Structure-and-Motion Pipeline on a Hierarchical Cluster Tree

Michela Farenzena, Andrea Fusiello, Riccardo Gherardi
Dipartimento di Informatica
Università di Verona
Strada Le Grazie 15, 37134 Verona (Italy)
name.surname@univr.it

Abstract

This papers introduces a novel hierarchical scheme for computing Structure and Motion. The images are organized into a tree with agglomerative clustering, using a measure of overlap as the distance. The reconstruction then follows this tree from the leaves to the root. As a result, the problems is broken into smaller instances, which are then separately solved and combined. Compared to the standard sequential approach, this framework has a lower computational complexity, it is independent from the initial pair of views, and copes better with drift problems. A formal complexity analysis and some experimental results support these claims.

1. Introduction

In recent years there has been a surge of interest in automatic architectural/urban modeling from images.

Literature covers several approaches for solving this problem. These can be categorized in two main branches: A first one is composed of methods specifically tailored for urban environments and engineered to run in real-time [6, 24]. These systems usually rely on a host of additional information, such as GPS/INS navigation systems and camera calibration.

The second category – where our contribution is situated – comprises Structure and Motion (SaM) pipelines that process images in batch and handle the reconstruction process making no assumptions on the imaged scene and on the acquisition rig [2, 16, 30, 36, 15].

The main issue to be solved in this context is the scalability of the SaM pipeline. This prompted a quest for efficiency that has explored several different solutions: the most successful have been those aimed at reducing the impact of the bundle adjustment phase, which – with feature extraction – dominates the computational complexity.

A class of solutions that have been proposed are the so-called *partitioning methods* [9]. They reduce the reconstruction problem into smaller and better conditioned subproblems which can be effectively optimized. In this paper we propose a new hierarchical scheme for SaM which provably cuts the computational complexity by one order of magnitude. The images are organized into a hierarchical cluster tree, as in Figure 1. This approach has some analogy with [27], where a spanning tree is built to establish in which order the images must be processed. After that, however, the images are processed in a standard incremental way. In our case, instead, the reconstruction proceeds hierarchically along this tree from the leaves to the root. Partial reconstructions correspond to internal nodes, whereas images are stored in the leaves. The parent node contains the merger of the two partial reconstructions associated to its children. The global complexity is trimmed further by limiting the number of views employed per node, with the introduction of a local bundle adjustment strategy. Beside improving in complexity, this framework copes better with initialization and drift problems, typical of sequential schemes.

Figure 1. An example of dendrogram for a 12-views set.

The use of partitioning methods for SaM has been already studied in the literature. Two main strategies can be distinguished.

The first one is to tackle directly the bundle adjustment algorithm, exploiting its properties and regularities. The idea is to split the optimization problem into smaller, more tractable components. The subproblems can be selected analytically as in [32], where spectral partitioning has been applied to SaM, or they can emerge from the underlying 3D structure of the problem, as described in [22]. The com-

putational gain of such methods is obtained by limiting the combinatorial explosion of the algorithm complexity as the number of images and feature points increases.

The second strategy is to select a subset of the input images and feature points that subsumes the entire solution. Hierarchical sub-sampling was pioneered by [9], using a balanced tree of trifocal tensors over a video sequence. The approach was subsequently refined by [23], adding heuristics for redundant frames suppression and tensor triplet selection. In [28] the sequence is divided into segments, which are resolved locally. They are subsequently merged hierarchically, eventually using a representative subset of the segment frames. A similar approach is followed in [11], focusing on obtaining a well behaved segment subdivision and on the robustness of the following merging step. The advantage of these methods over their sequential counterparts lays in the fact that they improve error distribution on the entire dataset and bridge over degenerate configurations. Anyhow, they work for video sequences, so they cannot be applied to unordered, sparse images.

A recent work [31] that works with sparse dataset describes a way to select a subset of images whose reconstruction provably approximates the one obtained using the entire set. This considerably lowers the computational requirements by controllably removing redundancy from the dataset. Even in this case, however, the images selected are processed incrementally.

Our strategy reaps the benefits of most the aforementioned methods: i) it applies to unorganized fairly large sets of images, ii) it partitions the problem into smaller instances and combines them hierarchically, iii) it is efficient and inherently parallelizable, iv) it is less sensible to typical problems of sequential approaches, namely sensitivity to initialization [33] and drift [6].

The rest of the paper is organized as follows. The next section outlines the matching stage, then Sec. 3 describes the way the hierarchical cluster tree is built. Section 4 presents our hierarchical approach, whereas the local bundle adjustment strategy is explained in Sec. 5. Experimental results are reported in Sec. 6, and finally conclusions are drawn in Sec. 7.

2. Keypoint Matching

In this section we describe the stage of our SaM pipeline that is devoted to the automatic extraction and matching of keypoints among all the n available images. Its output is to be fed into the geometric stage, that will perform the actual structure and motion recovery.

Although the building blocks of this stage are fairly standard techniques, we carefully assembled a procedure that is fully automatic, robust (matches are pruned to discard as much outliers as possible) and computationally efficient.

First of all, the objective is to identify in a computationally efficient way images that potentially share a good number of keypoints, instead of trying to match keypoints between every image pair (they are $O(n^2)$). We follow the approach of [1]. SIFT [18] keypoints are extracted in all n images. In this culling phase we consider only a constant number of descriptors in each image (we used 300, where a typical image contains thousands of SIFT keypoints). Then, each keypoint description is matched to its ℓ nearest neighbors in feature space (we use $\ell = 6$). This can be done in $O(n \log n)$ time by using a k-d tree to find approximate nearest neighbors (we used the ANN library [20]). A 2D histogram is then built that registers in each bin the number of matches between the corresponding views. Every image will be matched only to the m images that have the greatest number of keypoints matches with it (we use $m = 8$). Hence, the number of images to match is $O(n)$, being m constant. For example, on the *Pozzoveggiani* dataset composed by 54 images, the matching time is reduced from 13:40 hrs to 50 min. A further reduction in the computing time could be achieved by leveraging the processing power of modern GPUs.

Matching follows a nearest neighbor approach [18], with rejection of those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than a threshold (set to 1.5 in our experiments).

Homographies and fundamental matrices between pairs of matching images are then computed using MSAC [35]. Let e_i be the residuals after MSAC, following [38], the final set of inliers are those points such that

$$|e_i - \operatorname{med}_j e_j| < 3.5\sigma^*, \tag{1}$$

where σ^* is a robust estimator of the scale of the noise:

$$\sigma^* = 1.4826 \operatorname{med}_i |e_i - \operatorname{med}_j e_j|. \tag{2}$$

This outlier rejection rule is called X84 in [12].

The model parameters are eventually re-estimated on this set of inliers via least-squares minimization of the (first-order approximation of the) geometric error [19, 4].

The more likely model (homography or fundamental matrix) is selected according to the Geometric Robust Information Criterion (GRIC) [34]. Finally, if the number of remaining matches between two images is less than a threshold (computed basing on a statistical test as in [1]) then they are discarded.

After that, keypoints matching in multiple images are connected into *tracks*, rejecting as inconsistent those tracks in which more than one keypoint converges [30] and those shorter than three frames.

3. Views Clustering

The second stage of our pipeline consists in organizing the available views into a hierarchical cluster structure that will guide the reconstruction process.

Algorithms for image views clustering have been proposed in literature in the context reconstruction [27], panoramas [1], image mining [26] and scene summarization [29]. The distance being used and the clustering algorithm are application-specific.

In this paper we deploy an image affinity measure that befits the structure-and-motion reconstruction task. It is computed by taking into account the number of common keypoints and how well they are spread over the images.In formulae, let S_i and S_j be the set of matching keypoints in image I_i and I_j respectively:

$$a_{i,j} = \frac{1}{2}\frac{|S_i \cap S_j|}{|S_i \cup S_j|} + \frac{1}{2}\frac{CH(S_i) + CH(S_j)}{A_i + A_j} \qquad (3)$$

where $CH(\cdot)$ is the area of the convex hull of a set of points and A_i (A_j) is the total area of image I_i (I_j). The first term is an affinity index between sets, also known as Jaccard index. The distance is $(1 - a_{i,j})$, as $a_{i,j}$ ranges in $[0, 1]$.

Views are grouped together by agglomerative clustering, which produces a hierarchical, binary cluster tree, called *dendrogram*. The general agglomerative clustering algorithm proceeds in a bottom-up manner: starting from all singletons, each sweep of the algorithm merges the two clusters with the smallest distance. The way the distance between clusters is computed produces different flavors of the algorithm, namely the simple linkage, complete linkage and average linkage [7]. We selected the *simple linkage* rule: The distance between two clusters is determined by the distance of the two closest objects (nearest neighbors) in the different clusters.

Simple linkage clustering is appropriate to our case because: i) the clustering problem *per se* is fairly simple, ii) nearest neighbors information is readily available with ANN and iii) it produces "elongated" or "stringy" clusters which fits very well with the typical spatial arrangement of images sweeping a certain area or a building.

As will be clarified in the next section, the clusters composed by two views are the ones from which the reconstruction is started. These two views must satisfy two conflicting requirements: have both a large number of keypoints in common and a baseline sufficiently large so as to allow a well-conditioned reconstruction. The first requirement is automatically verified as these clusters are composed by the closest views according to the affinity defined in (3). The second requisite is tantamount to say that the fundamental matrix must explain the data far better than other models (namely, the homography), and this can be implemented by considering the GRIC, as in [25].

We therefore modify the linkage strategy so that two views i and view j are allowed to merge in a cluster only if:

$$\text{gric}(F_{i,j}) < \alpha\ \text{gric}(H_{i,j}) \quad \text{with } \alpha \geq 1, \qquad (4)$$

where $\text{gric}(F_{i,j})$ and $\text{gric}(H_{i,j})$ are the GRIC scores obtained by the fundamental matrix and the homography matrix respectively (we used $\alpha = 1.2$). If the test fail, consider the second closest elements and repeat.

4. Hierarchical Structure and Motion

The dendrogram produced by the clustering stage imposes a hierarchical organization of the views that will be followed by our SaM pipeline. At every node in the dendrogram an action must be taken, that augment the reconstruction (cameras + 3D points). There operations are possible: When a cluster is created a two-views reconstruction must be performed. When a view is added to a cluster a resection-intersection step must be taken (as in the standard sequential pipeline). When two clusters are joined together an absolute orientation problem must be solved. Each of these steps is detailed in the following.

Two-views reconstruction. We assume that at least the cameras from which the two-views reconstruction is performed are calibrated. This can be obtained by off-line calibration or by autocalibration [10].

The extrinsic parameters of two given views are obtained by factorizing the essential matrix, as in [14]. Then 3D points are reconstructed by *intersection* (or triangulation) and pruned using X84 on the reprojection error. Bundle adjustment is run eventually to improve the reconstruction.

One-view addition. The reconstructed 3D points that are visible in the view to be added provides a set of 3D-2D correspondences, that are exploited to solve an exterior orientation problem via a linear algorithm [8], or resection with DLT [13] in case that the view is not calibrated. MSAC is used in order to cope with outliers.

The 3D structure is then updated with tracks that are visible in the last view. Three-dimensional points are obtained by intersection, and successively pruned by carrying out X84 on the reprojection error. As a further caution, 3D points for which the intersection is ill-conditioned are discarded, using a threshold on the condition number of the linear system (10^4, in our experiments). Finally, bundle adjustment is run on the current reconstruction.

Clusters merging. The two reconstructions that are to be merged live in two different reference systems, therefore one has to be registered onto the other with a similarity transformation (or collineation, in case that at least one reconstruction is not calibrated). They have, by construction,

some 3D points in common, that are used to solve an absolute orientation problem with MSAC. Once the cameras are registered, the common 3D points are re-computed by intersection, with the same cautions as before, namely X84 on the reprojection error and test of the conditioning number. Intersection is also performed on any track that becomes visible after the merging. The new reconstruction is finally refined with bundle adjustment.

At the end, the number of reconstructed points in the final reconstruction is increases by triangulating the the tracks of length two, with outlier rejection (X84) based on the reprojection error.

4.1. Complexity analysis

The hierarchical approach that have been outlined above allows to decrease the computational complexity with respect to the sequential SaM pipeline. Indeed, if the number of views is n and every view adds a constant number of points ℓ to the reconstruction, the computational complexity[1] in time of sequential SaM is $O(n^5)$, whereas the complexity of our hierarchical SaM (in the best case) is $O(n^4)$, as it will be shown in App. A.

The worst case is when a single cluster is grown by adding one view at a time. In this case, which corresponds to the sequential case, the dendrogram is extremely unbalanced and the complexity drops to $O(n^5)$. On the average we found empirically that dendrograms are fairly balanced, so we claim that in practice the best-case complexity is attained.

5. Local bundle adjustment

In the pursue of a further complexity reduction, we adopted a strategy that consist in selecting a constant number k of views from each cluster C to be used in the bundle adjustment in place of the whole cluster. These *active views*, however, are not fixed once for all, but they are defined opportunistically with reference to the object that is is being added, either a single view or another cluster C'. This strategy is an instance of local bundle adjustment [37, 21], which is often used for video sequences, where the *active views* are the most recent ones.

Let us concentrate on the cluster merging step, as the one view addition is a special case of the latter. Consider the set of point that belongs to both clusters C and C': we first single out the views in C and C' where these points are visible. Among these views, we select the k closest pairs, according to the distance matrix already computed in Sec. 3, to be the active views.

[1]We are considering here only the cost of bundle adjustment, which clearly dominates the other operations.

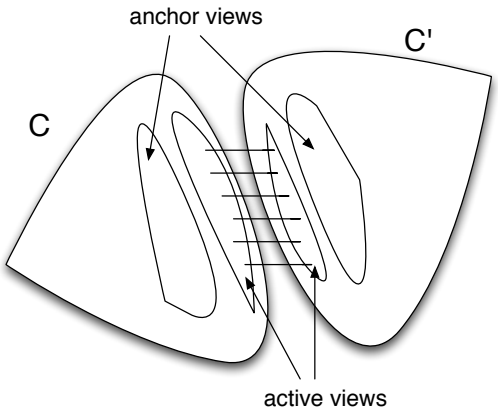

Figure 2. Local bundle adjustment. The active views are the k closest pairs between the two clusters with 3D points in common. They will be moved by bundle adjustment. The anchor views are the k closest views to the active ones inside each cluster. They contribute to the reprojection error, but are not affected by bundle adjustment.

The bundle adjustment involves the active views and the points that are reconstructable from them as variables, plus some other *anchor views* that are only used to compute the reprojection error. The anchor views are the k closest views to the active ones inside each cluster; they are not moved by bundle adjustment but contributes to anchor the 3D points involved to the remaining structure, acting as a damper that gives more rigidity to the piece of structure which is being bundle adjusted. Fig.2 illustrates this idea.

At the end, a bundle adjustment with all the views and all the points can be customarily run to obtain the optimal solution. If this is not feasible because of the dimension of the dataset, this strategy is able to produce a sub-optimal result anyway.

5.1. Complexity analysis

Every bundle adjustment but the last is now run on a constant number of views, hence its cost is $O(1)$. The number of bundle adjustments is $O(n)$, therefore the total cost is dominated by the final bundle adjustment, which is $O(n^4)$. Although the asymptotic complexity is the same as before, the local bundle adjustment clearly reduces the total number of operations.

The same complexity $O(n^4)$ is achieved by the sequential approach coupled with the local bundle adjustment. However, the hierarchical approach is easily parallelizable, and it is more robust and effective, as the experiments in the next section will show.

Figure 3. Two perspective views of the reconstruction of "Piazza Erbe" (Verona, Italy).

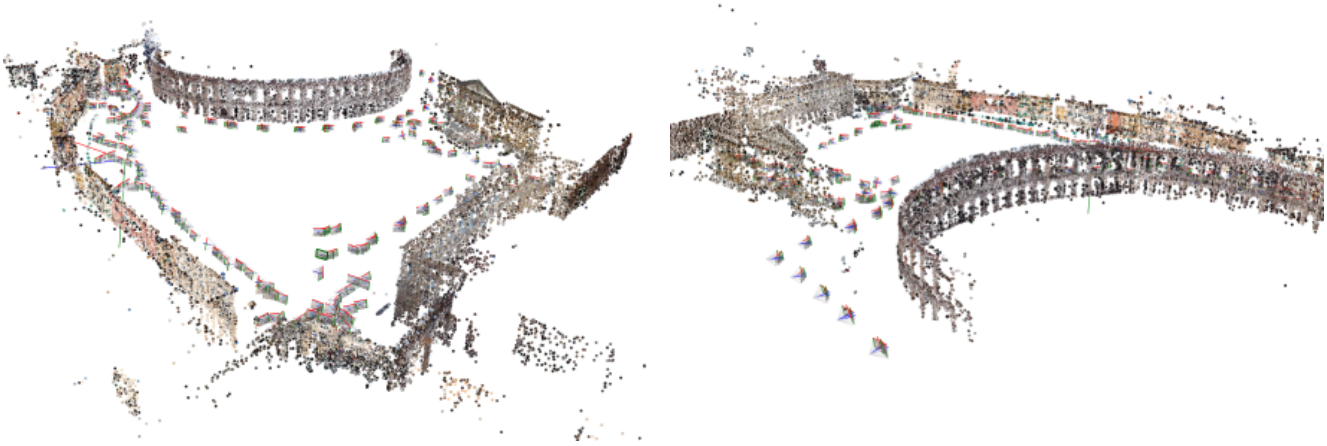

Figure 4. Two perspective views of the reconstruction of "Piazza Bra" with the Arena (Verona, Italy).

6. Experiments

We tested our algorithm (henceforth called SAMANTHA) on several datasets of pictures taken by the authors with a consumer camera with known internal parameters (available on the WWW[2]). Figure 3 and 4 illustrates the reconstruction from the "Piazza Erbe" and "Piazza Bra" datasets, respectively.

We compared our results with those produced by BUNDLER [3], an implementation of a state-of-the-art sequential SaM pipeline in C++. Inside our pipeline we used the C++ implementation of bundle adjustment (BA) described in [17]. Only time spent doing BA is reported, in order to factor out the differences due to our software being partially written in Matlab and to be consistent with our complexity analysis. Moreover, BUNDLER is extremely slow in the matching phase, as it matches every view to every other. All experiments were run on the same hardware (Intel Core2 Duo E4600@2.4Ghz, 2Gb ram).

Table 1 reports the result of the comparison. The results show that SAMANTHA takes significantly less time than BUNDLER, without any major differences in terms of number of reconstructed views and points.

As an example, Figure 5 and 6 show the top views of the final structure obtained with the two methods in the "Piazza Erbe" and "Piazza Bra" datasets, respectively, aligned and superimposed to an aerial image.

As a sequential algorithm, BUNDLER is very sensitive to initialization. Indeed, for some datasets it was necessary to carefully select the initial pair in order to make it produce a meaningful solution. In the case of "Piazza Bra", a total of four initial pairs were tried: the one chosen by default and three others selected with the same criterion employed by our clustering. In all cases, the result is only a partial reconstruction (witnessed by the small number of points reconstructed), with evident misalignments (Fig. 6). A similar result occurs for the "Tribuna" dataset.

In Table 2 we analyze the tradeoff between the number of active views, the computing time and the quality of the reconstruction for the local BA strategy. As expected, the computing time gracefully decreases as the number of

[2] http://profs.sci.univr.it/∼fusiello/demo/samantha/

		BUNDLER			SAMANTHA			
Dataset	# images	# views	# points	time BA	#views	#points	time BA	speedup
Dante	39	39	18360	7:50 m	39	10500	3:13 m	2.4
Tribuna	47	35	7722	22:58 m	39	10427	2:55 m	7.8
Pozzoveggiani	52	50	22133	21:33 m	48	11094	4:24 m	4.8
Madonna	73	73	25390	37:16 m	69	15518	10:04 m	3.7
Piazza Erbe	259	228	67436	5:18 h	198	39961	1:05 h	4.9
Piazza Bra	380	273	38145	11:36 h	322	104047	3:22 h	3.4

Table 1. Comparison between SAMANTHA and BUNDLER. Each row lists, for the two approaches: name of the dataset; number of images; number of reconstructed views; number of reconstructed points; BA running time. The last column reports the speedup achieved by our algorithm.

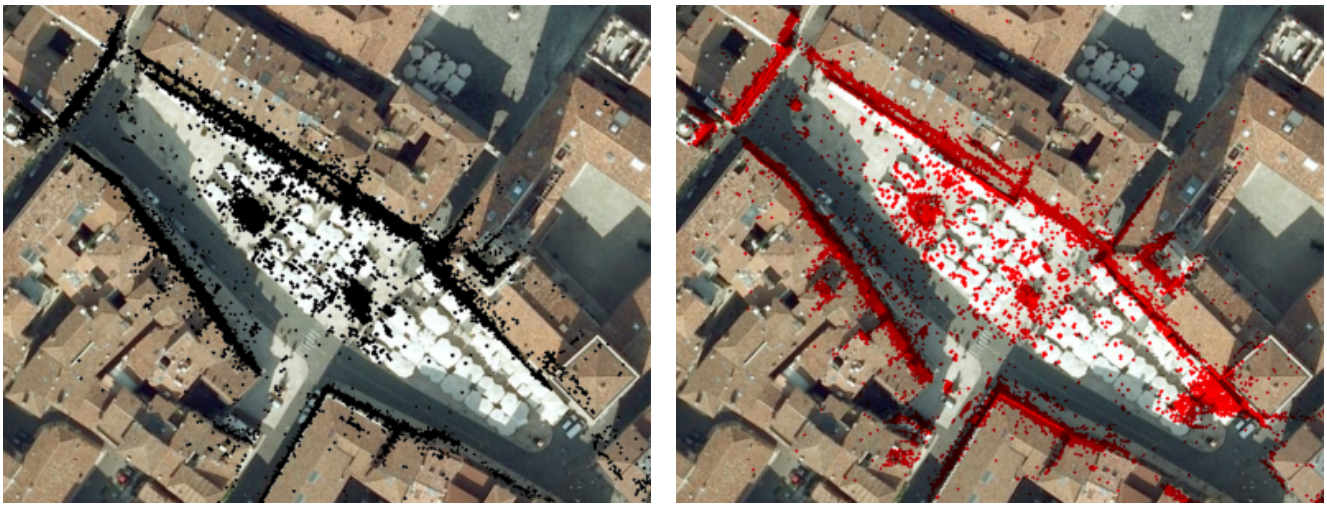

Figure 5. Top views aligned with an aerial image of "Piazza Erbe" (from Google Earth), reconstructed with SAMANTHA (left) and with BUNDLER (right).

active views diminishes, without any appreciable loss in terms of reconstructed points and views. Small variations in the number of points and views are expected and normal even among identical runs of the algorithm, because of non-deterministic steps. Accordingly, the average alignment error with respect to the baseline case (all active views) increases.

Eventually, when using very few active views, SAMANTHA could fail to merge clusters. Before that happens we noticed an increase in the BA running time due to the larger number of iterations needed by the bundle adjustment to converge in less than ideal settings. This prompt us to suggest using sufficiently large (20+) number of active views to ensure fast and reliable computing.

For a qualitative comparison, in Figure 7 we registered two top views of the final structure obtained with and without local BA (we used 15 active views).

7. Conclusions and Future Work

We have developed a novel Structure and Motion pipeline that provably boost computational efficiency by one order of magnitude, thanks to a hierarchical scheme based on views clustering. Beside being more efficient than the sequential one, our algorithm is more effective, because it is insensitive to initialization and copes better with drift problems.

Future research will aim at pushing forward the limits of our approach with larger and larger datasets by leveraging on its inherently parallel nature.

# active	time BA	speedup	# points	# views	error
all	1:05 h	1	39961	192	0 m
35	26:16 m	2.33	40641	196	0.45 m
25	24:53 m	2.63	40373	196	0.48 m
15	22:25 m	2.94	40669	198	0.75 m

Table 2. Reconstruction results vs number of active views for "Piazza Erbe" dataset. Each row lists: the number of active views; the BA running time; the speedup achieved; the number of reconstructed points; the number of reconstructed views; the average alignment error wrt to the baseline (all active). The metric scale have been obtained from Google Earth.

Acknowledgments

Thanks to Roberto Posenato and Cheng Dong Seon for many useful suggestions. The use of VLFeat by A. Vedaldi

Figure 6. Top views aligned with an aerial image of "Piazza Bra" (from Google Earth), reconstructed with SAMANTHA (left) and with BUNDLER (right).

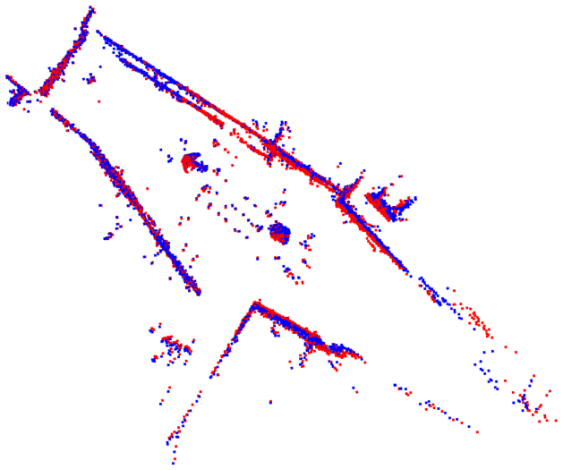

Figure 7. Comparison between the result obtained by SAMANTHA with (in red) and without (in black) local BA.

and B. Fulkerson, ANN by David M. Mount and Sunil Arya, SBA by M. Lourakis and A. Argyros, and Bundler by N. Snavely is gratefully acknowledged.

A. Complexity analysis

The cost of bundle adjustment with m points and n views is $O(mn(m+2n)^2)$ [28], hence it is $O(n^4)$ if $m = \ell n$.

In the sequential SaM, adding view i requires a constant number of bundle adjustments (typically one or two) with i views, hence the complexity is

$$\sum_{i=1}^{n} O(i^4) = O(n^5). \tag{5}$$

In the case of the hierarchical approach, consider a node of the dendrogram where two clusters are merged into a cluster of size n. The cost $T(n)$ of adjusting that cluster is given by $O(n^4)$ plus the cost of doing the same onto the left and right subtrees. In the hypothesis that the dendrogram is well balanced, i.e., the two clusters have the same size, this cost is given by $2T(n/2)$. Hence the asymptotic time complexity T in the best case is given by the solution of the following recurrence:

$$T(n) = 2T(n/2) + O(n^4) \tag{6}$$

that is $T(n) = O(n^4)$ by the third branch of the Master's theorem [5].

References

[1] M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, pages 1218–1225, October 2003. 2, 3

[2] M. Brown and D. G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *Proceedings of the International Conference on 3D Digital Imaging and Modeling*, June 2005. 1

[3] http://phototour.cs.washington.edu/bundler/. 5

[4] O. Chum, T. Pajdla, and P. Sturm. The geometric error for homographies. *Computer Vision and Image Understanding*, 97(1):86–102, 2005. 2

[5] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, USA, 2001. 7

[6] N. Cornelis, B. Leibe, K. Cornelis, and L. V. Gool. 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2-3):121–141, July 2008. 1, 2

[7] R. O. Duda and P. E. Hart. *Pattern Classification and Scene Analysis*, pages 98–105. John Wiley and Sons, 1973. 3

[8] P. D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001. 3

[9] A. W. Fitzgibbon and A. Zisserman. Automatic camera recovery for closed and open image sequencese. In *Proceedings of the European Conference on Computer Vision*, pages 311–326, 1998. 1, 2

[10] A. Fusiello, A. Benedetti, M. Farenzena, and A. Busti. Globally convergent autocalibration using interval analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(12):1633–1638, December 2004. 3

[11] S. Gibson, J. Cook, T. Howard, R. Hubbold, and D. Oram. Accurate camera calibration for off-line, video-based augmented reality. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 2002. 2

[12] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel. *Robust Statistics: the Approach Based on Influence Functions*. Wiley Series in probability and mathematical statistics. John Wiley & Sons, 1986. 2

[13] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2nd edition, 2003. 3

[14] R. I. Hartley. Estimation of relative camera position for uncalibrated cameras. In *Proceedings of the European Conference on Computer Vision*, pages 579–587, 1992. 3

[15] A. Irschara, C. Zach, and H. Bischof. Towards wiki-based dense city modeling. In *Proceedings of the 11th International Conference on Computer Vision*, pages 1–8, 2007. 1

[16] G. Kamberov, G. Kamberova, O. Chum, S. Obdrzalek, D. Martinec, J. Kostkova, T. Pajdla, J. Matas, and R. Sara. 3D geometry from uncalibrated images. In *Proceedings of the 2nd International Symposium on Visual Computing*, November 6-8 2006. 1

[17] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. Technical Report 340, Institute of Computer Science - FORTH, Heraklion, Crete, Greece, August 2004. 5

[18] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004. 2

[19] Q.-T. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1996. 2

[20] D. M. Mount and S. Arya. Ann: A library for approximate nearest neighbor searching. In *http://www.cs.umd.edu/ mount/ANN/*, 1996. 2

[21] E. Mouragnon, M. Lhuillier, M. Dhome, F. Dekeyser, and P. Sayd. Real time localization and 3d reconstruction. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition*, pages 363–370, 2006. 4

[22] K. Ni, D. Steedly, and F. Dellaert. Out-of-core bundle adjustment for large-scale 3D reconstruction. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007. 1

[23] D. Nistér. Reconstruction from uncalibrated sequences with a hierarchy of trifocal tensors. In *Proceedings of the European Conference on Computer Vision*, pages 649–663, 2000. 2

[24] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles. Detailed real-time urban 3D reconstruction from video. *International Journal of Computer Vision*, 78(2-3):143–167, 2008. 1

[25] M. Pollefeys, F. Verbiest, and L. V. Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the European Conference on Computer Vision*, pages 837–851, 2002. 3

[26] T. Quack, B. Leibe, and L. Van Gool. World-scale mining of objects and events from community photo collections. In *Proceedings of the International Conference on Content-based Image and Video Retrieval*, pages 47–56, 2008. 3

[27] F. Schaffalitzky and A. Zisserman. Multi-view matching for unordered image sets, or "how do i organize my holiday snaps?". In *Proceedings of the 7th European Conference on Computer Vision*, pages 414–431, 2002. 1, 3

[28] H.-Y. Shum, Q. Ke, and Z. Zhang. Efficient bundle adjustment with virtual key frames: A hierarchical approach to multi-frame structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, June 1999. 2, 7

[29] I. Simon, N. Snavely, , and S. M. Seitz. Scene summarization for online image collections. In *Proceedings of the Internation Conference on Computer Vision*, 2007. 3

[30] N. Snavely, S. M. Seitz, and R. Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pages 835–846, 2006. 1, 2

[31] N. Snavely, S. M. Seitz, and R. Szeliski. Skeletal graphs for efficient structure from motion. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, 2008. 2

[32] D. Steedly, I. Essa, and F. Dellaert. Spectral partitioning for structure from motion. In *Proceedings of the International Conference on Computer Vision*, pages 649–663, 2003. 1

[33] T. Thormählen, H. Broszio, and A. Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In *Proceedings of the European Conference on Computer Vision*, pages 523–535, 2004. 2

[34] P. H. S. Torr. An assessment of information criteria for motion model selection. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 47–53, 1997. 2

[35] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:2000, 2000. 2

[36] M. Vergauwen and L. V. Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006. 1

[37] Z. Zhang and Y. Shan. Incremental motion estimation through modified bundle adjustment. In *Proceedings of the International Conference on Image Processing*, pages II–343–6, Sept. 2003. 4

[38] M. Zuliani. *Computational Methods for Automatic Image Registration*. PhD thesis, University of California, Santa Barbara, Dec 2006. 2