Improving the Efficiency of Hierarchical Structure-and-Motion

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

Abstract

We present a completely automated Structure and Motion pipeline capable of working with uncalibrated images with varying internal parameters and no ancillary information. The system is based on a novel hierarchical scheme which reduces the total complexity by one order of magnitude. We assess the quality of our approach analytically by comparing the recovered point clouds with laser scans, which serves as ground truth data.

1. Introduction

In recent years, Structure and Motion (SaM) pipelines able of processing batches of images and to output a 3D reconstruction without making any assumptions on the imaged scene and on the acquisition rig [3, 15, 24, 31, 14] are making their way toward "real world" applications and out of their former intended scope of urban reconstruction.

The main challenges to be solved are computational efficiency (in order to be able to deal with more and more images) and generality.

As for the first issue, several different solutions has been explored: 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* [8]: they reduce the reconstruction problem into smaller and better conditioned subproblems which can be effectively optimized. 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[26, 17]. The computational 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 [8, 18, 23, 9]. 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 paper [25] 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. Moreover, this method does not avoid computing the epipolar geometry between *all* pairs of images.

There is actually a third solution covered in literature, orthogonal to the aforementioned approaches. In [1], the computational complexity of the reconstruction is tackled by throwing additional computational power to the problem. Within such framework, the former algorithmical challenges are substituted by load balancing and subdivision of reconstruction tasks. Such direction of research strongly suggest that the current monolithical pipelines should be modified to accommodate ways to parallelize and optimally split the workflow of reconstruction tasks.

In this direction goes [6], in which the authors propose a new hierarchical and parallelizable scheme for SaM. The images are organized into a hierarchical cluster tree, and 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. This scheme provably cuts the computational complexity by one order of magnitude (provided that the dendrogram is well balanced), and it is less sensible to typical problems of sequential approaches, namely sensitivity to initialization [27] and drift [5]. This approach has some analogy with [21], 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.

The generality issue – mentioned before – refers to the assumptions made on the input images, or, equivalently to the amount of ancillary information that is required in addition to pixels values. Existing pipelines either assumes known internal parameters [3, 14], or constant internal parameters [31, 15], or relies on EXIF data plus external informations (camera CCD dimensions) [24]. To the best of our knowledge, despite autocalibration with varying parameters have been introduced several years ago [19], no working SaM pipeline have been demonstrated yet with varying parameters *and* no ancillary information.

The contribution of this paper are twofold: first, building on [6], we introduce a clustering strategy derived from the simple linkage that makes the dendrogram more balanced, thereby reducing the *actual complexity* of the method. Second, we endow the SaM pipeline with the capability of dealing with uncalibrated images with varying internal parameters and no ancillary information. To this end, we devised a quasi-Euclidean initialization step that is crucial to make autocalibration converge.

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 with the balancing heuristics. Section 4 presents the hierarchical approach to structure and motiohn recovery, whereas the autocalibration strategy is explained in Sec. 5. Experimental results are reported in Sec. 6, and finally conclusions are drawn in Sec. 7.

2. Keypoint Matching

The keypoint matching stage is fairly standard, and mainly follows the approach of [2] and [24]. The details are reported in [6]. The output of this stage is a set of *tracks*, i.e., keypoints matching in more than three images, and a set of fundamental matrices and homographies linking pairs of views, each one endowed with GRIC (Geometric Robust Information Criterion) [28] scores, that reveals which of the two models is more likely.

3. Views Clustering

The second stage of the pipeline proposed in [6] consists in organizing the available views into a hierarchical cluster structure (a tree) that will guide the reconstruction process. This procedure allows to decrease the computational complexity with respect to the sequential SaM pipeline, from $O(n^5)$ to $O(n^4)$ in the best case, i.e. when the tree is well balanced (n is the number of views). If the tree is unbalanced this computational gain vanishes. It is therefore crucial to enforce the balancing of the tree and this is the goal of the technique that we shall describe in this section.

Figure 1. An example of one image (top left) from "Piazza Bra" and its six closest neighbors according to the affinity defined in Eq. 1.

The method starts from an affinity matrix among views, computed using the following measure, that takes into account the number of common keypoints and how well they are spread over the images:

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

where S_i and S_j are the set of matching keypoints in image I_i and I_j respectively, $CH(\cdot)$ is the area of the convex hull of a set of points and A_i (A_j) is the total area of the image. Figure 1 shows an example of the neighborhood defined by this affinity.

Then, 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. In particular, [6] uses the simple linkage strategy, which specifies that the distance between two clusters is to be determined by the distance of the two closest objects (nearest neighbors) in the different clusters.

In order to produce better balanced trees, we modified the agglomerative clustering strategy as follows: starting from all singletons, each sweep of the algorithm merges the pair with the smallest cardinality among the ℓ closest pair of clusters. The distance is computed according to the simple linkage rule. The cardinality of a pair is the sum of the cardinality of the two clusters.

In this way we are softening the "closest first" agglomerative criterion by introducing a competing "smallest first" principle that tends to produce better balanced dendrograms. The amount of balancing is regulated by the parameter ℓ: when $\ell = 1$ this is the standard agglomerative clustering with no balancing; when $\ell \geq n/2$ (n is the number of views) a perfect balanced tree is obtained, but the clustering is poor, since distance is largely disregarded. We found in our experiments (see Sec. 6) that a good compromise is $\ell = 5$.

Figure 2 shows an example of balancing achieved by our

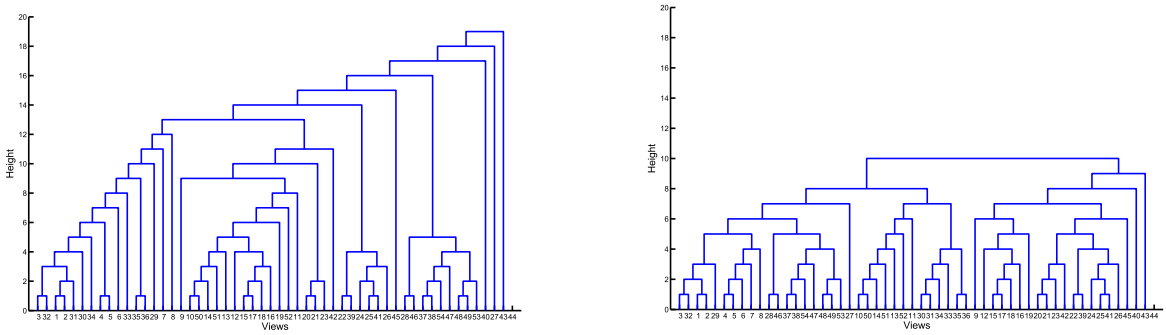

Figure 2. An example of the dendrogram produced by [6] (left) and the more balanced dendrogram produced by our technique (right) on a 52-views set, with $\ell = 5$.

technique. The height of the tree is reduced from 14 to 9 and more initial pairs are present in the dendrogram on the right.

There is a caveat, however, that must be applied when building clusters of cardinality two. These are pair of images from which the reconstruction will start, hence care must be taken to avoid pairs related by homographies. This is tantamount to say that the fundamental matrix must explain the data far better than an homography, and this can be implemented by considering the GRIC, as in [20]. We therefore modify the linkage strategy so that two views i and view j are allowed to merge in a cluster only if:

$$\text{gric}(F_{i,j}) < \alpha\, \text{gric}(H_{i,j}) \quad \text{with } \alpha \geq 1, \tag{2}$$

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): a two views reconstruction is performed when a cluster is first created, then there can be the addition of a single view to an existing cluster or the merging of two clusters. The first two are the typical operations of a sequential pipeline, whereas the latter is unique to the hierarchical pipeline.

The reconstruction starts uncalibrated, and as soon as an uncalibrated cluster reaches a given dimension m, the Euclidean upgrade procedure is triggered[1]. Please note that autocalibration is triggered only for nodes (clusters) of cardinality $\geq m$ with both children of cardinality $< m$, otherwise, if the cardinality of one child was $\geq m$ it would have been already upgraded to Euclidean.

[1] In principle autocalibration with known skew and aspect ratio requires a minimum of $m = 4$ views to work; for good measure we used $m = 12$.

4.1. Two-views reconstruction.

The reconstruction from two views is always projective in this pipeline, and proceeds from the fundamental matrix. It is well known that the following two camera matrices:

$$P_1 = [I \mid \mathbf{0}] \quad \text{and} \quad P_2 = [[\mathbf{e}_2]_\times F \mid \mathbf{e}_2], \tag{3}$$

yield the fundamental matrix F, as can be easily verified.

This canonical pair is related to the correct one (up to a similarity) by a collineation H of 3D space. Section 5 will describe how to guess a matrix H that provides a well conditioned starting point for the subsequent autocalibration step.

Given the upgraded versions of the perspective projection matrices $P_1 H$ and $P_2 H$, the position in space of the 3D points is then obtained by triangulation (Sec. 4.1.1) and projective bundle adjustment is run to improve the reconstruction.

4.1.1 Triangulation.

Triangulation (or intersection) is performed by the iterated linear LS method [12]. Points are pruned by analyzing the condition number of the linear system and the reprojection error. The first test discards ill-conditioned 3D points, using a threshold on the condition number of the linear system (10^4, in our experiments). The second test applies the so-called X84 rule [10], that establishes that, if e_i are the residuals, the inliers are those points such that

$$|e_i - \text{med}_j\, e_j| < 5.2\, \text{med}_i\, |e_i - \text{med}_j\, e_j|. \tag{4}$$

4.2. 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 glue the view to the cluster. This can be done by linear exterior orientation [7] or by resection with DLT [11], depending on whether the cluster corresponds to a Euclidean or projective reconstruction (a single view is always uncalibrated). MSAC [29] is used in both cases in order to cope with outliers. The view that has been glued might have brought in some new tracks, that are triangulated as described before (Sec. 4.1.1). Finally, bundle adjustment is run on the current reconstruction (either Euclidean or projective).

4.3. Clusters merging.

When two clusters merge the respective reconstructions live in two different reference systems, that are related by a similarity – if both are Euclidean – or by a projectivity of the space – if one is uncalibrated. The points that they have in common are the tie points that serve to the purpose of computing the unknown transformation, using MSAC to discard wrong matches. When merging a Euclidean cluster and a projective one, an homography of the projective space is sought that brings the second onto the first, thereby obtaining the correct Euclidean basis for the second. Once the cameras are registered, the common 3D points are recomputed by triangulation (Sec. 4.1.1), and the tracks obtained after the merging as well. The new reconstruction is eventually refined with bundle adjustment (either Euclidean or projective).

5. Autocalibration

As we saw previously, the reconstruction starts uncalibrated and the Euclidean upgrade procedure is triggered as soon as a cluster reaches a given dimension m. Hence, we assume that a projective reconstruction is available, and we want to upgrade it to the Euclidean level, using the constraints coming from the dual absolute quadric (DIAQ) [30]. The DIAQ elegantly encodes both intrinsic parameters and the plane at infinity in a single 4×4 matrix of rank 3. The basic relationship that links the DIAQ to the absolute conic is the following:

$$\omega_i^* = K_i \, K_i^\top = P_i \, \Omega_\infty \, P_i^\top \tag{5}$$

This equation can be used to transfer known constraints over the calibration parameters of each camera to the DIAQ which, being symmetric, can be encoded as a homogeneous vector of ten elements. The resulting linear system can then be solved using the direct linear transform algorithm, eventually followed by a non-linear optimization step.

Our implementation of the iterative dual linear autocalibration algorithm is based on the method described in [22], modified to use the weights of [20] and to enforce at every iteration the positive (negative) semi-definitess of the DIAQ. As explained in [13], the closest semi-definite approximation of a matrix in Frobenius norm can be obtained, assuming a single offending value, zeroing the eigenvalue with sign different from the others. This can be easily done during the rank-three approximation step of the original algorithm. Several informal tests, not reported here, demonstrated this algorithm to have better convergence properties of both its parents [22, 20].

Nevertheless, a good initialization is crucial to obtain convergence to the correct solution. To this end we devised an initialization procedure that provides a useful, well conditioned starting point for the subsequent autocalibration step. The canonical pair of camera matrices

$$P_1 = [I \mid \mathbf{0}] \quad \text{and} \quad P_2 = [Q_2 \mid \mathbf{e}_2], \tag{6}$$

is related to the Euclidean one by a collineation H of 3D space that has the following structure:

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & 1 \end{bmatrix}. \tag{7}$$

If reasonable assumptions on internal parameters of the cameras K_1 and K_2 can be made, the most critical part in order to recover the Euclidean structure is having a good estimate of the plane at infinity, which is related to $\mathbf{v}$ in Eq. 7. In the following we shall see how to obtain a consistent estimate of $\mathbf{v}$ given a guess of K_1 and K_2.

The upgraded, metric versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{8}$$

$$P_2^E = K_2 \, [R_2 | \mathbf{t}_2] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{e}_2 \mathbf{v}^\top | \mathbf{e}_2 \right] \tag{9}$$

The rotation R_2 can therefore be equated to the following:

$$R_2 \simeq K_2^{-1} \left(Q_2 K_1 + \mathbf{e}_2 \mathbf{v}^\top \right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{10}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term. Let R^* be the rotation such that: $R^* \mathbf{t}_2 = [\|\mathbf{t}_2\| \; 0 \; 0]^\top$. Left multiplying it to Eq. 10 yields:

$$R^* R_2 \simeq \overbrace{R^* \, K_2^{-1} Q_2 K_1}^{W} + [\|\mathbf{t}_2\| \; 0 \; 0]^\top \, \mathbf{v}^\top \tag{11}$$

Calling the first term W and its rows $\mathbf{w}_i^\top$, we arrive at the following:

$$R^* \, R_2 = \begin{bmatrix} \mathbf{w}_1^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2^\top \\ \mathbf{w}_3^\top \end{bmatrix} / \|\mathbf{w}_3\| \tag{12}$$

in which the last two rows of the right hand side are independent from the value of $\mathbf{v}$. Since the rows of the right hand side form a orthonormal basis, we can recover the first

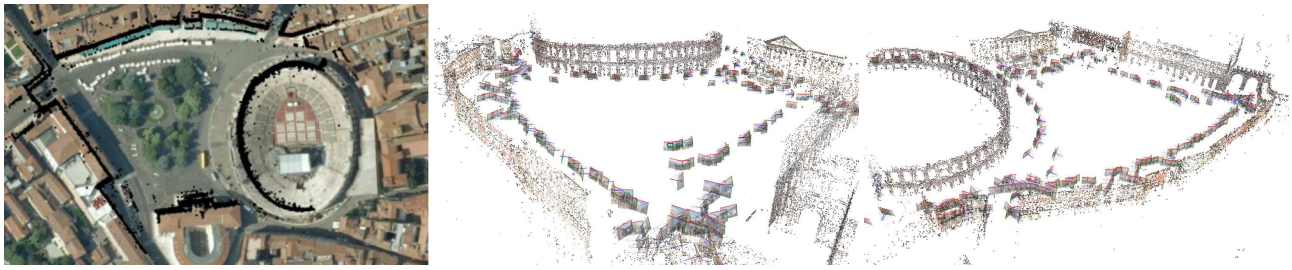

Figure 3. A top view and two perspective views of the reconstruction of "Piazza Bra" (Verona, Italy).

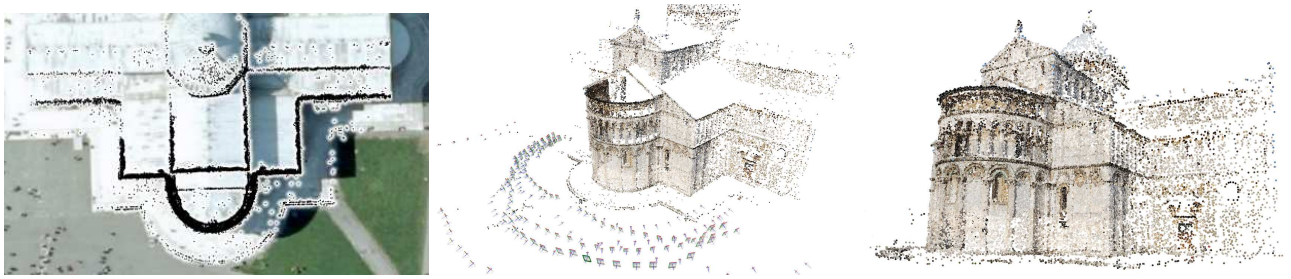

Figure 4. A top view and two perspective views of the reconstruction of "Duomo" (Pisa, Italy).

one taking the cross product of the other two. Vector $\mathbf{v}$ is therefore equal to:

$$\mathbf{v} = (\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1) / \|\mathbf{t}_2\| \qquad (13)$$

As a guess on the internal parameters we used, as customary, zero skew, unit aspect ratio and focal length equal to the magnitude of the image diagonal.

In case of failure of the autocalibration procedure the reconstruction remains projective and will be upgraded later on by merging with a Euclidean one.

Up in the tree, after autocalibration, an estimate of the internal parameters of each camera is available. They will be refined further with bundle adjustment as the reconstruction proceeds. In order to not to hamper the computation too much, the internal parameters of a camera becomes fixed as soon as they have been bundle-adjusted together with at least k cameras (we used $k = 25$).

6. Experiments

We tested our pipeline (henceforth called SAMANTHA+) on several datasets of pictures. Here we report the largest that have been used, namely "Piazza Bra" (from http://profs.sci.univr.it/~fusiello/demo/samantha/) and "Duomo" (courtesy of Visual Computing Lab (ISTI-CNR), Pisa). Figure 3 and 4 illustrate the reconstruction from these datasets.

Our pipeline works with uncalibrated images with varying internal parameters. The "Duomo" dataset contains pictures taken with three different camera settings, whereas "Piazza Bra" was originally taken with constant parameters. We therefore added 31 images taken from Flickr to the dataset, and discarded the information of which images are from the original dataset. In such a way, the internal parameters of each camera are treated independently of the others.

6.1. Time efficiency

We compared our results with those produced by SAMANTHA [6] and by BUNDLER [4] (an implementation of a state-of-the-art sequential SaM pipeline in C++).

Table 1 reports the result of the comparison with BUNDLER. Only time spent doing BA (C++ implementation of [16]) is reported, because BA dominates the computational complexity after matching, and BUNDLER is extremely slow in the matching phase, as it matches every view to every other. Moreover our pipeline is partially written in Matlab, so the total execution time would have been meaningless. All experiments were run on the same hardware (Intel Core2 Duo E4600@2.4Ghz, 2Gb ram).

The figures show that SAMANTHA+ takes significantly less time than BUNDLER, without any major difference in terms of number of reconstructed views and points. The total speed up achieved with respect to bundler is 13 and 4.8 for "Piazza Bra" and "Duomo" respectively, which compares favorably with the speed-up reported in [25] (on different dataset, though).

The improvement in the computing time is achieved thanks to the balancing strategy in the construction of the dendrogram. The effect of this strategy can be appraised in Fig. 5, where the number of reconstructed points/views and the computing time for "Piazza Bra" are plotted as the number of closest pairs ℓ is increased. After $\ell = 5$ the com-

Table 1. Comparison between BUNDLER and SAMANTHA+. Each row lists, for the two approaches: name of the dataset; number of images; number of reconstructed views; number of reconstructed points; running time (only BA).

		BUNDLER			SAMANTHA+		
Dataset	# img	# views	# points	time	#views	#points	time
Piazza Bra	411	292	41703	12:16 h	332	53634	58 min
Pisa	309	309	105401	13:43 h	309	121047	2:57 h

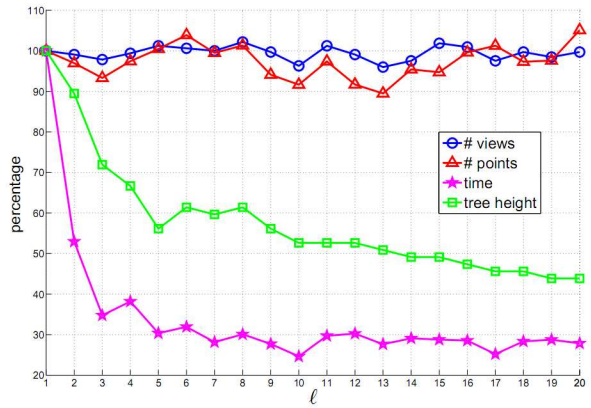

Figure 5. This plot shows the number of reconstructed points, views, height of the tree and computing time as a function of the parameter ℓ in the balancing heuristics. The values on the ordinate are in percentage with respect to the baseline case $\ell = 1$ which correspond to the original simple-linkage clustering of [6].

puting time stabilizes at around 30% of the baseline case (which corresponds to SAMANTHA [6]), without any significant difference in terms of number of reconstructed views and points.

It can also be noted that, as theory prescribes in [6], the computing time is directly linked to the height of the tree.

6.2. Metric accuracy

Thanks to the availability of ground truth for both datasets obtained from laser scanning, we were able to assess the accuracy of our results. We subsampled the cloud of points generated from laser scanners in such a way that they have roughly double the number of points of our reconstruction, then we run Iterative Closet Point (ICP) in order to find the best similarity that brings our data onto the model (see Fig. 6). The residual distances between closest pairs are measured and their average – the reconstruction accuracy – is about 35cm for "Piazza Bra" and 15cm for "Duomo". The final error of BUNDLER on the same datasets is 17cm for "Duomo", whereas for "Piazza Bra" BUNDLER failed to produce a meaningful result.

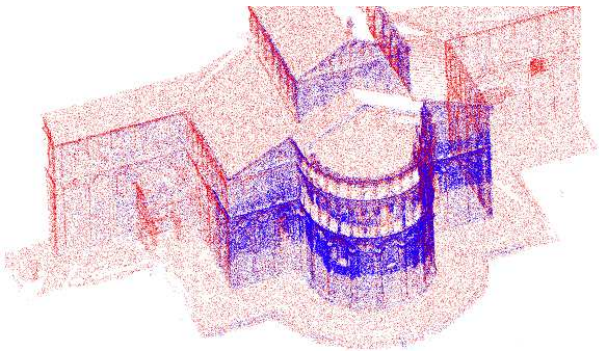

Figure 6. A view of "Duomo" reconstruction (blue) superimposed to the ground truth (red).

7. Conclusions and Future Work

We presented a novel Structure and Motion pipeline that – for the first time – deals with uncalibrated images with varying internal parameters and no ancillary information. Moreover the pipeline improves on efficiency, with respect to the state of the art, thanks to a hierarchical scheme based on a balanced agglomerative clustering of the images. The accuracy of our approach has been assessed analytically by comparing the recovered point clouds with laser scans, which serves as ground truth data. Future work will aim at improving the efficiency of the overall pipeline and the performances of the autocalibration procedure. Data and additional material are available from from http://profs.sci.univr.it/∼fusiello/demo/samantha/.

Acknowledgments

Thanks to Enrico Previdi who implemented part of the code. The laser scanning of "Piazza Bra" have been conducted by Gexcel s.r.l. with the EU JRC - Ispra and the permission of the municipality of Verona. The laser data of the "Duomo di Pisa" comes from the "Cattedrale Digitale[2]" project, while the photo set is courtesy of the Visual Computing Lab (ISTI-CNR, Pisa). The use of VLFeat by A. Vedaldi 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.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. M. Seitz, and R. Szeliski. Building rome in a day. In *International Conference on Computer Vision*, Kyoto, Japan, 2009. 1

[2] http://vcg.isti.cnr.it/cattedrale

[2] 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] 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, 2

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

[5] 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

[6] M. Farenzena, A. Fusiello, and R. Gherardi. Structure-and-motion pipeline on a hierarchical cluster tree. In *IEEE International Workshop on 3-D Digital Imaging and Modeling*, Kyoto, Japan, October 3-4 2009. 1, 2, 3, 5, 6

[7] P. D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001. 4

[8] 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

[9] 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. 1

[10] 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. 3

[11] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2003. 4

[12] R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997. 3

[13] N. J. Higham. Computing a nearest symmetric positive semidefinite matrix. *Linear Algebra and its Applications*, 103:103 – 118, 1988. 4

[14] 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, 2

[15] 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, 2

[16] 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

[17] 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

[18] 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. 1

[19] M. Pollefeys, R. Koch, and L. Van Gool. Self-calibration and metric reconstruction in spite of varying and unknown internal camera parameters. In *Proceedings of the International Conference on Computer Vision*, pages 90–95, Bombay, 1998. 2

[20] 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, 4

[21] 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

[22] Y. Seo, A. Heyden, and R. Cipolla. A linear iterative method for auto-calibration using the DAC equation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, page 880, 2001. 4

[23] 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. 1

[24] 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

[25] 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. 1, 5

[26] 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

[27] 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. 1

[28] 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

[29] 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. 4

[30] B. Triggs. Autocalibration and the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 609–614, Puerto Rico, 1997. 4

[31] M. Vergauwen and L. V. Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006. 1, 2