers are detected based on the computed fundamental matrix F. If a pair is detected as an outlier, then the algorithm (1) removes the unmatched features and (2) insert it at the front of the list. (b) Adding the third frame: (1) Features are extracted, and (2) matched to the last item of each vector. Note that the keypoints from Frame 2 can be matched to both those from Frame 1 and those from Frame 0.

The initial poses and structure are computed the same way as the traditional method. When a new view is available, the extracted keypoints are compared to the last item of each vector in the list. Again the outliers are "discarded" by splitting the matches rather than removing them. Figure 3(b) shows, as an example, the status of the list after adding Frame 2. Note that the keypoints from Frame 2 can be matched to both those from Frame 1 and those from Frame 0. This is important because the matching is no longer restricted to adjacent frames. The framework described here natively solves the "phantom points" problem.

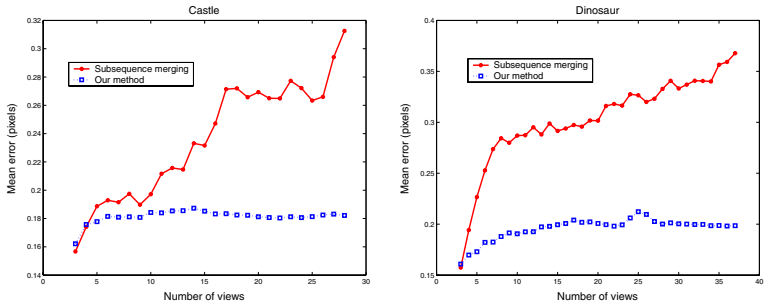
4 Experimental Results

Our SfM framework has been tested with the dinosaur sequence [16] (see Figure 5(b)) from the Robotics Group, University of Oxford. Our work is different from theirs [16] in that we do not require any prior knowledge of the input sequence, i.e. we do not need to know whether it is a turntable sequence or not. To provide a comparison, we first reconstruct the sequence with the traditional

Fig. 4. Comparison of the reconstruction with the traditional method and our method (view from the top). (a): With the traditional method, severe "phantom points" lead to misalignment of the tail. (b): There are no "phantom points" with our method; thus the shape of the tail is consistent.

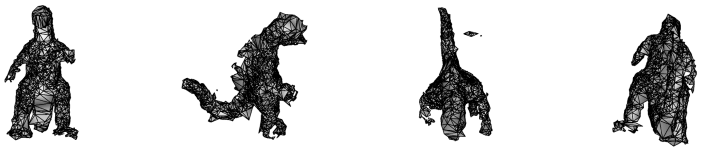
(a) (b)

Fig. 5. Image sequences used in the comparison of the reprojection error. (a) Castle sequence, 3 samples of 28 images; (b) Dinosaur sequence, 3 samples of 37 images.



(a) (b)

Fig. 6. Comparison of mean reprojection error between subsequence merging and our method: (a) Castle sequence and (b) Dinosaur sequence



(a) (b) (c) (d)

Fig. 7. Meshed model of the Dinosaur sequence: (a)front view, (b)side view, (c)top view and (d)back view. The model comprises of more than 6000 robustly tracked and well-distributed 3D points. With our system the whole reconstruction process (from the initial input images to the final output model) requires less than 10 minutes.

method, where the features from current frame only relate to the previous adjacent frame. To better illustrate the reconstruction of feature points, we generate a novel view from the top of the dinosaur. From Figure 4(a) it is obvious that this method suffers from the "phantom points" problem: the tail of the dinosaur exhibits slight misalignment, although the dinosaur is supposed to have only one integral tail. Note that the misalignment effect is exaggerated by error propagation in camera auto-calibration. The sequence is then tested with our new SfM framework, where features from current frame are matched to those from all the previous frames, and the result is shown in Figure 4(b).

Quantitative assessment was carried out to validate the advantages of our proposed SfM framework. Two publicly available image sequences were chosen:

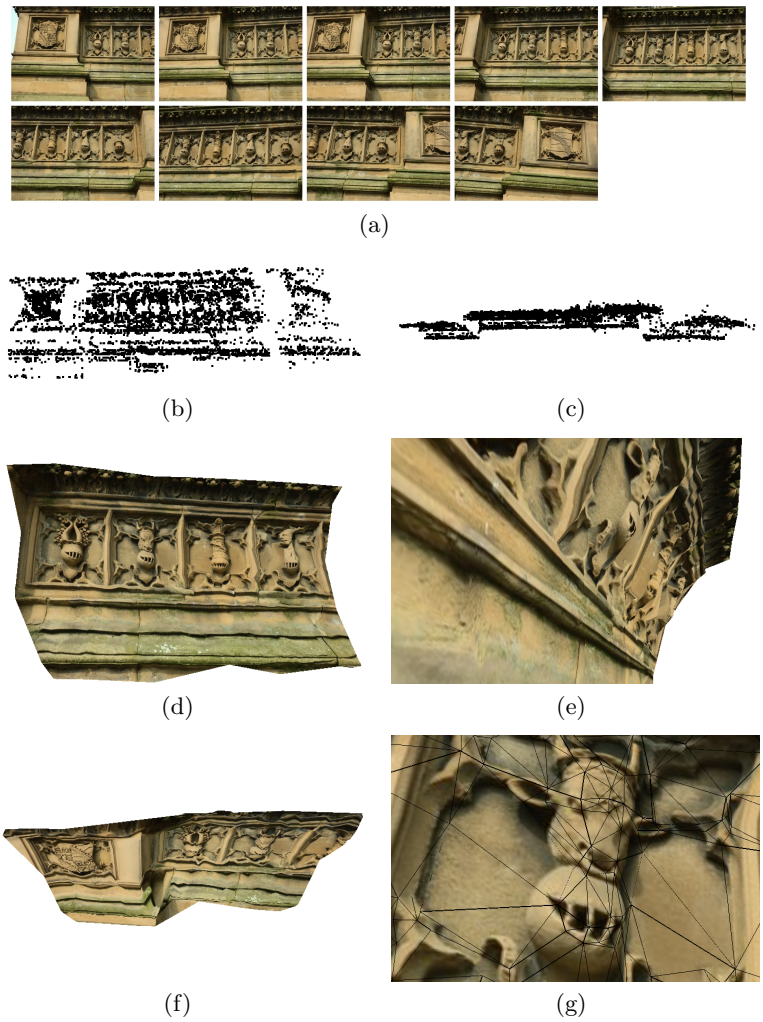


Fig. 8. (a) A challenging test case consisting of 9 images, each of which is 1024 × 679 in resolution. This small set of images involve complicated camera movements including scaling and wide baseline translation. The surface point reconstruction viewed (b) from the front and (c) from the top illustrates that our system performs well in linking the widely separated frames into a consistent scene structure. In (d) a reference image is selected and a textured model is viewed from the front. We move the viewpoint to somewhere very different from the original ones and a novel view is shown in (e) from the very left and (f) from the top (textured with a different reference image). The straightness of lines demonstrates the accurate recovery of the depth information. We further analyse the quality of reconstruction by super-imposing the triangular meshes onto the textured model. The zoomed-in details are shown in (g). Meshes are more refined in complicated areas than in plain areas. This is desirable because the computational resources are better distributed, biasing towards fine recognisable details in both scene reconstruction and model rendering. The reconstruction finishes within 5 minutes on a 2GHz processor.

the Castle sequence [15] (see Figure 5(a)) and the Dinosaur sequence [16] (see Figure 5(b)). A commonly used criterion to analyse the quality of reconstruction is the "reprojection error", which is the geometric Euclidean distance (or L_2 norm) between the projection of the reconstructed 3D point and the measured image point. In our experiments the mean reprojection error for all the reconstructed 3D points is used as an indication for the quality of the SfM methods. Our results are compared to the results using subsequence merging [17, 18].

Even though the subsequence merging technique performs well in constraining the overall mean reprojection error, it still shows moderate signs of error propagation. Results in Figures 6(a) and 6(b) suggest that our method is significantly less prone to error propagation compared to the subsequence merging technique. It is also interesting to see that our method performs surprisingly well for the Dinosaur sequence, considering the fact that it is a turntable sequence involving frequent self-occlusions. The ability to relate non-adjacent frames is important for pose estimation, as it results in projection matrices in a more consistent projective framework.

Figure 7 shows the reconstructed model of the Dinosaur sequence. Our system recovers 6105 surface points which are subsequently meshed using Cocone[19]. Several views are taken from positions very different from the original viewpoints and the results indicate that the model structure is accurately reconstructed. The whole reconstruction process requires no user intervention and finishes within 10 minutes on a 2GHz processor. Our system was further tested with photos taken with a Nikon D70s digital camera. Figure 8 shows our reconstruction of a sculptured memorial.

5 Conclusions and Future Work

We have presented a scene reconstruction system based on scale-invariant feature points. Our system is carefully designed such that the features from non-adjacent frames can be matched efficiently. We solve the "phantom points" problem and greatly reduce the chance of error propagation. Experimental results show that relatively dense and well-distributed surface points can be recovered. Our system assigns refined and detailed meshes to complicated areas, but coarse and simple meshes to plain areas. This is desirable because the computational resources are better distributed, biasing towards fine recognisable details in both scene reconstruction and model rendering.

Future work includes a more sophisticated meshing scheme and inclusion of edge information to better represent the scene structure.

References

[1] Harris, C.J., Stephens, M.: A combined corner and edge detector. In: Proceedings of 4th Alvey Vision Conference. (1988) 147–151

[2] Zhang, Z., Deriche, R., Faugeras, O., Luong, Q.T.: A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. AI **78**(1-2) (1995) 87–119

[3] Fitzgibbon, A.W., Zisserman, A.: Automatic 3D model acquisition and generation of new images from video sequences. In: EUSIPCO. (1998) 1261–1269
[4] Pollefeys, M., Gool, L.V., Vergauwen, M., Cornelis, K., Verbiest, F., Tops, J.: Image-based 3D acquisition of archaeological heritage and applications. In: Proceedings of the 2001 conference on Virtual reality, archeology, and cultural heritage. (2001) 255–262
[5] Gibson, S., Hubbold, R.J., Cook, J., Howard, T.L.J.: Interactive reconstruction of virtual environments from video sequences. Computers and Graphics **27** (2003) 293–301
[6] Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision (DARPA). In: Proceedings of the 1981 DARPA Image Understanding Workshop. (1981) 121–130
[7] Shi, J., Tomasi, C.: Good features to track. In: CVPR. (1994) 593–600
[8] Lowe, D.G.: Object recognition from local scale-invariant features. In: ICCV. (1999) 1150
[9] Lowe, D.G.: Distinctive image features from scale-invariant keypoints. IJCV **60**(2) (2004) 91–110
[10] Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. TPAMI **02** (2005) 257
[11] Brown, M., Lowe, D.G.: Recognising panoramas. In: ICCV. (2003) 1218–1225
[12] Gordan, I., Lowe, D.G.: Scene modelling, recognition and tracking with invariant image features. In: ISMAR. (2004) 110–119
[13] Beis, J.S., Lowe, D.G.: Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: CVPR. (1997) 1000
[14] Ke, Y., Sukthankar, R.: PCA-SIFT: A more distinctive representation for local image descriptors. In: CVPR. (2004) 506–513
[15] 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
[16] Fitzgibbon, A.W., Cross, G., Zisserman, A.: Automatic 3D model construction for turn-table sequences. In Koch, R., Gool, L.V., eds.: Proceedings of SMILE Workshop on Structure from Multiple Images in Large Scale Environments. Volume 1506. (1998) 154–170
[17] Gibson, S., Cook, J., Howard, T.L.J., Hubbold, R.J., Oram, D.: Accurate camera calibration for off-line, video-based augmented reality. In: ISMAR. (2002) 37
[18] Repko, J., Pollefeys, M.: 3D models from extended uncalibrated video sequence: addressing key-frame selection and projective drift. In: Fifth International Conference on 3-D Digital Imaging and Modeling. (2005) 150–157
[19] Dey, T.K., Goswami, S.: Provable surface reconstruction from noisy samples. In: Proceedings of 20th ACM-SIAM Symposium on Computational Geometry. (2004) 330–339