Divide and Conquer: Efficient large-scale structure from motion using graph partitioning

Brojeshwar Bhowmick[1], Suvam Patra[1], Avishek Chatterjee[2], Venu Madhav Govindu[2], Subhashis Banerjee[1]

[1] Indian Institute of Technology Delhi, New Delhi, India
{brojeshwar, suvam, suban}@cse.iitd.ac.in
[2] Indian Institute of Science, Bengaluru, India
{avishek, venu}@ee.iisc.ernet.in

Abstract. Despite significant advances in recent years, structure-from-motion (SfM) pipelines suffer from two important drawbacks. Apart from requiring significant computational power to solve the large-scale computations involved, such pipelines sometimes fail to correctly reconstruct when the accumulated error in incremental reconstruction is large or when the number of 3D to 2D correspondences are insufficient. In this paper we present a novel approach to mitigate the above-mentioned drawbacks. Using an image match graph based on matching features we partition the image data set into smaller sets or components which are reconstructed independently. Following such reconstructions we utilise the available epipolar relationships that connect images across components to correctly align the individual reconstructions in a global frame of reference. This results in both a significant speed up of at least one order of magnitude and also mitigates the problems of reconstruction failures with a marginal loss in accuracy. The effectiveness of our approach is demonstrated on some large-scale real world data sets.

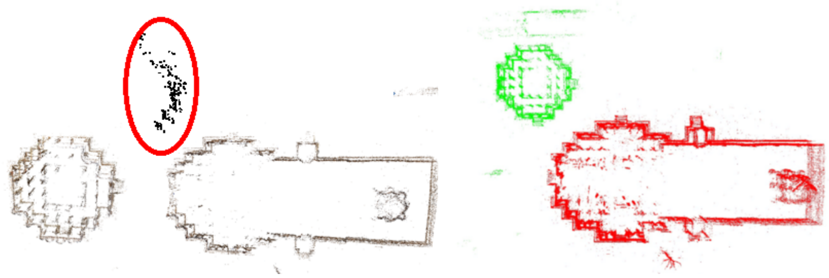
1 Introduction

In structure from motion (SfM) we typically use many images of a scene to solve for both the 3D scene being viewed and the parameters of the cameras involved. Most contemporary large-scale SfM methods [1–5] use the bundle adjustment method [6] which simultaneously optimises for both structure and camera parameters using point correspondences in images by minimising a global cost function. However, being a joint optimisation over all cameras and 3D points, bundle adjustment often fails for large data sets. This is typically due to an accumulation of error in an incremental reconstruction or when cameras are weakly connected to 3D feature points. In addition, owing to the very large number of variables involved, bundle adjustment is also very computationally demanding and time consuming. In this paper we adopt a divide-and-conquer strategy that is designed to mitigate these problems. In essence, our approach partitions the full image data set into smaller sets that can each be independently reconstructed using a standard approach to bundle adjustment. Subsequently, by utilising available geometric relationships between cameras across the individual partitions, we solve a global registration problem that correctly and accurately places each individual 3D reconstructed component into a single global frame of reference.

In what follows we show that this approach is not only more robust with respect to failures in reconstruction but also gives significant improvements over the state-of-the-art techniques in terms of computational speed. The main contributions of our paper are:

1. A principled method based on normalised cuts [7] to partition the match graph of a large collection of images into disjoint connected components which can be independently and reliably reconstructed. This process also automatically identifies a set of connecting images between the components which can be used to register the independent reconstructions. Specifically, these are the image pairs specified by the cut edges in the graph.
2. A method for registering the point clouds corresponding to the independent connected components using pairwise epipolar geometry relationships. The epipolar based registration technique proposed in this paper is more robust than the standard techniques for registering point clouds using 3D-3D or 3D-2D correspondences. Registration methods based on 3D point correspondences do not use all available information (image correspondences) and may fail when the point clouds do not have sufficient number of 3D points in common. 3D-2D based methods, such as a sequential bundler [1, 2, 8], often result in broken reconstructions when the number of points available are inadequate for re-sectioning or when common 3D points are removed at the outlier rejection stage [1] (see Table 4). The proposed registration algorithm using pairwise epipolar geometry alleviates this problem as is shown in Figure 1 and discussed in Section 4. Considered as an independent approach, the epipolar based algorithm can also be used to register independently reconstructed point clouds by introducing a few connecting images.

Matching all pairs of images in an iterative bundler is computationally expensive, especially when the number of images in the collection is large. There have been several attempts to reduce the number of images to be matched. Frahm *et al.* [9, 10] try to find some representative "iconic images" from the image data set and then partition the iconic scene graph, reconstruct each cluster and register them using 3D similarity transformations. Snavely *et al.* [11, 12] and Havlena *et al.* [13] compute skeletal sets from the match graph to reduce image matching. All these methods reduce the set of images on which they run SfM. Moreover, incremental bundle adjustment is also known to suffer from drift due to accumulation of errors which increase as the number of images increase [5, 14, 1]. Crandall *et al.* [5, 14] propose an MRF based discrete formulation coupled with continuous Levenberg-Marquadt refinement for large-scale SfM to mitigate this problem. To reduce the matching time, Wu [1] (henceforth VSFM) proposed preemptive matching to reduce the number of pairs to be matched. Moreover, all cameras and 3D points are optimised only after a certain number of new cameras



(a) Reconstruction failure by VSFM [1]. (b) Successful reconstruction by our method.

Fig. 1: Plan view of reconstruction of two temples at the Hampi site in India : (a) illustrates the failure of VSFM [1] due to inadequate points during re-sectioning (marked in red) whereas (b) our approach correctly solves the reconstruction problem. Please view this figure in color.

are incorporated into the iterative bundler. Although VSFM demonstrates approximately linear running time, sometimes it fails for large data sets when the accumulated errors of iterative bundler become large [1]. Although there have been some recent global methods [15, 16], to be able to solve large-scale SfM problems, global methods need to be exceedingly robust. Farenzena *et al.* [17] also propose to merge smaller reconstructions in a bottom up dendrogram. However, their largest datasets are of only 380 images and their use of reprojection errors of common 3D points for merging is unsuitable for very large datasets. In our approach, we propose to decompose the image set into smaller components so that the match graph of each component is densely connected. This is likely to yield correct 3D reconstructions, since fewer problems are encountered during the re-sectioning stage of a standard iterative bundler and the reconstruction is robust. Restricting pairwise image matching to within each component also yields a significant reduction in computation time. Moreover SfM based reconstruction of each component can be carried out in parallel. Our approach is conceptually depicted in Figure 2.

The rest of the paper is organised as follows. Section 2 discusses our method of decomposing the image set into smaller groups and also determining the connecting images between individual groups. Section 3 provides the overview of our registration process. Section 4 reports the results of our experiments on different data sets, and Section 5 concludes the paper.

2 Data set decomposition using normalised cuts

Images used for bundle adjustment can either be acquired from a site or aggregated from various sources on the internet. When the images are acquired

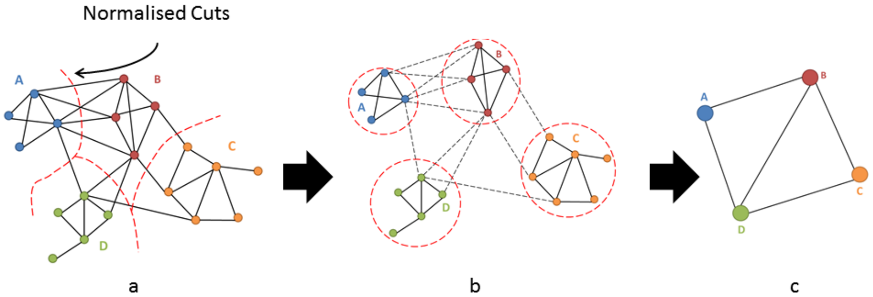


Fig. 2: (a) Original match graph where images (nodes) are connected by edges having similar image features. The edge weights represent similarity scores. (b) Normalised cut partitions the full image set into connected components which can be reconstructed independently. The "connecting images" across components are defined by the cut edges. (c) The individual cuts are now equivalent to individual nodes that represent independent rigid 3D reconstructions which are registered using pairwise epipolar relationship of the connecting images.

from a site in an organised manner, the problem of decomposition into smaller sets becomes trivial. In what follows we provide an illustration. Figure 3 shows the Google Earth view of the Vitthala temple at Hampi in Karnataka, India, which is a world heritage site maintained by the Archaeological Survey of India (Latitude: 15.342276, Longitude: 76.475287). Figure 4 shows a typical example where images of two buildings are captured separately and it also shows a typical *connecting* image which sees parts of both the buildings. We call such data sets *organised*.

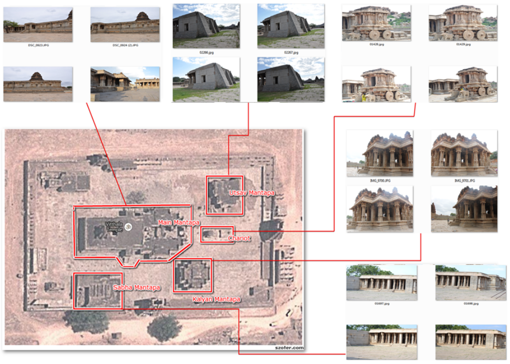


Fig. 3: Google Earth view of the Vitthala temple, Hampi, Karnataka, India. The red boxes denote different buildings of the temple. Images for each building were captured separately.

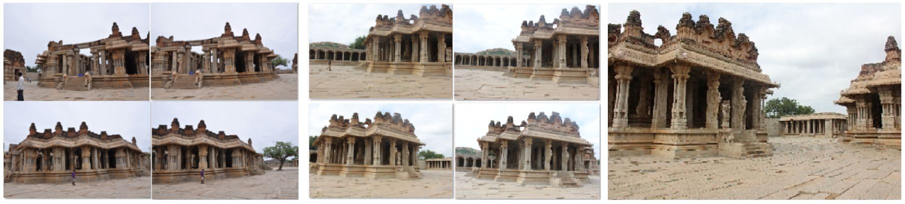
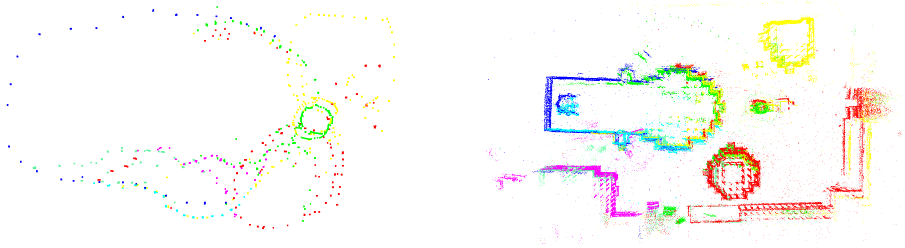


Fig. 4: (a) and (b) Two buildings of the Hampi temple complex, and (c) a typical connecting image.

In case such planned acquisition is not possible, the collection of images need to be automatically partitioned into smaller components. Unorganised data sets downloaded from the Internet are typical examples. In such cases a method for automatically grouping into visually similar sets and finding connecting images needs to be established. To this end, we train a vocabulary tree [18] using all image features (SIFTGPU [19]) and extract top p (typically p = 80) similar images for each image in the set. We form a match graph where each node is an image and the edge weights between two nodes are the similarity values obtained from the vocabulary tree. We aim to partition the set of images such that each partition is densely connected. The partitions only capture dense connectivity of matched image features and need not represent a single physical structure. Here the dense connectivity ensures that SFM reconstruction is less likely to fail due to the paucity of reliable matches or accumulated error or drift.

We use the multi-way extension [7] of the normalised cut (NC) formulation to automatically partition the match graph $\mathbf{G} = (\mathbf{V}, \mathbf{E})$ into individual clusters. Since, in our case edge weights are based on visual similarity, the normalised cut would yield those connected components in which connected images are visibly similar. We use the images that belong to the cut as candidate connecting images. In Figure 5 we show the result of our estimation upon applying the normalised cut to the set of images collected at the Hampi site illustrated in Figure 3, i.e. when we treat the images as an unorganised dataset. Figure 5a shows the cameras partitioned into connected components in different colours. Figure 5b shows the plan view of the 3D reconstructions obtained for each component marked in corresponding colours. It should be noted that in this example, the graph weights are based only on pairwise image feature similarity scores. We can improve the quality of the graph by incorporating geometric information such as the robustness of computation of pairwise epipolar geometries of connected images. Such a scheme would not only ensure that the connected pairs of images can be reliably matched but would also ensure that the pairwise epipolar geometries can be robustly estimated. The corresponding result is provided in Figure 8b and discussed in Section 4.

Extracting connecting images: The number of candidate connecting images are often very large. Reducing the number of connecting images will reduce



(a) Cameras partitioned into connected components. Each component is shown in a different colour.

(b) 3D reconstructions of each component marked in same colours.

Fig. 5: SfM results on the Hampi dataset (unorganised data) illustrating the effect of graph partitioning.

the time for estimation of pairwise epipolar geometry. The connecting image extraction process is described below:

1. For each of the connecting images reject the outlier **out edges** (both within and across components) using a measure of the robustness of the epipolar computation (Equation 6).
2. If the number of out edges retained is less than T (typically T = 60% of the original out degree) then remove the image from the set of connecting images.
3. Compute the mean of the similarity scores of all the retained out edges for the current image.
4. If the similarity score for a cut edge exceeds the mean similarity values of the images they connect, then mark the images as connecting images.

3 Registration of independent component reconstructions

In this Section, we describe how each of the individually reconstructed groups of cameras are aligned or registered to a single frame of reference. To register a pair of 3D reconstructions, we need to estimate the relative transformation between them. In what follows, we describe how we estimate relative rotation, translation and scale between a pair of reconstructions using epipolar relationships between the reconstructed cameras and the connecting cameras. While estimating epipolar geometry, we use focal lengths extracted from the EXIF information of the images.

Let us consider two independently reconstructed groups of cameras A and B[3]. Let $\mathbb{C}_{AB}$ be the set of connecting cameras between A and B. We first fix

[3] In this section, we use lower case letters to denote individual cameras and upper case letters to denote groups of reconstructed cameras.

the relative scale between A and B using the approach described in Section 3.1. Once this relative scale is fixed, the two reconstructions A and B are now related by a rigid or Euclidean transformation which can be estimated using the method detailed in Section 3.2.

3.1 Relative scale estimation between a pair of reconstructions

To estimate the relative scale between A and B, we first estimate the position of all the connecting cameras ($k \in \mathbb{C}_{AB}$) in the local frames of reference of both A and B separately. Then we compare the pairwise distances of common cameras in the two reconstructions. If a connecting camera $k \in \mathbb{C}_{AB}$ shares common features with the cameras in A then, the rotation and translation of k can be found in the local reference frame of A. Let the unknown rotation and translation of k, with respect to the frame of reference of A be denoted by R_{Ak} and T_{Ak} respectively. Now, consider a camera i that belongs to the group A. Suppose, the rotation and translation of i with respect to the local frame of reference of A is R_{Ai} and T_{Ai} respectively (as estimated within A). If $i \in A$ and $k \in \mathbb{C}_{AB}$ share common features, then using epipolar relationships, we can find the relative rotation (R_{ik}) and the direction of relative translation (t_{ik}) between i and k. Clearly the following relations should hold:

$$R_{ik} = R_{Ak}R_{Ai}^T \Rightarrow R_{Ak} = R_{ik}R_{Ai} \tag{1}$$

$$t_{ik} \propto T_{Ak} - R_{ik}T_{Ai} \Rightarrow [t_{ik}]_\times \left(T_{Ak} - R_{ik}T_{Ai}\right) = 0 \tag{2}$$

where, $[.]_\times$ is the skew-symmetric matrix representation for vector cross product [20]. These relations hold for all $i \in A$ such that i and k share common features and the epipolar geometry between them can be estimated. Therefore, we take the geodesic mean [21] of all such estimates of the rotation R_{Ak} as,

$$\widehat{R}_{Ak} = \underset{i \in A}{\text{mean}} \left(R_{ik}R_{Ai}\right) \tag{3}$$

Similarly, the average estimate of the translation T_{Ak} is obtained as

$$\widehat{T}_{Ak} = \underset{T_{Ak}}{\text{argmin}} \sum_{i \in A} \frac{\left\| [t_{ik}]_\times \left(T_{Ak} - R_{ik}T_{Ai}\right)\right\|^2}{\left\|T_{Ak} - R_{ik}T_{Ai}\right\|^2} \tag{4}$$

which can be solved using the iterative method proposed in [22].

The center of projection of camera k, in the frame of reference of A, is given by $-\widehat{R}_{Ak}\widehat{T}_{Ak}$. Thus, we compute the camera centers ($-\widehat{R}_{Ak}\widehat{T}_{Ak}$) for all $k \in \mathbb{C}_{AB}$ in the frame of reference of A. Similarly, we compute the camera centers ($-\widehat{R}_{Bk}\widehat{T}_{Bk}$) for all $k \in \mathbb{C}_{AB}$ in the frame of reference of B. Then, the relative scale between A and B can be robustly estimated by comparing pairwise distances of common cameras in the two reconstructions as:

$$\widehat{s}_{AB} = \underset{k_1, k_2 \in \mathbb{C}_{AB}}{\text{median}} \frac{\left\| -\widehat{R}_{Bk_1}\widehat{T}_{Bk_1} + \widehat{R}_{Bk_2}\widehat{T}_{Bk_2} \right\|}{\left\| -\widehat{R}_{Ak_1}\widehat{T}_{Ak_1} + \widehat{R}_{Ak_2}\widehat{T}_{Ak_2} \right\|} \tag{5}$$

Once the relative scale between A and B is estimated, we scale the reconstruction of A to have the same scale as that of B. Therefore, the rotation of camera k in the frame of reference of A remains $\widehat{R}_{Ak}$ whereas, the translation of camera k becomes $\widehat{s}_{AB}\widehat{T}_{Ak}$ in the scaled local frame of reference of A. We do not change the scale of B. Hence the rotation ($\widehat{R}_{Ak}$) and the translation ($\widehat{T}_{Ak}$) of camera k in the frame of reference of B remains unaltered.

It should be noted here that the translation directions estimated using epipolar geometry may have outliers. To remove outliers, we check whether the two non-zero eigenvalues of the essential matrix have similar values [20]. We discard the estimated essential matrix as well as the corresponding translation direction if the ratio of the two largest eigenvalues (σ_2 and σ_1 in sorted order) is less than a threshold, i.e.

$$\frac{\sigma_2}{\sigma_1} < T \tag{6}$$

Typically we take T as 0.95 for our experiments.

3.2 Relative rotation and translation estimation between a pair of reconstructions

Once A is resized to have same scale as that of B, the two reconstructions are related by a rigid or Euclidean transformation. Earlier, we estimated the motion of k in the frame of reference of A to be a rotation and translation of $\widehat{R}_{Ak}$ and $\widehat{s}_{AB}\widehat{T}_{Ak}$ respectively. Similarly, the motion of k in the frame of reference of B is $\widehat{R}_{Bk}$ and $\widehat{T}_{Bk}$ respectively.

In the following we denote the 3D rotation and translation compactly as the Euclidean motion model

$$\mathbf{M} = \left[\begin{array}{c|c} R & T \\ \hline \mathbf{0} & 1 \end{array}\right] \tag{7}$$

where $\mathbf{0}$ denotes a 1×3 vector of zeros. Suppose the unknown motions that align A and B to the global frame of reference be $\mathbf{M}_A$ and $\mathbf{M}_B$ respectively. After applying these transformations to A and B, all common connecting cameras ($k \in \mathbb{C}_{AB}$) between A and B should have the same motion parameters. Therefore, after A and B are registered with the global frame of reference, we have

$$\widehat{\mathbf{M}}_{Ak}\mathbf{M}_A = \widehat{\mathbf{M}}_{Bk}\mathbf{M}_B \tag{8}$$

where we reiterate that the translation component of $\widehat{\mathbf{M}}_{Ak}$ is the scaled version, i.e. $\widehat{s}_{AB}\widehat{T}_{Ak}$. Therefore, the relative motion between A and B is

$$\mathbf{M}_{AB} = \mathbf{M}_B\mathbf{M}_A^{-1} = \widehat{\mathbf{M}}_{Bk}^{-1}\widehat{\mathbf{M}}_{Ak} \tag{9}$$

From Equation 9 we can see that we have

$$R_{AB} = R_B R_A^T = \widehat{R}_{Bk}^T\widehat{R}_{Ak} \tag{10}$$

If there are many connecting cameras between A and B, then we have many estimates of R_{AB}, which we average to provide the estimate :

$$\widehat{R}_{AB} = \underset{k \in \mathbb{C}_{AB}}{\text{mean}} \left(\widehat{R}_{Bk}^T \widehat{R}_{Ak} \right) \tag{11}$$

Similarly, from Equation 9, the relative translation between A and B can be seen to be given by:

$$T_{AB} = T_B - R_B R_A^T T_A = \widehat{s}_{AB} \widehat{R}_{Bk}^T \widehat{T}_{Ak} - \widehat{R}_{Bk}^T \widehat{T}_{Bk} \tag{12}$$

Therefore, we estimate relative translation between A and B by robustly averaging all pairwise estimates from different connecting cameras as:

$$\widehat{T}_{AB} = \underset{T}{\text{argmin}} \sum_{k \in \mathbb{C}_{AB}} \left\| T - \left(\widehat{s}_{AB} \widehat{R}_{Bk}^T \widehat{T}_{Ak} - \widehat{R}_{Bk}^T \widehat{T}_{Bk} \right) \right\|_1 \tag{13}$$

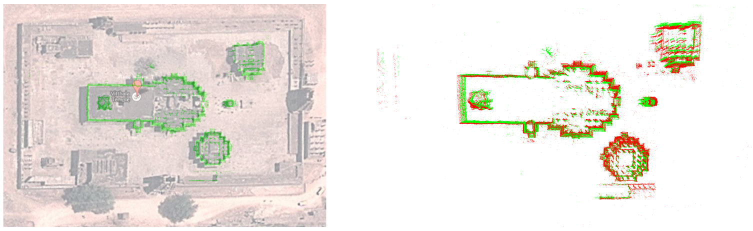
In our implementation, we start the process of global registration using the largest reconstruction (with maximum number of images) as the seed and register all other reconstructions which are connected to this seed and merge them into a single model. We also remark that the motion models required for registering individual reconstructions connected to the current model can be estimated in parallel.

4 Experimental Results

We present our results on both *organised* and *unorganised* image data sets. For our experiments, we used an Intel i7 quad core machine with 16GB RAM and GTX 580 graphics card. We first present our result on an *organised* image set acquired from Hampi (see Figure 3) . The data set consists of 2337 images covering 4 temple buildings. The physical footprint of these 4 buildings covers an area of approximately 160 × 94 metres.[4] For reconstructing the images in each individual set we use VSFM [1] as the iterative bundler. We merge each of these reconstructions using the method described in Section 3 into a common frame of reference. Figure 6a shows our reconstruction after registration superimposed on a view from Google Earth. As we do not have ground truth for such real-world data, to analyse the quality of our reconstruction we use the output of VSFM applied on the entire data set using all pairs matching as our baseline reconstruction. We note that all pairs matching is necessitated here as the scheme of preemptive matching suggested in [1] fails on this data set. Figure

[4] We point out here that these buildings are far more complex compared to urban buildings and even heritage sites such as the Notre Dame cathedral reconstructed in [4]. Specifically, these temples have fluted pillars, are repleted with ornate carvings and sculptures, repeated patterns as well as layered cupola. The complexity of these structures can also be judged from the building footprint as seen in the plan view presented in Figure 6a.

6b shows the comparison where the red point cloud is obtained from VSFM and the green points are obtained using our method. VSFM took 5760 minutes to reconstruct the data set using all pairs matching. In contrast our method takes 2578 minutes (using all pairs matching) to reconstruct the same data set. The computation time of our method is calculated by considering the time required for reconstruction of the largest component and the total time for registration, since the reconstruction of each component is done in parallel. We also compare the 3D camera rotations and positions (i.e. translations) obtained by our method against the 'ground truth' provided by VSFM. As the two camera estimates are in different frames of reference and may also differ in scale, we align them in a common Euclidean reference frame by computing the best similarity (Euclidean transformation and a global scale) transformation between them. The results of our comparison are presented in Table 1. Here, while the rotation error is in absolute degrees, since the overall scale of the reconstruction is arbitrary, we present the errors in translation (position) estimates as a fraction of the graph diameter of the full reconstruction. As can be seen, apart from being much faster than VSFM, our result is qualitatively similar to that obtained by VSFM.



(a) Reconstruction overlaid on Google map.

(b) Comparison between VSFM (red) and our method (green).

Fig. 6: Validation of reconstruction of Hampi data set (organised data).

Table 1: Comparison of total reconstruction by VSFM against individual reconstructions being registered by our method

Error entity	Error unit	Mean error	Median error	RMS error
Camera rotation	Degrees	1.93	1.57	2.66
Camera translation	Ratio of graph diameter	0.012	0.0091	0.041

For experimenting with unorganised image datasets we consider a total of 3017 images from the Hampi data set. We train a vocabulary tree [18] using SIFT [23, 19] features and take 80 most similar images from vocabulary tree for

Table 2: Data sets used in our experiments

Data set	No. of images	No. of components	No. of images reconstructed
Rome	13783	24	10534
Hampi	3017	7	2584
St Peter's Basilica	1275	5	1236
Colosseum	1164	3	1032

each image in the set to construct a match graph. Normalised cut is applied on this match graph and connected components are obtained. In our experiments, the expected number of connected components is decided intuitively and is used as an input parameter for the number of components needed using normalised cut. We use the process described in Section 2 to find the connecting images. We then run VSFM on individual connected components and merge them into a single coordinate frame. Figure 7 shows a frontal view of the reconstruction by our method. Figure 5b shows the 3D reconstructions corresponding to each of the connected components registered and in different colors. To validate our result, we overlay our reconstruction on the corresponding site map from Google Earth and Figure 8c shows that the registration is accurate. We also run VSFM with all 3017 images and compare the results. Figure 8a shows the comparison results where the VSFM output is marked in red and the output obtained using our method is marked in green. Figure 8b shows the corresponding results using a measure of robustness of epipolar estimation as edge weights in normalised cut. It can be noted that the results are marginally superior to that of Figure 8a especially near the top left corner of the plan view. This is because the images corresponding to this region are no longer distributed across different segments by normalised cut.

Table 3: Time statistics of our method on different data sets compared with VSFM

Data set	Match graph creation using vocabulary tree (mins)	Pairwise matching (mins)	Reconstruction and registration (mins)	Total time by us (mins)	Pairwise matching by VSFM (mins)	Reconstruction by VSFM (mins)	Total time by VSFM (mins)
Rome	768	502	**27**	**1297**	N/A	N/A	N/A
Hampi	481	424	**8**	**913**	9522	59	9581
St Peter's Basilica	98	22	**4**	**124**	1385	10	1395
Colosseum	83	24	**3**	**110**	1394	9	1403

We also tested our algorithm on some standard *unorganised* data sets downloaded from the Internet. We downloaded approximately 13K images of Central Rome from Flickr and tested our algorithm on this data set. Figure 9 shows the reconstruction using our method. This data set could not be reconstructed using VSFM with our hardware resources. Figure 9d shows the reconstruction overlaid on Google map. We also ran our algorithm on the St Peter's Basilica and Colosseum data sets obtained from [1], the results of which are shown in Figures 10 and 11 respectively. Table 2 shows the total number of connected components and the total number of images reconstructed for each of the data sets. The time statistics of our algorithm for different data sets are presented in Table 3. For most of the cases we had to use all pairs matching in VSFM as preemptive matching was causing the reconstruction to break in the middle, which is also reported in [1]. In our case we used the initial match graph obtained from vocabulary tree. It is evident that most of the time is consumed for matching. The reconstruction and the total registration time taken by our approach is significantly less than the reconstruction time of VSFM. The overall speed up achieved is at least one order of magnitude superior. We also note that iterative bundle adjustment schemes often results in broken reconstruction even within a component. In Table 4 we present statistics of such breaks. In all such cases we have been able to register the broken components using pairwise epipolar geometry on the connecting images in the broken components identified automatically from the match graph. Finally, we also remark in passing that we also experimented with the method presented in [17] using the author's code. On the Hampi dataset, [17] failed to reconstruct in more than 24 hours. While [17] is faster than original BA, its runtime complexity is far inferior to the $O(n)$ complexity of VSFM. In an additional test, for a 300 image subset of the Hampi dataset, [17] was 10 times slower than VSFM and produced a significantly poorer result.

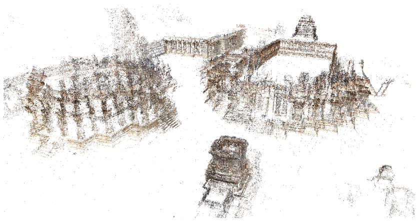
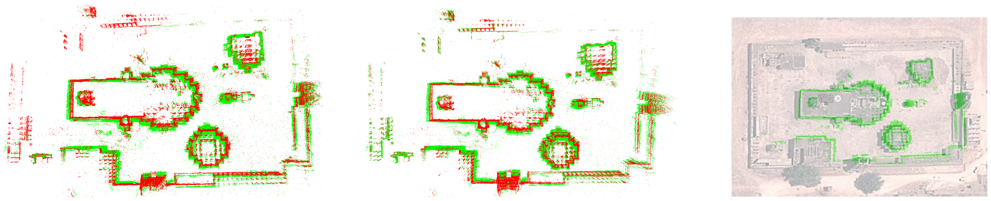


Fig. 7: Frontal view of the Hampi reconstruction (considered as unorganised data).

5 Conclusion

We have presented a new pipeline for automatic 3D reconstruction from a large collection of images. We have demonstrated the utility of partitioning the images into clusters that can be independently and reliably reconstructed and then aligned in a global frame of reference. Results on a number of large data sets demonstrates that our method results in large speed improvements compared to

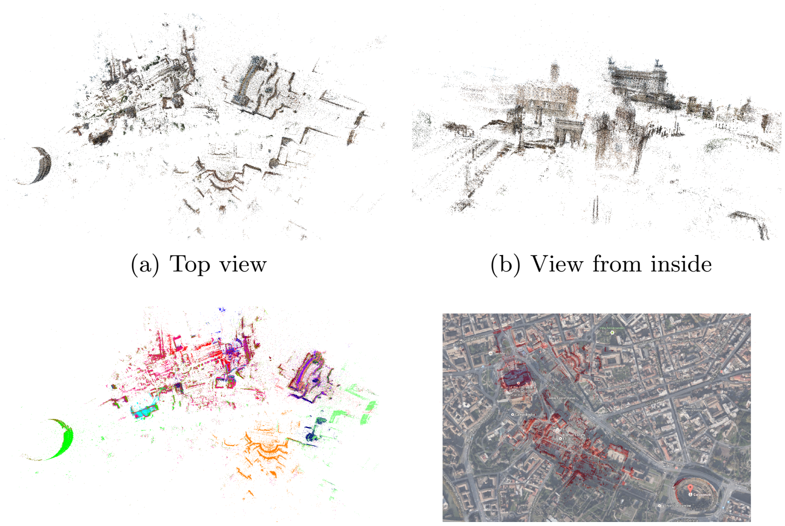


(a) Comparison with VSFM (red) and our method (green).

(b) Comparison with VSFM (red) and our method with epipolar robustness (green).

(c) Overlaid on Google map.

Fig. 8: Reconstruction of the Hampi data set (considered as unorganised data) validated against VSFM reconstruction and Google Earth.



(a) Top view

(b) View from inside

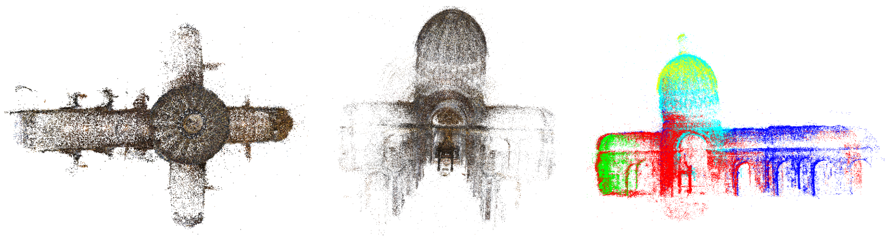
(c) Colored components

(d) Overlaid on Google map

Fig. 9: Reconstruction of Central Rome using our method.

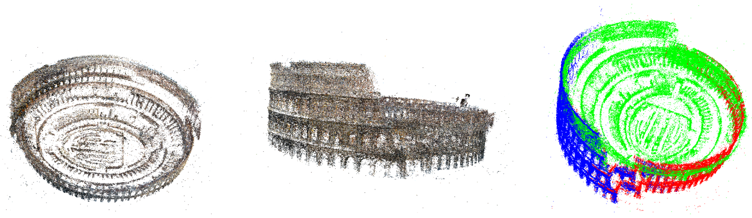
Table 4: Statistics of breaks in reconstruction of the data sets

Data set	No. of components	No. of components broken by VSFM	Total no. of components including broken sub-components
Rome	24	5	33
Hampi	7	2	9
St Peter's Basilca	5	1	6
Colosseum	3	2	6



(a) Top view (b) View from inside (c) Colored components

Fig. 10: Reconstruction of St Peter's Basilica using our method (1275 images used).



(a) Top view (b) Side view (c) Colored components

Fig. 11: Reconstruction of Colosseum using our method.

the state-of-the-art without any significant loss of accuracy.

Acknowledgement : The authors affiliated with the Indian Institute of Technology Delhi (BB, SP and SB) acknowledge the support of the Department of Science and Technology, Government of India under the Indian Digital Heritage programme.

References

1. Wu, C.: Towards linear-time incremental structure from motion. In: Proceedings of the International Conference on 3D Vision. 3DV '13 (2013) 127–134
2. Agarwal, S., Snavely, N., Seitz, S., Szeliski, R.: Bundle adjustment in the large. In: Proceedings of the European Conference on Computer Vision. (2010) 29–42
3. Snavely, N., Seitz, S., Szeliski, R.: Modeling the world from internet photo collections. International Journal of Computer Vision **80** (2008) 189–210
4. Snavely, N., Seitz, S., Szeliski, R.: Photo tourism: Exploring photo collections in 3d. In: Proceedings of ACM SIGGRAPH. (2006) 835–846
5. Crandall, D.J., Owens, A., Snavely, N., Huttenlocher, D.P.: Discrete-continuous optimization for large-scale structure from motion. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2011) 3001–3008

6. Triggs, B., Mclauchlan, P., Hartley, R., Fitzgibbon, A.: Bundle adjustment – a modern synthesis. In: Vision Algorithms: Theory and Practice, LNCS. (2000) 298–372
7. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000) 888–905
8. Wu, C., Agarwal, S., Curless, B., Seitz, S.: Multicore bundle adjustment. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2011) 3057–3064
9. Frahm, J., Georgel, P., Gallup, D., Johnson, T., Raguram, R., Wu, C., Jen, Y., Dunn, E., Clipp, B., Lazebnik, S., Pollefeys, M.: Building rome on a cloudless day. In: Proceedings of the European Conference on Computer Vision: Part IV. (2010) 368–381
10. Raghuram, R., Wu, C., Frahm, J., Lazebnik, S.: Modeling and recognition of landmark image collections using iconic scene graphs. International Journal of Computer Vision **95** (2011) 213–239
11. Agarwal, S., Snavely, N., Simon, I., Seitz, S., Szeliski, R.: Building rome in a day. In: Proceedings of the International Conference on Computer Vision. (2009) 72–79
12. Snavely, N., Seitz, S., Szeliski, R.: Skeletal graphs for efficient structure from motion. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2008) 1–8
13. Havlena, M., Torii, A., Pajdla, T.: Efficient structure from motion by graph optimization. In: Proceedings of the European Conference on Computer Vision. Volume 6312 of Lecture Notes in Computer Science. (2010) 100–113
14. Crandall, D.J., Owens, A., Snavely, N., Huttenlocher, D.P.: SfM with MRFs: Discrete-continuous optimization for large-scale reconstruction. IEEE Transactions on Pattern Analysis and Machine Intelligence **35** (2013) 2841–2853
15. Moulon, P., Monasse, P., Marlet, R.: Global fusion of relative motions for robust, accurate and scalable structure from motion. In: Proceedings of IEEE International Conference on Computer Vision. (2013) 3248–3255
16. Jiang, N., Cui, Z., Tan, P.: A global linear method for camera pose registration. In: Proceedings of IEEE International Conference on Computer Vision. (2013) 481–488
17. Farenzena, M., Fusiello, A., Gherardi, R.: Structure-and-motion pipeline on a hierarchical cluster tree. In: Proceedings of IEEE International Conference on Computer Vision Workshop on 3-D Digital Imaging and Modeling. (2009) 1489–1496
18. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. Volume 2. (2006) 2161–2168
19. Wu, C.: SiftGPU: A GPU implementation of scale invariant feature transform (SIFT). http://cs.unc.edu/ ccwu/siftgpu (2007)
20. Hartley, R., Zisserman, A.: Multiple View Geometry in Computer Vision. 2nd edn. Cambridge University Press, New York (2004)
21. Govindu, V.M.: Lie-algebraic averaging for globally consistent motion estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2004)
22. Govindu, V.: Combining two-view constraints for motion estimation. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition. (2001) 218–225
23. Lowe, D.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110