Riccardo Gherardi

Advances in 3D Reconstruction

Ph.D. Thesis

March 15, 2010

Università degli Studi di Verona
Dipartimento di Informatica

Advisor:
prof. Andrea Fusiello

Series N°: **TD-04-10**

Università di Verona
Dipartimento di Informatica
Strada le Grazie 15, 37134 Verona
Italy

A mia madre,
al suo sostegno nel tempo, e al suo esempio.

God thinks geometrically.
Plato, 428 BC – 348 BC

Foreword

We live in exciting times for Computer Vision. The progress in the 3D reconstruction research has been rapid and hectic, fueled by the recent breakthroughs in keypoint matching, the advances in computational power of desktop and mobile devices, the advent of digital photography and the subsequent availability of large dataset of public images, the interest of industry and general public.

Just five years ago Structure and Motion research was mainly using controlled dataset recovered in carefully managed, closed lab environments. Today, the goal of finally bridging the gap between physical reality and the digital world seems within reach given the magnitude, breadth and scope of current reconstruction pipelines.

We document here our contributions to the current state of the art: this thesis is divided in two parts exploring related, orthogonal goals.

Part 1 will discuss how to solve the Structure and Motion problem in a truly scalable and robust manner. We will describe our main result, a novel hyerarchical framework for 3D reconstruction, and propose a new self-calibration approach capable of dealing with crowd-sourced datasets.

Part 2 will deal in how to properly convey, visualize the obtained 3D reconstructions from the previous part. This will touch stereo matching, to densify the obtained point clouds, model fitting, to achieve a more abstract, semantic understanding of the scene, and different visualization modalities tailored to urban environments.

Acknowledgements

There are things that can be learnt studying, while others can only be understood through experience, or by example. I am truly grateful and indebted to my supervisor, Andrea Fusiello, which has been during these years both a caring mentor and a model of human and scientific integrity. His contagious passion for research, the boundless technical expertise have been an invaluable and inspiring support.

I would also wish to thank Michela Farenzena and Roberto Toldo, which I was lucky enough to collaborate with on several projects. They have been helpful, reliable, knowledgeable but more than everything else good friends.

Finally, the VIPS laboratory has been a great place to be, full of smart, fun, friendly people. Thanks to everyone who contributed to make it such a stimulating and friendly environment.

Riccardo Gherardi
Verona, March 15, 2010

Contents

List of Figures

List of Tables

1

Introduction

Three dimensional reconstruction is the process of recovering the properties of the environment and optionally of the sensing instrument from a series of measures.

This generic definition is wide enough to accomodate very diverse methodologies, such as time-of-flight laser scanning, photometric stereo or satellite triangulation.

The "Structure and Motion" field of research is concerned with the recovery of the three dimensional geometry of the scene (the structure) when observed through a moving camera (the motion). Sensor data is either a video or an unstructured set of pictures; additional informations, such as the calibration parameters, can be used if available.

This thesis will describe our contributions to the problem of uncalibrated Structure and Motion from pictures which, in layman terms, is the problem of recovering a three dimensional model of a scene given a set of images. The sought result is generally a 3D point cloud consisting of the interest points which were identified and tracked in the scene and a set of camera matrices, identifying position and direction of each picture with respect to an arbitrary reference frame.

We will approach the problem from a holistic point of view: we will propose improvements to the tecnique itself but will also describe means to produce efficient and compelling rendering of the obtained models. These two arguments correspond to the two main parts into which this thesis is divided.

The first part, covering our contribution to the Structure and Motion problem, will describe our main breakthrough, a hierarchical Structure and Motion pipeline. Current state of the art follows typically a sequential pattern, incrementally growing a seed reconstruction adding one or several views at a time. We instead propose to describe the entire reconstruction process as a binary tree, constructed by agglomerative clustering over the set of views. Each leaf correspond to a single image, while internal nodes represent partial reconstructions obtained merging the left and right sub-nodes. Reconstruction proceed from bottom to top, starting from several seed couple and eventually reaching the root node, corresponding to the final, complete result. We will demonstrate that this paradigm

has several advantages over the sequential one, both in terms of computational performance (which improves by one order of magnitude on average) and overall error containement. Such a system provides true scalability, since it is inherently parallelizable.

We will also describe our approach to self-calibration, which is the process of automatic estimation from images of the internal parameters of the cameras that captured them. Current Structure and Motion research has partly sidestepped the issue using ancillary data such as EXIF tags embedded in recent image formats. Their presence or consistency however is not guaranteed, and poses a problem especially when the number of images is not big enough to provide sufficient information to jumpstart the reconstruction process. Our proposal is a novel self-calibration algorithm comparable to the current state of the art, at a fraction of the complexity. Its robustness will be tested against the same datasets used for Structure and Motion tasks; we will therefore demonstrate the first uncalibrated Structure and Motion pipeline capable of using crowd-sourced picture datasets.

The second part of this thesis will be spent describing how to use the recovered data. This will touch both rendering algorithm, in order to provide visualization closer to the captured environment, and model fitting and selection, which was introduced to achieve a more compact and semantically meaningful representation of the scene. We will explore tecniques for the automatic extraction of dominant planes and quadrics, and we will use them to cluster reconstructed points into the original surfaces of the scene, described either as a triangulated mesh or a textured surface plus displacement.

Part I

Structure and Motion

2

Hierarchical Structure-and-Motion

In this chapter, we will describe our approach to 3D reconstruction describing the three major revisions that our Structure and Motion pipeline has undergone. Our final result will be a reconstruction pipeline departing from the sequential paradigm prevalent in current literature and embracing instead a hierarchical approach. Our method has several advantages, like a provably lower computational complexity which is necessary to achieve true scalability and better error containement leading to more stability and less drift.

2.1 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 [19, 76]. These systems usually rely on a host of additional information, such as GPS/INS navigation systems and camera calibration.

The second category – where our contributions are 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 [8, 49, 51, 90, 100].

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* [32]. 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. The subproblems can be selected analytically as in [91], where spectral partitioning has been applied to SaM, or they can emerge from the underlying 3D structure of the problem, as described in [69]. 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. Hierarchical sub-sampling was pioneered by [32], using a balanced tree of trifocal tensors over a video sequence. The approach was subsequently refined by [70], adding heuristics for redundant frames suppression and tensor triplet selection. In [87] 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 [35], 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 [89] 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.

The generality issue refers to the assumption that are made on the input images, or, equivalently on the amount of ancillary information that is required in addition to pixels values. Existing pipelines either assumes known internal parameters [8, 49], or constant internal parameters [51, 100], or relies on EXIF data plus external informations (camera CCD dimensions) [90]. To the best of our knowledge, despite autocalibration with varying parameters have been introduced several years ago [75], no working SaM pipeline have been demonstrated yet with varying parameters *and* no ancillary information.

We propose a new hierarchical and parallelizable scheme for SaM. The images are organized into a hierarchical cluster tree, as in Figure 2.1, 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 [93] and drift [19]. It is also scalable and efficient, since it partitions

the problem into smaller instances and combines them hierarchically, making it inherently parallelizable.

Fig. 2.1. An example of dendrogram for a 12-views set.

This approach has some analogy with [80], 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 the rest of this chapter, we will first describe the first, sequential incarnation of our reconstruction pipeline and will subsequently derive its hierarchical evolution. As a final step, we will introduce a clustering strategy derived from the simple linkage that makes the dendrogram more balanced, thereby reducing the *actual complexity* of the method and endow the SaM pipeline with the capability of dealing with uncalibrated images with varying internal parameters and no ancillary information (the actual, complete self-calibration algorithm will be detailed in the chapter 3).

2.2 Sequential Structure and Motion

The initial incarnation of our reconstruction pipeline can be considered nowadays a standard Structure and Motion methodology. We will describe it in detail because its design and components have influenced the subsequent evolution of our approach. Its obsessions for consistency and its radically conservative approach to 3D reconstruction were the solid base over which we built the following hyerarchical pipeline.

In this first algorithm we process a collection of uncalibrated images and output camera parameters, pose estimates and a sparse 3D points cloud of the scene. We do not use external calibration parameters nor EXIF tags embedded into pictures but instead assume the images to have been captured with constant intrinsic parameters. This requirement will be eliminated in the next sections and chapter.

The sequential pipeline follows an incremental greedy approach, similar to [90] and [74]. The most efforts have been made in the direction of a robust and automatic approach, avoiding unnecessary parameters tuning and user intervention. A sample output is shown in Fig. 2.2.

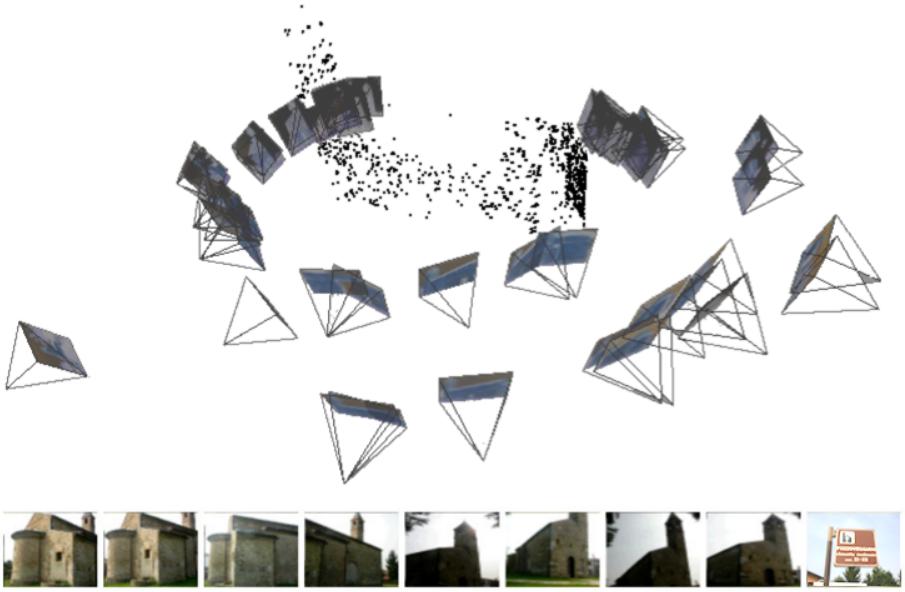

Fig. 2.2. Reconstruction of the "Pozzoveggiani" dataset.

Since the hierarchical pipeline description will be self-consistent, the remainder of this section can be skipped if familiar with standard, sequential Structure and Motion pipelines.

2.2.1 Multimatching

Initially, keypoints are extracted and matched over different images. This is accomplished using SIFT [58] for detection and description of local point features. Matching follows a nearest neighbor approach [58], with rejection of those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than 2.0.

Homographies and fundamental matrices between pairs of images are then computed using RANSAC [31]. At this point we have a set of matches that are considered inliers for a certain model. However, in order to increase the robustness of the method further, we apply an outlier rejection rule, called X84 [40]. Let e_i be the residuals, a robust noise scale estimator is the Median Absolute Deviation (MAD):

$$\sigma^* = 1.4826 \,\mathrm{med}_i \, |e_i - \mathrm{med}_j \, e_j|. \tag{2.1}$$

The robustified inliers [109] are those points such that $e_i < 3.5\sigma^*$. The model parameters are eventually re-estimated via least-squares minimization of the (first-order approximation of) geometric error [42].

The best-fit model (homography or fundamental matrix) is selected according to the Geometric Robust Information Criterion (GRIC) [96]:

$$\text{GRIC} = \sum \rho(e_i^2) + nd\log(r) + k\log(rn) \tag{2.2}$$

$$\rho(e) = \min\left(\frac{e^2}{\sigma^2}, 2(r-d)\right) \tag{2.3}$$

where σ is the standard deviation of the measurement error, k is number of parameters of the model, d is dimension of the fitted manifold, and r is the dimension of the measurements. In our case, $k = 7, d = 3, r = 4$ for fundamental matrices and $k = 8, d = 2, r = 4$ for homographies. The model with the lower GRIC is the more likely.

The final matches are the inliers from the best-fit model. If the number of surviving matches between two images is less than a threshold (25 in our experiments) then they are discarded, and the corresponding homography or fundamental matrix as well.

After that, keypoints matching in multiple images (at least three) are connected into *tracks*, rejecting as inconsistent those tracks in which more than one keypoint converges [90].

2.2.2 Autocalibration

The intrinsic parameters K of the camera are constant but unknown. A globally convergent autocalibration algorithm [33], based on Kruppa equations and the Huang-Faugeras constraint, is employed to recover them automatically from the set of fundamental matrices calculated during the matching phase. In short, the algorithm uses Interval Analysis to minimize the following cost function:

$$\chi(K) = \sum_{i,j} w_{ij} \frac{2\,\text{tr}(E_{ij}E_{ij}^{\mathsf{T}})^2 - \text{tr}^2(E_{ij}E_{ij}^{\mathsf{T}})}{\text{tr}^2(E_{ij}E_{ij}^{\mathsf{T}})} \tag{2.4}$$

where F_{ij} is the fundamental matrix between views i and j, and $E_{ij} = K^{\mathsf{T}}F_{ij}K$.

2.2.3 Initialization

Once the intrinsic parameters are known, the position of each view as well as the 3D location of the tracks is recovered using an incremental approach that entails to start from a seed reconstruction, made up of two calibrated views and the relative 3D points in a Euclidean frame. The extrinsic parameters of two given views is obtained by factorizing the essential matrix, as in [43]. Then 3D points are reconstructed by intersection (via the midpoint algorithm [3]) and pruned using X84 on the reprojection error. Bundle adjustment (BA) [57] is run eventually to improve the reconstruction.

The choice of the two views for initialization turns out to be critical [93]. It should be a compromise between distance of the views and the number of keypoints in common. We require that the matching points must be well spread in the two images, and that the fundamental matrix must explain the data far better than other models (namely, homography), according to the GRIC, as in [74]. This should ensure that the baseline between the two images is large, and that

the fundamental matrix correctly captures the structure of the scene, so that triangulation is well-conditioned and the estimation of the starting 3D structure is reliable. The heuristic adopted in practice is then:

$$\mathcal{S}_{i,j} = \frac{CH_i}{A_i} + \frac{CH_j}{A_j} + \frac{\text{gric}(F_{i,j})}{\text{gric}(H_{i,j})}, \tag{2.5}$$

where CH_i (CH_j) is the area of the convex hull of the keypoint in image I_i (I_j), A_i (A_j) is the total area of image I_i (I_j) and $\text{gric}(F_{i,j})$, $\text{gric}(H_{i,j})$ are the GRIC scores obtained by the fundamental matrix and the homography matrix respectively. The two views with highest $\mathcal{S}_{i,j}$ and with at least 100 matches in common are chosen.

This heuristic will be the base for the following hierarchical evolution of the pipeline, since it will used to guide a agglomerative clustering algorithm on the set of pictures.

Structure and motion pipeline

1. Multimatching:
   a) Extract keypoints in each image;
   b) Match keypoints between each pair of images;
   c) Find the best-fit model using RANSAC and GRIC;
   d) Reject outliers using X84 rule on distance to the best-fit model;
   e) Link keypoints into tracks.
2. Autocalibration, using the fundamental matrices;
3. Initialization:
   a) Select two views according to (2.5);
   b) Compute their extrinsic parameters via factorization of essential matrix.
4. Incremental Step Loop:
   a) Compute 3D points with intersection and run X84 on the reprojection error;
   b) Add new 3D points to the reconstruction;
   c) Run BA on the current reconstruction;
   d) Select the next view;
   e) Initialise camera pose with RANSAC and linear exterior orientation;
   f) Add the camera to the reconstruction;
   g) Run BA on the current reconstruction;
   h) Select new tracks;

2.2.4 Incremental Step Loop

After initialization, a new view at a time is added until there are no remaining views. The next view to be considered is the one that contains the largest number of tracks whose 3D position has already been estimated. This gives the maximum number of 3D-2D correspondences, that are exploited to solve an exterior orientation problem via a linear algorithm [30]. The algorithm is used inside a RANSAC iteration, in order to cope with outliers. The extrinsic parameters are then refined with BA.

Afterwards, the 3D structure is updated by adding new tracks, if possible. Candidates are those tracks that have been seen in at least one of the cameras in the current reconstruction. 3D points are reconstructed by intersection (midpoint algorithm), and successively pruned using 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.

Finally, we run BA again, including the new 3D points. If BA, at any stage, does not converge, then the view is rejected.

2.3 Hierarchical Structure and Motion

We will proceed describing a first hierarchical adaption, which we will then finally further optimize to obtain better average complexity and to support our novel self-calibration approach described in chapter 3.

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, is independent from the initial pair of views, and copes better with drift problems, typical of sequential schemes. The global complexity is trimmed further by limiting the number of views employed per node, with the introduction of a local bundle adjustment strategy.

The initial part of the pipeline, regarding keypoint and view matching, while substantially similar to the corresponding parts of the sequential approach described in the previous section, have been updated when developing this new incarnation of the Structure and Motion pipeline. They will be described in their entirety to make this section self-consistent.

2.3.1 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 [7]. SIFT [58] 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 [65]). 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 [58], 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 [97]. Let e_i be the residuals after MSAC, following [109], the final set of inliers are those points such that

$$|e_i - \text{med}_j\, e_j| < 3.5\sigma^*, \tag{2.6}$$

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

$$\sigma^* = 1.4826\, \text{med}_i\, |e_i - \text{med}_j\, e_j|. \tag{2.7}$$

This outlier rejection rule is called X84 in [40].

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 [13, 59].

The more likely model (homography or fundamental matrix) is selected according to the Geometric Robust Information Criterion (GRIC) [96].

$$\begin{aligned} \text{GRIC} &= \sum \rho(e_i^2) + nd\log(r) + k\log(rn) \\ \rho(e) &= \min\left(e/\sigma^2, 2(r-d)\right) \end{aligned} \tag{2.8}$$

where σ is the standard deviation of the measurement error, k is number of parameters of the model, d is dimension of the fitted manifold, and r is the dimension of the measurements. In our case, $k = 7, d = 3, r = 4$ for fundamental matrices and $k = 8, d = 2, r = 4$ for homographies. The model with the lower GRIC is the more likely.

Finally, if the number of remaining matches between two images is less than a threshold (computed basing on a statistical test as in [7]) 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 [90] and those shorter than three frames.

2.3.2 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 [80], panoramas [7], image mining [78] and scene summarization [88]. The distance being used and the clustering algorithm are application-specific.

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:

$$a_{i,j} = \frac{1}{2}\frac{|S_i \cap S_j|}{|S_i \cup S_j|} + \frac{1}{2}\frac{CH(S_i) + CH(S_j)}{A_i + A_j} \tag{2.9}$$

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 [23]. 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 (2.9). 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 [77].

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

$$\text{gric}(F_{i,j}) < \alpha\, \text{gric}(H_{i,j}) \quad \text{with } \alpha \geq 1, \tag{2.10}$$

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.

2.3.3 Hierarchical Reconstruction

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 [33].

The extrinsic parameters of two given views are obtained by factorizing the essential matrix, as in [43]. 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 [30], or resection with DLT [42] 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.

2.3.4 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 $O(n^4)$.

The cost of bundle adjustment with m points and n views is $O(mn(m+2n)^2)$ [87], 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

$$\sum_{i=1}^{n} O(i^4) = O(n^5). \tag{2.11}$$

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:

$$T(n) = 2T(n/2) + O(n^4) \tag{2.12}$$

that is $T(n) = O(n^4)$ by the third branch of the Master's theorem [16].

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.

2.3.5 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 [66,107], 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. 2.3.2, to be the active views.

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

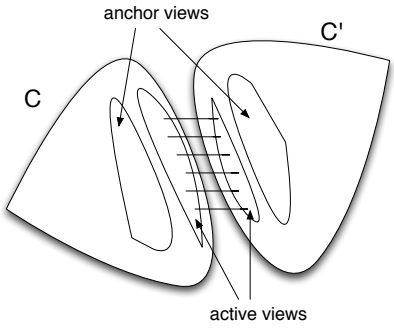

Fig. 2.3. Local bundle adjustment. The active views are the k closest pairs between the two clusters with 3D points in common. They will be moved by bundle adjustment. The anchor views are the k closest views to the active ones inside each cluster. They contribute to the reprojection error, but are not 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.2.3 illustrates this idea.

The points involved in BA are the ones that are reconstructable from $\mathcal{P}(C) \cup A$, let us call them P. Let P' be the same set after BA. In order to approximately propagate the trasformation undergone by P, we compute the least squares affinity that bring P onto P' and apply it to the remaining points. The remaining views of C (and possibly C') are adjusted by resection with minimization of the reprojection error.

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.

2.3.6 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.

2.3.7 Evaluation of the hyerarchial framework

Fig. 2.4. Two perspective views of the reconstruction of "Piazza Erbe" (Verona, Italy).

Fig. 2.5. Two perspective views of the reconstruction of "Piazza Bra" with the Arena (Verona, Italy).

We tested our algorithm (henceforth called SAMANTHA) on several datasets of pictures taken by the authors with a consumer camera with known internal parameters. Figure 2.4 and 2.5 illustrates the reconstruction from the "Piazza Erbe" and "Piazza Bra" datasets, respectively.

We compared our results with those produced by BUNDLER [10], a sequential SaM pipeline implemented in C++ and still considered part of the state of the art. Inside our pipeline we used the C++ implementation of bundle adjustment (BA) described in [57]. 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 2.1 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.

		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 2.1. 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.

As an example, Figure 2.6 and 2.7 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.

Fig. 2.6. Top views aligned with an aerial image of "Piazza Erbe" (from Google Earth), reconstructed with SAMANTHA (left) and with BUNDLER (right).

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. 2.7). A similar result occurs for the "Tribuna" dataset.

In Table 2.2 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 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

Fig. 2.7. Top views aligned with an aerial image of "Piazza Bra" (from Google Earth), reconstructed with SAMANTHA (left) and with BUNDLER (right).

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.

# 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 2.2. 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.

For a qualitative comparison, in Figure 2.8 we registered two top views of the final structure obtained with and without local BA (we used 15 active views).

2.4 Dendrogram balancing and self-calibration

In this section we describe our latest revision of out reconstruction pipeline. As demonstrated in precedence, the hierarchical framework can provide a provable computational gain, *provided that the resulting tree is well-balanced*.

The worst case complexity, corresponding to a sequence of single view additions, is no better than the standard sequential approach. It is therefore crucial

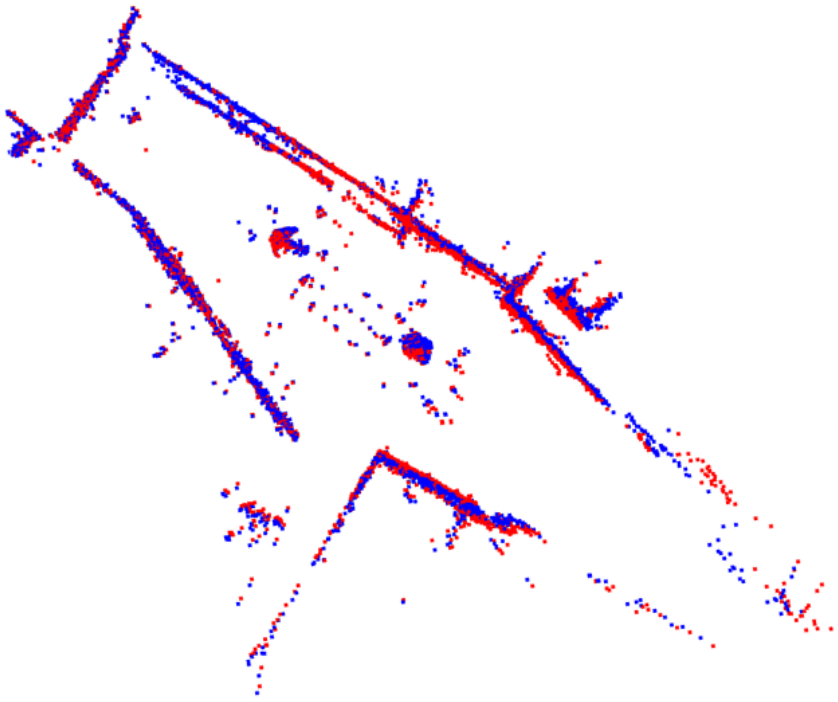

Fig. 2.8. Comparison between the result obtained by SAMANTHA with (in red) and without (in black) local BA.

to ensure a good balance during the clustering phase. Our solution is to employ a novel clustering procedure, which instead of following a completely greedy strategy promotes the creation of better balanced dendrograms.

In this section we will also integrate in our approach a variant of the self-calibration algorithm which is the topic of the next chapter. To our knowledge, our solution was the first published capable of dealing with variable internal parameters without using ancillary information, such as EXIF tags embedded in some image formats.

2.4.1 Balanced view clustering

The view clustering procedure proposed in the previous section allows us to organize the available views into a hierarchical cluster structure (a tree) that will guide the reconstruction process. This approach decreases the computational complexity with respect to sequential SaM pipelines, 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 gains vanishes. It is therefore crucial to enforce the balancing of the tree.

We use the same view affinity heuristic that was employed in the previous section, but modify the linking strategy. The preceding solution, which used the

simple rule, specified 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. 2.4.4) that a good compromise is $\ell = 5$.

Figure 2.9 shows an example of balancing achieved by our technique. The height of the tree is reduced from 14 to 9 and more initial pairs are present in the dendrogram on the right.

2.4.2 Uncalibrated Hierarchical Reconstruction

The main difference from the previous hierarchical approach is that now leaf nodes do not have proper calibration right from the start of the reconstruction process.

The reconstruction starts uncalibrated, and as soon as an uncalibrated cluster reaches a given dimension m, the Euclidean upgrade procedure is triggered (in principle autocalibration with known skew and aspect ratio requires a minimum of $m = 4$ views to work; for good measure we used $m = 12$). 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.

Each step of hierarchical reconstruction must therefore be modified to accomodate for clusters not yet in a euclidean reference frame.

Two-views reconstruction.

The reconstruction from two views is always projective in this pipeline (autocalibration is triggered for clusters larger than m). However, we strive to mantain even for clusters smaller than m a quasi-euclidean reference frame, for numerical stability and to better condition the following autocalibration step.

We propose to compute the best plane at infinity compatible with rough focal estimates obtained from the magnitude of the image diagonal. Even when the true focal lengths are far from the estimates, this procedure will provide a useful, well conditioned starting point for the subsequent reconstruction steps.

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{2.13}$$

$$P_2^E = K_2 \left[R_2 | \mathbf{t}_2\right] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top | \lambda \mathbf{q}_2\right] \tag{2.14}$$

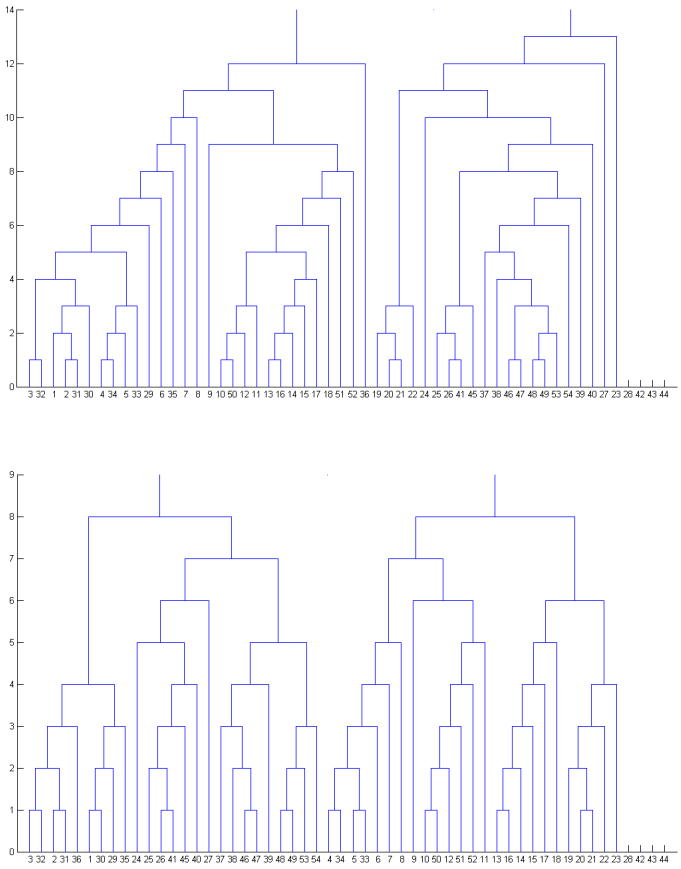

Fig. 2.9. An example of the dendrogram produced by simple linkage (left) and the balanced rule on a 52-views set.

$$R_2 \simeq K_2^{-1}\left(Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top\right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{2.15}$$

The procedure will be described in detail and further refined in 3.2.1. It consists however in identifying the best $\mathbf{v}$ that will render the right hand side of equation 2.15 equal up to a scale to a rotation. Please note however that the self calibration algorithm proposed there is novel; the autocalibration procedure used in this chapter is instead based on equations derived from the dual image of the absolute quadric.

This initialization can then be further refined enforcing cheirality and any other constraints on the internal parameters by non-linear minimization. Given approximate P_1^E and P_2^E, the position in space of the points is then obtained by triangulation (or intersection). Outliers are pruned by analyzing the reprojection error. Projective 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 glue the view to the cluster. This can be done by linear exterior orientation [30] or by resection with DLT [42], depending on whether the cluster corresponds to a Euclidean or projective reconstruction (a single view is always uncalibrated). MSAC [97] 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 using the iterated linear LS method [44], and pruned by analyzing the reprojection error. Bundle adjustment is run on the current reconstruction (either Euclidean or projective).

Clusters merging.

When two cluster merges 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 re-computed by intersection, with the same cautions as before, namely analysis of the reprojection error and test on the conditioning number. Tracks obtained after the merging are also triangulated. The new reconstruction is refined with bundle adjustment (either Euclidean or projective) and upgraded to a Euclidean frame when the conditions stated beforehand are met.

2.4.3 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 (DIAC).

The autocalibration method we use here comes from the merge of [84] and [77].

Our implementation of the iterative dual linear self-calibration algorithm is based on the method described in [85], modified to use the weights of [77] and to enforce at every iteration the positive (negative) semi-definitess of the dual absolute quadric.

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 3 approximation step of the original algorithm.

Formal tests, not reported here, demostrated this algorithm to have better convergence properties of both its parents [77, 85]. This is not sufficient for the method to consistently converge to a reasonable solution.

The quasi-euclidean upgrade step summarily described in subsection 2.4.2 is crucial to this effect, providing a good enough approximation of the plane at infinity which usually guarantees the convegence of methods based on DIAQ constraints.

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 $k = 25$).

2.4.4 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 2.10 and 2.11 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.

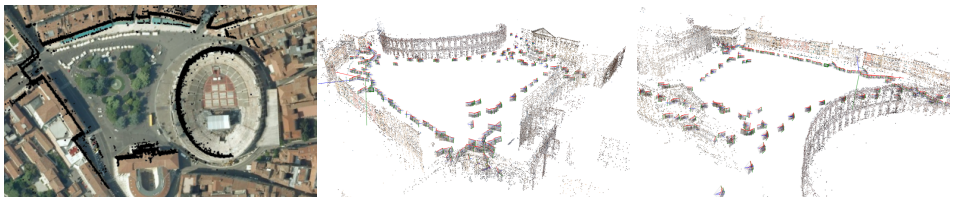

Fig. 2.10. A top view and two perspective views of the reconstruction of "Piazza Bra" (Verona, Italy).

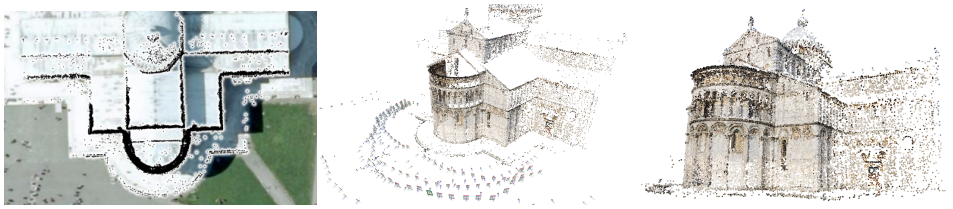

Fig. 2.11. A top view and two perspective views of the reconstruction of "Duomo" (Pisa, Italy).

Time efficiency

We compared our results again with those produced by BUNDLER [10] (an implementation of a state-of-the-art sequential SaM pipeline in C++). Our previous pipeline, described in section 1.3, could not be used because unable to process datasets with different camera intrinsics.

Table 2.3 reports the result of the comparison with BUNDLER. Only time spent doing BA (C++ implementation of [57]) 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 differences in terms of number of reconstructed views and points. The total speed up achieved with respect to bundler is 14 and 4.8 for "Piazza Bra" and "Duomo" respectively, which compares favorably with the speed-up reported in [89] (on different dataset, though).

		BUNDLER			SAMANTHA		
Dataset	# img	# views	# points	time	#views	#points	time
Piazza Bra	411	292	41703	12:16 h	335	55598	52 min
Pisa	309	309	105401	13:43 h	309	121047	2:57 h

Table 2.3. 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).

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. 2.12, where the number of reconstructed points/views and the computing time are plotted as the number of closest pairs ℓ is increased. After $\ell = 5$ the computing time stabilizes at around 30% of the baseline case, without any significant differences in terms of number of reconstructed views and points.

As theory prescribes, the computing time is directly linked to the height of the tree.

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. The residual distances between closest pairs are measured and their average – the reconstruction accuracy – is about 35cm for "Piazza Bra" and 15cm for "Duomo".

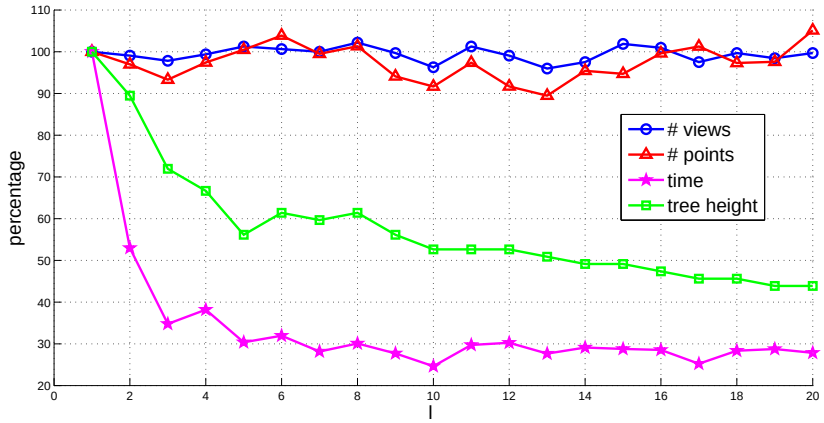

Fig. 2.12. This plot shows the number of reconstructed points, views, height of the tree and computing time as a function of the parameter ℓ in the balancing heuristics. The values on the ordinate are in percentage with respect to the baseline case $\ell = 1$ which correspond to the original simple-linkage clustering of section 1.3.

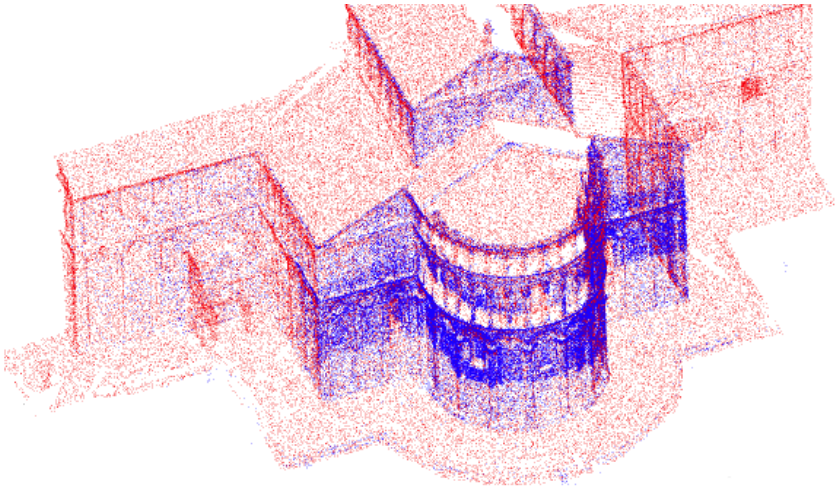

Fig. 2.13. A view of "Duomo" reconstruction (blue) superimposed to the ground truth (red).

The final error of BUNDLER on the same datasets is 17cm for "Duomo", whereas for "Piazza Bra" BUNDLER failed to produce a meaningful result.

3

Practical Self-calibration

As it has been noted several times in literature, the difficult part of autocalibration efforts resides in the structural non-linearity of the search for the plane at infinity.

In this chapter we show how to easily compute it from a estimate of the intrinsic parameters of at least two uncalibrated cameras. The procedure is leveraged to build an autocalibration algorithm which is both robust and versatile.

The method works by enumerating through the inherently bounded space of internal camera parameters in order to find the best rectifying homography.

We compare our approach with several other algorithms on both synthetic and concrete cases, obtaining favourable results.

3.1 Introduction

Autocalibration (a.k.a. self-calibration) has generated a lot of theoretical interest since its introduction in the seminal paper by Maybank and Faugeras [62]. The attention spawned by the problem however is inherently practical, since it eliminates the need for off-line calibration and enables the use of content acquired in an uncontrolled environment. Modern computer vision has partly sidestepped the issue using ancillary information, such as EXIF tags embedded in some image formats. Such data unfortunately is not always guaranteed to be present or consistent with its medium, and does not extinguish the need for reliable autocalibration procedures.

Lots of published methods rely on equations involving the dual image of the absolute quadric, introduced by Triggs in [98]. Earliest approaches for variable focal lenghts were based on linear, weighted systems [75, 77], solved directly or iteratively [85]. Their reliability were improved by more recent algorithms, such as [11], solving super-linear systems while forcing directly the positive definiteness of the DIAQ. Such enhancements were necessary because of the structural non-linearity of the task: for this reason the problem has also been approached using branch and bound schemes, based either on the Kruppa equations [33], dual linear autocalibration [5] or the modulus constraint [12].

The algorithm described in [41] shares with the branch and bound approaches the guarantee of convergence; the non-linear part, corresponding to the localization

of the plane at infinity Π_∞, is solved exaustively after having used the cheiral inequalities to compute explicit bounds on Π_∞.

The technique we are about to describe is closely related to the latter: first, we derive the location of the plane at infinity given two perspective projection matrices and a guess on their intrinsic parameters, and subsequently use this procedure to iterate through the space of camera intrinsic parameters looking for the best rectifying collineation. The search space is inherently bounded by the finiteness of the acquisition devices; each sample and the corresponding plane at infinity defines a collineation of space whose likelyhood can be computed evaluating skew, aspect ratio, principal point and related constraints for each transformed camera. The best solution is eventually refined via non-linear least squares.

Such approach has several advantages: it's fast, easy to implement and reliable, since a reasonable solution can always be found in non-degenerate configurations, even in extreme cases such as when upgrading just two cameras.

3.2 Method

As customary, we assume being given a projective reconstruction $\{P_i; X_j\}$ $i = i \dots n$; $j = 1 \dots m$. The purpose of autocalibration is therefore to find the collineation of space H such as $\left\{P_i H; H^{-1} X_j\right\}$ is a Euclidean reconstruction, i.e., it differs from the true one by a similarity.

The set of camera matrices can always be transformed to the following canonical form by post-multiplying each P_i by the matrix $[P_1;\ 0\ 0\ 0\ 1]^{-1}$.

$$P_1 = [I \mid \mathbf{0}] \qquad P_i = [Q_i \mid \mathbf{q}_i] \tag{3.1}$$

In this situation, the rectifying homography H performing the euclidean upgrade has the following structure:

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & \lambda \end{bmatrix} \tag{3.2}$$

where K_1 is the calibration matrix of the first camera, $\mathbf{v}$ a vector which determines the location of the plane at infinity and λ a scalar fixating the overall scale of the reconstruction.

The technique we are about to describe is based on two stages:

1. Given a guess on the internal parameters of two cameras compute a consistent rectifying homography. This yields an estimate of all but the first camera.
2. Score the internal parameters of these $n - 1$ cameras based on the likelyhood of skew, aspect ratio and principal point.

The space of the intrinsic parameters of the two cameras is enumerated and the best solution is eventually refined via non-linear least squares.

3.2.1 Estimation of the plane at infinity

In this section we will show how to compute the plane at infinity given two perspective projection matrices and their intrinsic parameters. This procedure is, in

a sense, the dual of the second step of the stratified autocalibration [29] in which the internal parameters are recovered given the plane at infinity. This problem has been dealt with for the first time in [6] where it has been turned into a linear least squares system. We shall derive here a closed form solution.

Given two projective cameras

$$P_1 = [I \mid \mathbf{0}] \qquad P_2 = [Q_2 \mid \mathbf{q}_2] \tag{3.3}$$

and their intrinsic parameters matrices K_1 and K_2 respectively, the upgraded, Euclidean versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{3.4}$$

$$P_2^E = K_2 [R_2 | \mathbf{t}_2] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top | \lambda \mathbf{q}_2 \right] \tag{3.5}$$

with the symbol $\simeq$ meaning "equality up to a scale". The rotation R_2 can therefore be equated to the following:

$$R_2 \simeq K_2^{-1} \left(Q_2 K_1 + \mathbf{q}_2 \mathbf{v}^\top \right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{3.6}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term.

Using the constraints on orthogonality between rows or columns of a rotation matrix, one can solve for $\mathbf{v}$ finding the value that makes the right hand side of Eq. 3.6 equal up to a scale to a rotation. The solution can be obtained in closed form by noting that there always exists a rotation matrix R^* such as: $R^* \mathbf{t}_2 = [\|\mathbf{t}_2\| \; 0 \; 0]^\top$. Left multiplying it to Eq. 3.6 yields:

$$R^* R_2 \simeq \overbrace{R^* \; K_2^{-1} Q_2 K_1}^{W} + [\|\mathbf{t}_2\| \; 0 \; 0]^\top \mathbf{v}^\top \tag{3.7}$$

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

$$R^* \; R_2 = \begin{bmatrix} \mathbf{w}_1{}^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2{}^\top \\ \mathbf{w}_3{}^\top \end{bmatrix} / \|\mathbf{w}_3\| \tag{3.8}$$

in which the last two rows of the right hand side are independent from the value of $\mathbf{v}$ and the correct scale has been recovered normalizing to unit norm each side of the equation.

Since the rows of the right hand side form a orthonormal basis, we can recover the first one taking the cross product of the other two. Vector $\mathbf{v}$ is therefore equal to:

$$\mathbf{v} = \left(\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1 \right) / \|\mathbf{t}_i\| \tag{3.9}$$

The upgrading homography can be computed using Eq. 3.2; the term λ can be arbitrarily chosen, as it will just influence the overall scale of the reconstruction. Its sign however will affect the cheirality of the reconstruction, so it must be chosen positive if cheirality was previously adjusted.

3.2.2 Estimation of the interal parameters

In the preceding section we showed how to compute the location of the plane at infinity given the calibration parameters of two of the cameras of the projective reconstruction to upgrade.

The autocalibration algorithm we propose consists in enumerating through all possible matrices of intrinsics of two cameras checking whether the entire resulting reconstruction has the desired properties.

The process is well-defined, since the search space is naturally bounded by the finiteness of the acquisition devices; in practice, we use the same realistic expectations that were used in [77] to compute the linear system weights.

To score each sampled point, we test the aspect ratio, skew and principal point location of the resulting transformed projection matrices and aggregate their respective value into a single cost function. The actual form of the cost function that can be used depends from the number of cameras since the counting argument [60] still applies. In general, we found a simple absolute summation of the chosen properties weighed as in [77] to give good results.

$$\{K_1, K_2\} = \arg\min_{K_1,K_2} \sum_{\ell=2}^{n} f(K_\ell) \tag{3.10}$$

$$f(K) = w_{sk}|k_{1,2}| + w_{ar}|k_{1,1} - k_{2,2}| + w_{u_o}|k_{1,3}| + w_{v_o}|k_{2,3}| \tag{3.11}$$

where $k_{i,j}$ denotes the entry (i, j) of K and w are suitable weights.

The last open problem is how to properly sample the space of calibration parameters. We can safely assume, as usual, null skew and unit aspect ratio: this leaves the focal length and the principal point location as free parameters. However, as expected, the value of the plane at infinity is in general far more sensitive to errors in the estimation of focal lenght values rather than the image center. Thus, we can iterate just over focal lenghts assuming the principal point to be centered on the image; the error introduced with this approximation is normally well-within the radius of convergence of the subsequent non-linear optimization.

Figure 3.1 shows various cost profiles for each of the term of Eq. 3.11, obtained with the aforementioned method. As it can be seen, the cost profiles have very clear valleys and globally concur to identify the correct solution, displayed in the graphs as an asterisk.

What happens when the trial estimates of K_1 and K_2 are not correct? In that case, we are not guaranteed anymore that the right hand side of Eq. 3.8 will be a rotation matrix. w_2 and w_3 will not be mutually orthogonal, nor have equal, unit norm. Such errors will be transferred to the internal parameters of all cameras but the first after upgrade with the resulting homography. Equation 3.9 will still yield the value of $\mathbf{v}$ that makes the right hand side of Eq. 3.8 closest to a rotation in Frobenius norm.

As each row of Fig. 3.1 shows, the cost profiles from just a single camera can still identify a unambigous minima. This situation is equivalent to the task of identifying the focal lengths of two cameras from their fundamental matrix. This problem, studied extensively in [68, 92], was demonstrated to be essentially ill-conditioned. Our approach is stabler since it structurally requires the solution to

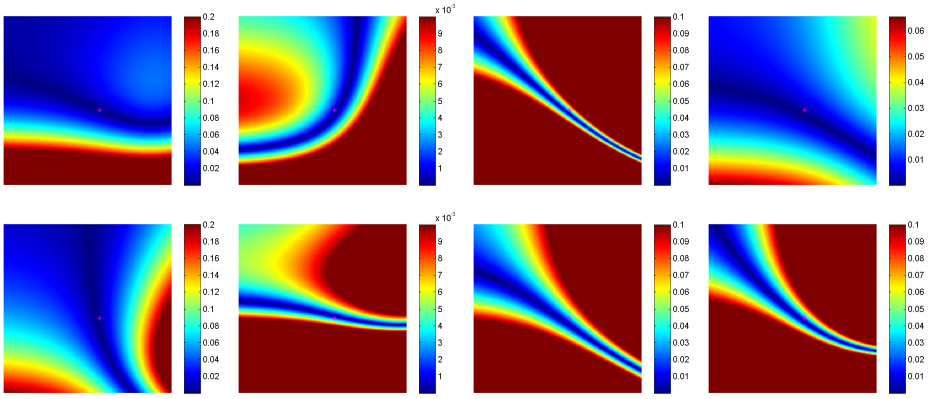

Fig. 3.1. Cost functions. The two rows refer to cost functions relative to different cameras of a same dataset. From left to right, are shown the profiles of aspect ratio, skew and principal point x and y coordinates as function of the focal lengths of the reference cameras. Cooler colors correspond to lower, better values of the objective function. A asterisk marks the correct solution.

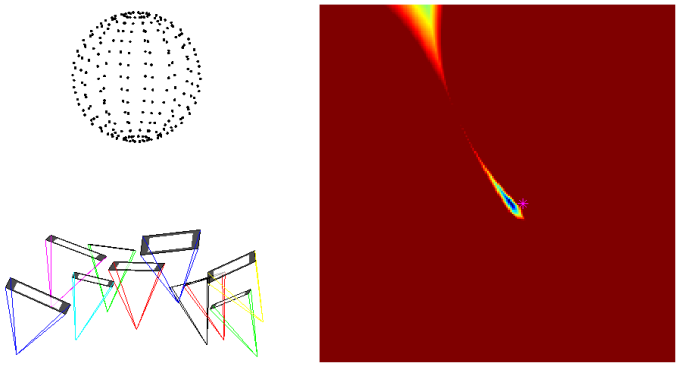

Fig. 3.2. A ten camera synthetic reconstruction and the resulting aggregated cost function. An asterisk marks the correct solution.

be in a valid region of the parameter space. The solution clearly improves as more and more cameras are added. Figure 3.2 shows the aggregated cost for the ten camera synthetic dataset.

Finally, the solution selected is refined by non-linear minimization; since it is usually very close to a minima, just a few iterations of a Levemberg-Marquardt [56] solver are necessary for convergence. The employed cost function is the same reported in Eq. 3.10.

The entire procedure is presented as pseudo-code in algorithm 1. With the perspective projection matrices the code presented takes as input also the viewport matrices of the cameras, defined as per Eq. 3.12 where w and h are respectively the width and height of each image. While this proposed normalization is not mandatory, we recommend it to improve the numerical properties of the algorithm.

Algorithm 1: Autocalibration pseudo-code

input : a set of PPMs P and their viewports V
output: their upgraded, euclidean counterparts

foreach P **do** $P \leftarrow V^{-1}P/\|P_{3,1:3}\|$ /* normalization */
foreach K_1, K_2 **do** /* iterate over focal pairs */
  compute Π_∞
  build H *from eq. 3.2*
  foreach P **do** /* compute cost profiles */
    $P_E \leftarrow PH$
    $K \leftarrow$ *intrinsics of* P_E
    compute $f(K)$ *from eq. 3.11*
  end
end
aggregate cost and select minimum
refine non-linearly
foreach P **do** $P \leftarrow VPH$ /* de-normalization, upgrade */

$$V_{w,h} = \begin{bmatrix} \sqrt{w^2+h^2} & 0 & w \\ 0 & \sqrt{w^2+h^2} & h \\ 0 & 0 & 2 \end{bmatrix} /2 \tag{3.12}$$

The proposed algorithm shows remarkable convergence properties; we observed it fail only when the sampling of the focal space was not sufficiently dense, and therefore a value for the plane at infinity close to the correct one was not generated. Such problems are easy to detect, since they usually bring the final, refined solution outside the legal parameter space.

3.3 Experimental evaluation

We report here several tests on synthetic and concrete datasets. For the experiments, unless otherwise specified, we sampled the focal space using 20 logarithmically spaced divisions in the range $[0.3 \dots 3]$.

3.3.1 Synthetic tests

For this series of tests, we generated several synthetic reconstructions constructed with ten or more camera looking at the unit sphere. Each camera was chosen having different parameters except for skew, which was set equal to zero for all perspective projection matrices. The other characteristics were selected by a random process inside the valid parameter space. The virtual viewport size for each camera was $[1024, 768]$ units, leading to focal lengths and principal points coordinates of comparable magnitude. We built projectively equivalent reconstructions multiplying the euclidian frame for a random homography.

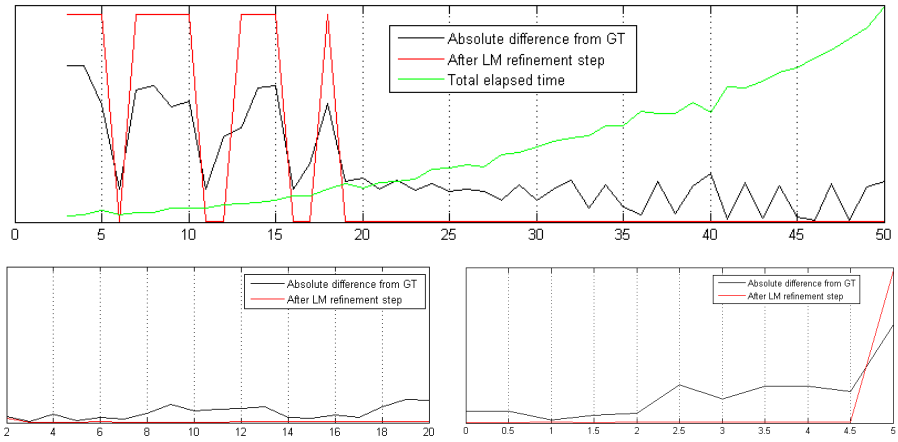

Fig. 3.3. Synthetic tests. Top: the residual from the ground truth as a function of the sampling divisions. Bottom left: stability as the number of cameras varies. Bottom right: resiliance to noise.

Sampling rate.

The top graph of figure 3.3 shows the relationship between the sampling rate used in the focal search phase, running along the x axis, and the accuracy of the resulting self-calibration, expressed as the sum of the Frobenius norms of the differences between the estimated and correct calibration parameters. For too low rates of sampling, corresponding to the left side of the diagram, the chance of picking a solution close to the correct one is very low. Most of the time the subsequent LM minimization outputs parameters outside the valid range, generally converging towards the trivial null focal solution. As soon as the focal lengths are sampled with a sufficient degree of accuracy, the residual of the recovered solution becomes and stay low. When this happens, the proposed solution is usually very near to the correct one, and the following non-linear minimization has no problem to converge to the correct, best calibration parameters.

The sawtooth pattern that can be noted both at the start and end of the sequence is a sampling artifact which depends on the distance between the current estimate and the correct one. At the far right of the sequence, it roughly correlates to odd and even divisions of the sampling space.

The total elapsed time, shown in green, follows a quadratic law, as expected. At the far right of the diagram, corresponding to fifty divisions for each focal, the total time spent (search plus refinement) is roughly 3 seconds, implemented as a Matlab script.

Number of cameras.

In this section we verify the stability of the algorithm as the number of cameras varies from two to twenty. For uniformity all reported results were obtained with the full cost function described in equation 3.11, even for experiments which, having a sufficient number of cameras, could use fewer constraints. Results reported in

the bottom left graph of figure 3.3 are averaged over 50 runs of the algorithm. As shown, the algorithm is able to converge to the correct calibration parameters for all but the two cameras setup, in which it gets caught in a local minima still very close to the ground truth. From three cameras onwards the method successfully disambiguates the uncertainty.

Noise resiliance.

Our final synthetic test verifies the resiliance to noise; several reconstructions were built from the ground truth perturbing the point projections with gaussian noise and recovering each camera by DLT based resection. The last panel of figure 3.3 shows the dependency of the residual from the ground truth in relationship with the standard deviation of the added noise for a ten camera dataset. Again, the results were averaged over 50 runs of the algorithm. As it can be seen the method is fairly stable, starting to fail for standard deviation higher than five unit with respect to a 1024×768 picture frame. This is not surprising given the deterioration that was observed under this conditions on the cameras returned by DLT resection, with focal lengths differing more than two hundred unit from the ground truth.

3.3.2 Comparative tests

We compare our approach to a classical, linear tecnique based on the DIAQ constraints and a recent stratified method based on least squares minimization of the modulus constraint embedded in a branch and bound framework.

The first algorithm is our implementation of the iterative dual linear self-calibration algorithm described in [85], modified to use the weights of [77] and to enforce at every iteration the positive (negative) semi-definitess of the dual absolute quadric. 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 3 approximation step of the original algorithm. Formal tests, not reported here for brevity, demostrated this algorithm to have better convergence properties of both its parents [77, 85]. We report also the results obtained by this method when coupled with the preliminary quasi-affine upgrade step detailed in [45].

The second method we compare to is the algorithm described in [12], a stratified self-calibration approach based on a branch and bound framework using convex relaxations minimizations. We used the reference implementation from the authors of the paper.

The synthetic test dataset, also coming from [12], is composed of twenty projective cameras and points, with known ground truth and gaussian noise of standard deviation σ added to image coordinates. We report results obtained by our and the aforementioned methods over a hundred trials in the case of $\sigma = 0.1\%$ using the same metric defined in the original article.

$$\Delta f = |\frac{f_1 + f_2}{f_1^0 + f_2^0} - 1| \tag{3.13}$$

Algorithm	Cameras	Δf	Success rate
Dual linear	5	5.4012e-2	57
	10	2.6522e-3	84
	20	1.5433e-3	90
DL + QA upgrade	5	2.7420e-2	63
	10	1.8943e-3	83
	20	1.1295e-3	92
Chandraker *et al* [12]	5	9.9611e-3	100
	10	4.7925e-3	100
	20	1.0461e-3	100
Our method	5	2.7546e-3	100
	10	1.3005e-3	100
	20	8.2266e-4	100

Table 3.1. Comparison of results obtained on the dataset from [11].

Results are reported in table 3.1. The linear algorithm, which we pick as baseline case, achieves good results in terms of metric 3.13 but shows poor convergence properties, especially for lower number of cameras. Similar numerical results are unsurprisingly obtained coupling the method with the quasi-affine upgrade of [45], with slightly higher percentuals of success. Both the algorithm described in [12] and our method never failed on this dataset, with a slight numerical advantage of our proposal.

3.3.3 Real world example

We finally test our algorithm on two concrete dataset, respectively the "Pozzoveggiani" and "Duomo" reconstruction produced by our hyerarchical pipeline. Each set is composed respectively of 52 and 333 cameras.

The euclidean reconstructions, as refined through bundle adjustment, were used as ground truth for the subsequent tests. Again, a total of a hundred trials were conducted for each set, multiplying the projective reconstructions for a random homography while discarding the ones with very low condition number. In our method we also picked at random the reference views to be used for the estimation of the plane at infinity.

Results are reported in table 3.2. With respect to the synthetic case, we can note a substantial decrease of the success rate of both linear algorithms which was instead expected to increase with the number of cameras. An informal audit of the code showed the effect to be caused both by noise and by the larger number of iterations required for convergence, which in turn increase the chance of encountering a failure case.

Algorithm [12] is missing from table 3.2 because we were not able to obtain acceptable solutions on our datasets; we tried, to no avail, varying the tolerance ϵ and the maximal number of iterations for both the affine and metric upgrade steps.

Our approach achieves on both datasets flawless success rate. Instances of each upgraded reconstruction can be qualitatively evaluated in figure 3.4.

Algorithm	Pozzoveggiani (55 cameras)		Duomo (333 cameras)	
	Δf	Success rate	Δf	Success rate
Dual linear	3.0815e-2	19	9.3255e-2	8
DL + QA upgrade	8.9261e-3	22	7.6403e-2	13
Our method	3.9733e-3	100	2.9293e-3	100

Table 3.2. Comparison of results obtained on concrete reconstructions.

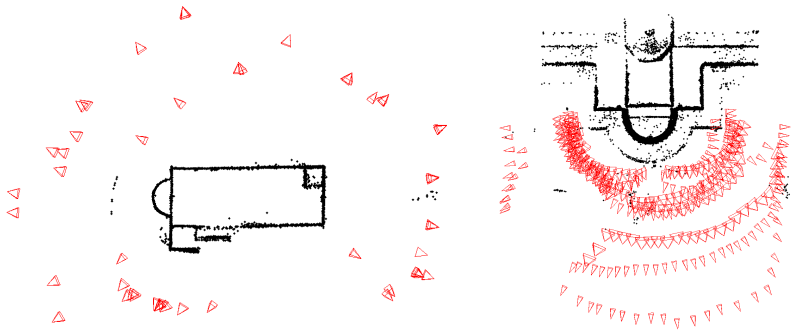

Fig. 3.4. Concrete tests. Two instances of the results obtained on the "Pozzoveggiani" and "Duomo" datasets. As it can be seen, perpendicular features are correctly reproduced. The skew of each camera in normalized coordinates is always less than 10^{-2}.

3.4 Final remarks

We presented a practical self-calibration algorithm showing results at least comparable to the state of the art. Our approach is fast, easy to implement and can successfully disambiguate fringe cases previously considered bad conditioned.

The significance of this method is twofold. The first important point is its ability to work, as demonstrated in our last batch of experiments, with modern Structure and Motion datasets, making it possible to analyze pictures without requiring EXIF information to be always present. The second point is the possibility of upgrading single couples of cameras. This result is what enables us to use it in our hierarchical reconstruction framework, and to produce metric point clouds from the very bottom of the reconstruction tree.

Future research will be aimed at developing sub-linear search strategies in the space of calibration parameters, which is made possible by the structure of the cost profiles.

Part II

Visualization

4

Confidence-based stereo matching

With this chapter, we start to tackle the problem of how to transform the point cloud recovered in the preceding part of this thesis into something more than a collection of points and cameras in space.

If the sought goal is to obtain a reconstruction more faithful to the original scene, a straighforward solution is to employ a stereo matching algorithm to densify the reconstructed point cloud and eventually to organize it into dense triangulated surfaces.

The correspondances recovered during the Structure and Motion computation however can be used to obtain reliable disparities for each reference couple used for stereo matching; in this chapter we develop a way to incorporate such high confidence information into any stereo algorithm.

We do this defining a novel operator to be applied at raw matching costs. It aims at improving matching reliability by efficiently modulating pixel-wise pairing costs, injecting a confidence backed bias before the aggregation step. It works analyzing a noisy estimate of the correspondances in order to favor or prune potential matches.

We will test the operator by developing a local, realtime stereo matching algorithm and showing that our solution can drastically clean the resulting depth map while also reducing border bleeding. Its good performance is also evaluated quantitavely by testing the algorithm against the popular Middlebury benchmark where our local greedy implementation is able to obtain results comparable to those of näive global approaches.

Finally, we will use it to obtain to cluster the points of a reconstruction of the Valbonne church (made famous by the self-calibration literature) in order to obtain rough, connected 3D model.

4.1 Introduction

In this chapther we describe a pixel-wise operator aimed at refining and improving the reliability of a underlying matching cost in the context of low level vision. The current trend for tasks like stereo matching and optical flow computation has been an ever-increasing sofistication, exacerbated and fueled by the publication of common dataset and benchmarks [2, 81]. The top performers in each category

are often composed of several complex modules like plane fitting, edge-preserving smoothing, image segmentation and many others.

It's easy to see that any improvement in the earliest step of the matching computation, namely in the calculation of the first matching cost, can have profound and beneficial effects on the remainder of a algorithm pipeline.

We propose a simple and efficient operator capable of drastically pruning potential correspondances for a pixel. It works analyzing (or in a sense, refining) a noisy initial approximation of the depth or flow map, smoothly inhibiting matching pairs without sufficient support in a local, unstructured neighbourhood.

To support our claim that incorporating such operator into existing algorithms could provide additional reliability while allowing a simplification in the regularization tecniques, we implement a local, greedy stereo matching implementation whose results are comparable to näive global approaches at a fraction of the sofistication and time complexity.

At the end of this chapter, we will use the stereo results in order to obtain the connectivity information in a point cloud coming from a Structure and Motion pipeline. To do this, we will also describe a simple heuristic for selecting the reference views used for computing the disparity maps. The final result will be a rough but properly connected model of the scene.

4.2 Related work

The interested reader can find further information on matching measures and aggregation strategies in the following papers [47, 101], which contain some recent and fair comparaison.

Regarding the representation of confidence, literature reports several successfull approaches in stereo matching research. Historically, autocorrelation and left-right consistency constraint have been used to characterize the ambiguity of a pixel, but several other metric exist like for example image entropy or curvature metric [25].

The notion of distinctiveness maps [61], recently reinterpreted by [103], or that of stability [79] are also reconducible to confidence measure.

Confidence is usually employed to guide the matching process or the constraint enforcement in a high confidence first fashion, or as a weighting function in depth map fusion [36].

Our approach, instead of computing confidence as a by-product of the matching process, extrapolate it *a posteriori* from a initial, given, possibly noisy disparity estimate and use it to directly modulate the underlying matching costs.

4.3 Confidence-based cost modulation

In the past confidence measures have usually been calculated as a function of the entire x, y, d space. We propose instead to infer a confidence measure from a initial, possibly noisy estimate of the sought flow or disparity map and to use it to modulate the underlying matching cost function, as follows:

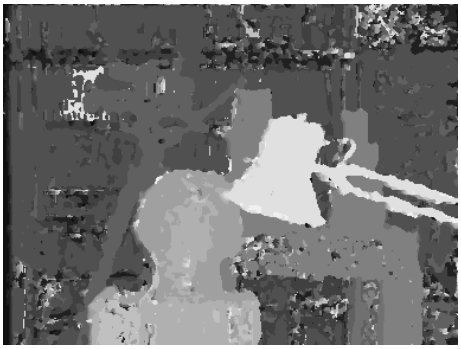

Fig. 4.1. Initial disparity estimation.

$$C'_{x,d} = \frac{\sum_{y \in \mathcal{N}} e^{-\frac{|d_x - d_y|}{k}}}{||\mathcal{N}||} \cdot [C_{x,d} - \mathbf{P}] + \mathbf{P}$$

$C_{x,d}$ and $C'_{x,d}$ are respectively the old and new native matching costs for pixel x at disparity d, $\mathcal{N}$ is the neighbourhood of x and d_w represents the disparity value of location w in the given initial estimate of the disparity map.

The value assumed by the first fraction is proportional to the ratio of pixels with a similar disparity value found in the chosen neighbourhood (the notion of "similarity" is controlled by the parameter k). This ratio is then used to modulate a linear interpolation between the actual cost $C_{x,d}$ and the penalization constant $\mathbf{P}$.

We purposely not inserted any locality principle or distance based penalty because we wanted our operator to be able to non-uniformly incentivate similar regions even if distant or unconnected. The global effect of the operator, when properly configured, is to enable the self-organization of the support regions, favoring compactness and inhibiting small or isolated areas. Thin structures, once established, usually provide themselves enough support to thrive.

4.4 Stereo algorithm

In order to evaluate our confidence modulation we developed as a testbed a simple, local stereo algorithm based on a greedy, fixed window correlation algorithm. Such methods are simple to implement and well-understood, letting us concentrate on assessing and factoring out the properties and the effects of our proposal.

We stress that, even if the resulting algorithm is capable of realtime performance and overall produces decent results it was never meant to be compared with the current state of the art but just as a evaluation platform.

4.4.1 Initial disparity estimation

We start by calculating a initial approximate disparity needed for our confidence operator. We choose as cost matching a truncated version of the popular Birchfield

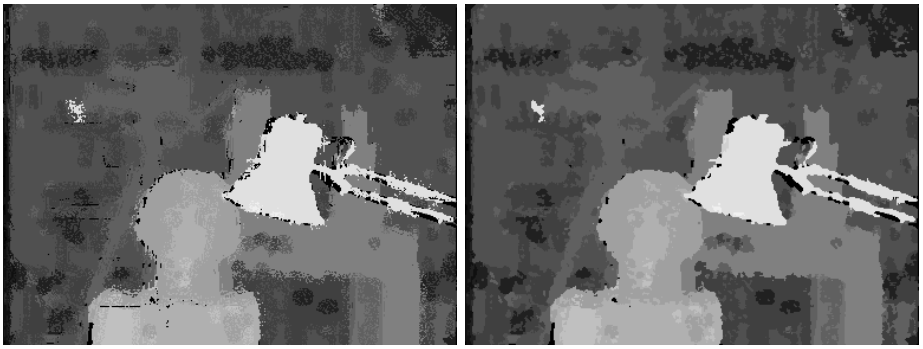

Fig. 4.2. Raw and aggregated output of the confidence estimator.

and Tomasi sampling insensitive measure [4]. To aggregate matching cost, we use a 5x5 gaussian filter with σ=2. The resulting disparity map, shown in fig.4.1, displays all the typical shortcomings of fixed-window correlation algorithms.

4.4.2 Aggregation with modulated costs

Subsequently, we compute a novel disparity map using the confidence values extrapolated from the estimate built in the previous step. Our neighbourhood choice is a uniform disk of radius 7.

The left side of figure 4.2 shows the map obtained when not using any form of cost aggregation: each pixel then assumes the disparity value that minimizes its cost. The picture presents some curious visual artifacts near discontinuities, caused by the influence of pixels across the depth gap. On the right the same cost volume is shown but aggregated with a small 3x3, σ=1 gaussian filter.

What both pictures have in common is a drastic decrease of the noise levels with respect to the initial disparity estimation. Other effects include the reduction of border bleeding and the minor entropy of untextured region which are not filled with a wrong yet uniform disparity layer.

4.4.3 Disparity cleaning

In this step we apply some common and simple heuristics to remove small and untextured regions. To remove this second category we compute an estimate of the noise magnitude and variance and use them to threshold the sum of pixel-wise matching costs (fig.4.3). The resulting regions are then assigned to the best overall disparity for the entire group. Small holes caused by removing small regions are filled with the minima between the neighbouring left and right disparity. On the right of figure 4.3 is shown the resulting depth map.

4.4.4 Final regularization step

Since in the previous step we have obtained a new disparity estimate, we can now use it again to produce a confidence modulated depth map. The resulting depth

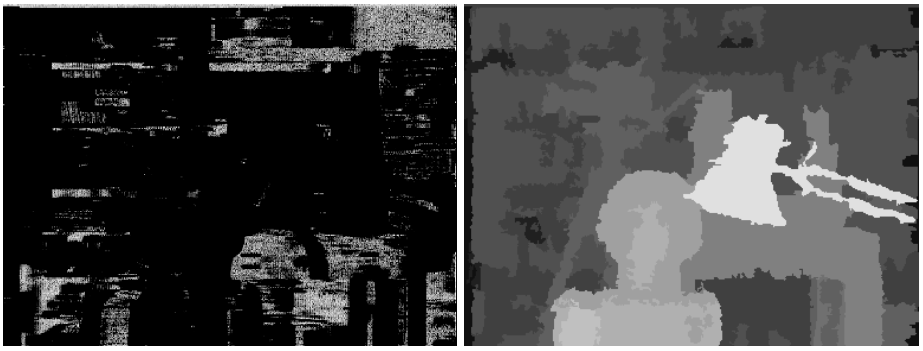

Fig. 4.3. Untextured regions and the cleaned depth map.

map is further checked for consistency using the unicity constraint. The final result is shown in fig.4.4. It is surprisingly good if considering that it was produced from a standard winner-take-all window correlation algorithm.

4.5 Experiments

In order to obtain quantitative results we have run the algorithm described in the previous section on all the four couples in the Middlebury stereo benchmark with and without the proposed confidence based cost modulation. The obtained results are reported in table 4.1.

Our complete results on the Middlebury dataset are shown in figure 4.5: skipping the analysis of the Tsukuba set that was already covered in section 4 we proceed to notice in the Venus couple some of the shortcomings of our näive approach: most of the bad pixels (marked in black) come from disparity holes erroneously filled across disparity boundaries and over untextured region.

The Teddy and Cones couples share the same problems, but moreover they exhibit strong lateral interference. Regarding Teddy, the whole ground plane is missing, due to its steep angle and fine texture and are completely incompatible with fixed-window correlation algorithms.

Fig. 4.4. Final disparity map for the Tsukuba dataset.

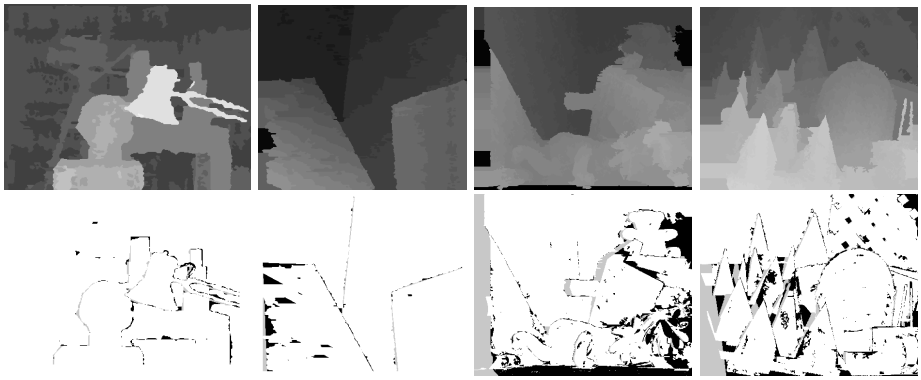

Fig. 4.5. Results on the Middlebury dataset: Tsukuba, Venus, Teddy and Cones.

Table 4.1. Comparison of stereo results.

Image pair	with Cost modulation	without
Tsukuba	1.77_{23}	8.7_{41}
Venus	2.33_{32}	22.1_{41}
Teddy	15.0_{34}	24.3_{41}
Cones	10.0_{33}	21.4_{41}
Overall	30.9	76.5

Overall, we can state that the obtained results are surprisingly good considering that they were computed with a local algorithm just by modulating its cost function. The method, while stabilizing itself in the lower half of the Middlebury table is at par with low-end global algorithm implementations.

4.6 Stereo for surface extraction

We finally apply the algorithm developed in the preceding part of this chapter to enhance a point cloud obtained from a set of fifteen images of the church of Valbonne, popularized from self-calibration literature.

We will use the obtained disparity maps to cluster the cloud points into groups belonging to a single surface of the scene. Each recovered group will be triangulated in order to obtain a three dimensional model.

Given our objective, is not necessary to compute and merge all possible disparity maps; it suffices to select the minimal number of couples necessary to cluster the majority of points.

There are several possible criteria we could take into account: a ideal pair has a sufficiently large baseline, lots of common tracks to jumpstart the stereo process and a limited disparity range. This last requirement promotes couples containing surfaces parallel to both image planes, a favourable configuration for most stereo matching approaches.

Our selection heuristic uses a weighted combination of the number of common tracks and the disparity range to select the current best couple, which is rectified

Fig. 4.6. The first two couples automatically recovered by the proposed view selection heuristic and the corresponding (left) disparity map.

before computing its disparity maps. Surfaces are identified in the depth maps, grouping pixel with neighbouring disparity values. Visible points reprojecting into the grouped areas are assigned to them and not considered in the following steps. The process is repeated until the majority of points have been assigned to a surface.

Figure 4.6 shows the two couples selected in the Valbonne experiment which was sufficient to cover the 96% of the point cloud. The missing point were either not covered by any disparity value or placed in ambiguous locations, such as the internal corners of the building.

The preference of the chosen selection heuristic for frontoparallel surfaces can be used in the triangulation phase to connect the clustered points as they appear when projected onto the images. Figure 4.7 shows the three clusters obtained on the Valbonne dataset and the corresponding 3D model obtained discarting the triangles spanning over regions not covered by the underlying surface.

Fig. 4.7. The clustered point cloud and the corresponding triangulated model.

5

Visualization of Urban and Architectural Models

State of the art three dimensional reconstruction pipelines can nowadays produce models up to several million polygons without any human intervention from a set of digital images or video. Such models are able to stretch the rendering capabilities of current hardware.

The problems however is not exclusively technical, because of the magnitude of the reconstructed data; a point cloud is not easily editable, and does not retain any information about the grouping or connectivity of the original scene.

With this chapter we consider several options for the rendering of recovered models alternative to densification and/or triangulation. Along the way, we will explore automatical recovery of dominant surfaces, obtaining a more high level model of the imaged environment.

5.1 Fitting of geometric primitives

Our first proposal is to augment a typical structure from motion pipeline with two additional steps, automatic fitting of high-level solid primitives and relief maps extraction, thus recovering both the overall structure of a building and its fine geometry. The objective is to obtain a more tractable and semantic model of the imaged scene, allowing for efficient and compelling rendering.

5.1.1 Introduction

The recent advances in Structure and Motion pipelines coupled with the availability of large repositories of digital photos and aerial images have enabled the creation of some of the largest architectural models ever composed [37, 63].

Even if the problems arising in the visualization of large, detailed urban environments have been actively investigated, rendering such massive amounts of data is still problematic. The major source of difficulty lies in the fact that there is virtually no limit to the nature and quantity of the acquired details. State of the art systems can already build scenes with features spanning more than three scales of magnitude. On top of that, recovered meshes suffer from uneven sampling and connectivity problems.

Such magnitude and complexity is able to stretch the rendering capabilities of current rendering platforms, even when taking into account their steady power growth. To speed up the visualization process and to counter the exponential increase in size of the recovered data we propose to augment the typical structure from motion pipeline with two additional steps, high-level primitive fitting and relief map extraction.

High-level primitives such as planes and generalized cones are ideal descriptors for architectural buildings and in general human manufacts. Automatically fitting such primitives to the outputs of a reconstruction pipeline enable the characterization of structure and the extraction of high level properties (such as symmetry, or function) and unseen geometry. Relief map extraction recovers the fine geometry that is lost in the previous step, and stores it in a compact format directly usable by graphic hardware.

The final output of such a system is a set of automatically recovered geometric primitives, relief maps and textures that can be used to concisely describe and to efficiently render the imaged scene. The process leverages the former dense point cloud to a sensible, editable representation ready for manipulation in a CAD software.

5.1.2 Previous art

One of the most scalable approach in Structure and Motion recovery for urban environments was shown in [17], developed for compact visualization on consumer navigation products. Road ground and building façades were forced to lie on textured, mutually-orthogonal, gravity-aligned, geo-located planes. The resulting system is fast but heavily constrained, thus trading efficiency for expressive power.

More generic systems have been demonstrated for the reconstruction of the semantic structure of urban elements. The two most similar articles to our proposal are [21] and [82].

In [21] is described a system that specializes in creating a architectural models from a limited number of images. Initially a coarse set of planes is extracted by grouping point features; the models are subsequently refined by casting the problem in a Bayesian framework where priors for architectural parts such as doors and windows are incorporated or learnt.

A similar deterministic approach is developed in [82] where dominant planes are recovered using a orthogonal linear regression scheme: façade features, which are modeled as shaped protrusions or indentations, are then selected from a set of predefined templates.

Both methods rely on a large amount of prior knowledge to operate, either implicitly or explicitly, and make strict assumption on the imaged scene.

In our approach instead, the amount of injected prior knowledge is limited to the non-critical type and number of primitives used: the recovery process rather than being top-down is entirely data-driven, and structure emerges from the data rather than being dictated by a set of potentially incorrect architectural priors.

Relief maps, not present in the two aforementioned methods, serve both to preserve the information necessary for accurate rendering and to decouple the

numerical errors inherently present in the stereo reconstruction process from the recovery of structure.

While the problem of fitting quadric primitives has been extensively investigated in literature (see [73] for a survey of the topic) most of published material is designed to be applied to dense point clouds produced by laser scanners or to already triangulated meshes. Common assumptions include uniform sampling and negligible acquisition noise; such methods can't therefore be used for processing 3D clouds produced by Structure and Motion pipelines which don't provide connectivity, are un-evenly sampled and corrupted by a comparatively large signal-to-noise ratio.

5.1.3 High-level primitive fitting

Literature offers several algorithms for model estimation: we selected the approach described in [105] because natively developed for multiple structures.

Given a distribution of points corrupted by outliers, the algorithm generates a set of model hypotheses by repeteadly drawing at random the minimal required number of samples for each desired structure, such as planes, cylinder or spheres. The actual number of hypothesis that must be constructed can be calculated knowing the number of points and estimating the percentage of outliers in the dataset.

After hypotheses have been generated, their residual is calculated for each data point. The number of models is estimated analyzing for each data point the peaks in the histogram of the hypotheses residuals. This approach enables data self-organization and requires fewer samples than solutions based on naive RANSAC algorithms. The final number of models is calculated taking the median of all the estimates: for each hypothesis the correct supporting cluster is then identified.

Figure 5.2 shows a model in which every point has been attributed to one of the planes of which the scene is composed.

5.1.4 Image consistent triangulation

At this point we can proceed to relief map extraction with the models recovered in the preceding step, or try to obtain from each of them a triangulated patch.

Estimating a sound triangulation on the output of a structure and motion pipeline is inherently difficult because the recovered 3D information suffer from uneven sampling and reconstruction errors. This inhibits the use of a large part of algorithms for recovering meshes from unorganized point clouds like for example [48]. Therefore, we turn our attention to *image-consistent* triangulation algorithms, i.e., algorithms that uses information from the images to guide the triangulation of 3D points.

Following [15] we first augment our point cloud by adding points along the intersections between the recovered primitives, provided that these points projects onto actual image edges. As a result the model's boundaries are better preserved, as seen in Fig. 5.3.

The initial triangulation is calculated by projecting the recovered 3D points to their belonging surface and applying the 2D Delaunay triangulation algorithm.

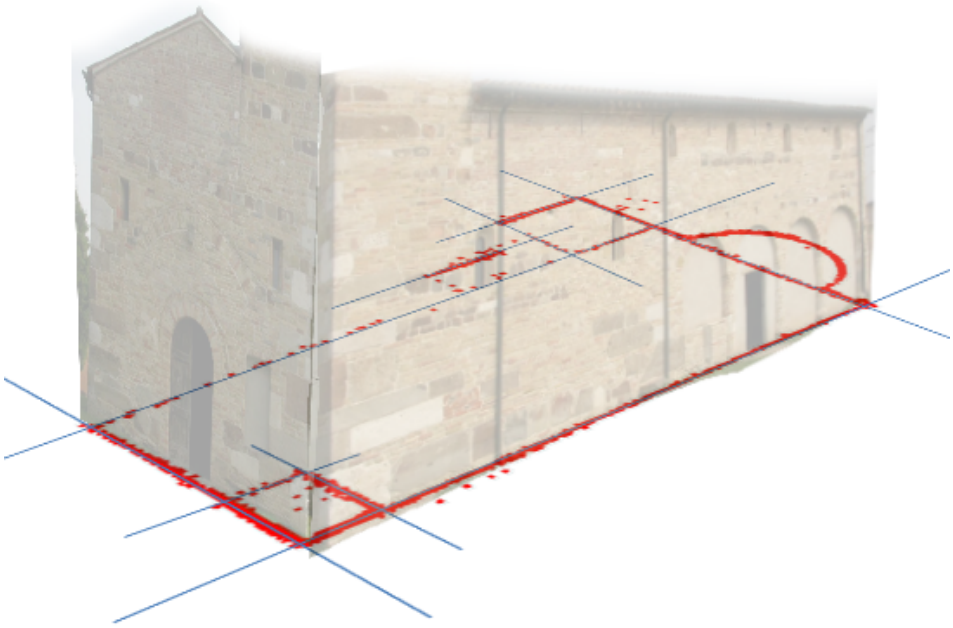

Fig. 5.1. Automatically recovered perimetral planes from the 3D point cloud.

This approximation contains *spurious triangles* that does not correspond to a planar patch in 3D, either linking originally separated surfaces or spanning over concavities of the point cloud projection (since a Delaunay triangulation is convex by construction). They can be eliminated using a series of boundary-preserving heuristics, based on photo-consistency. Sky patches, which being uniform usually survive such checks, have to be dealt separately: they are trimmed from the resulting triangulation starting from the outer border, looking for a strong edge signaling a foreground object. Results are shown in Fig. 5.4.

5.1.5 Relief map extraction

We can be build relief maps both on the original models fitted to the point cloud data or on the triangulated mesh recovered in the preceding section.

In either case, for the recovery of relief maps we developed a simplified version of a recent stereo algorithm based on gestalt principles [104]. Alternatively, the stereo algorithm developed in chapter 4 could be used. While based on local methods, it can achieve good performance by employing large disparity neighbourhoods. The problem usually associated with large correlation windows are minimized by weighting the stereo cost function with a measure of similarity and proximity between candidate matches, thus mimicking the behaviour of stereo algorithms based on explicit segmentation.

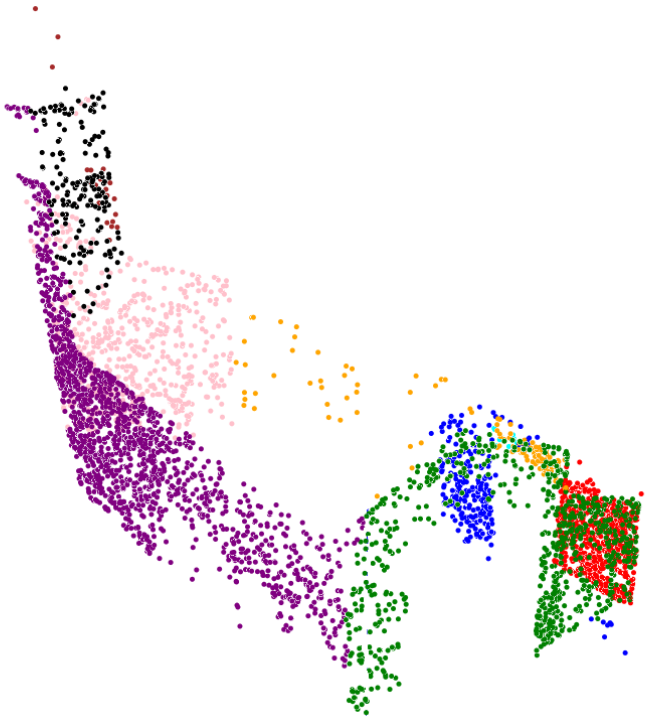

Fig. 5.2. Planes recovered by model fitting. The colour of the point encodes the plane it belongs to.

Candidate views for disparity estimation are selected by identifying those that both contain a large set of visible points from the considered surface. The views are

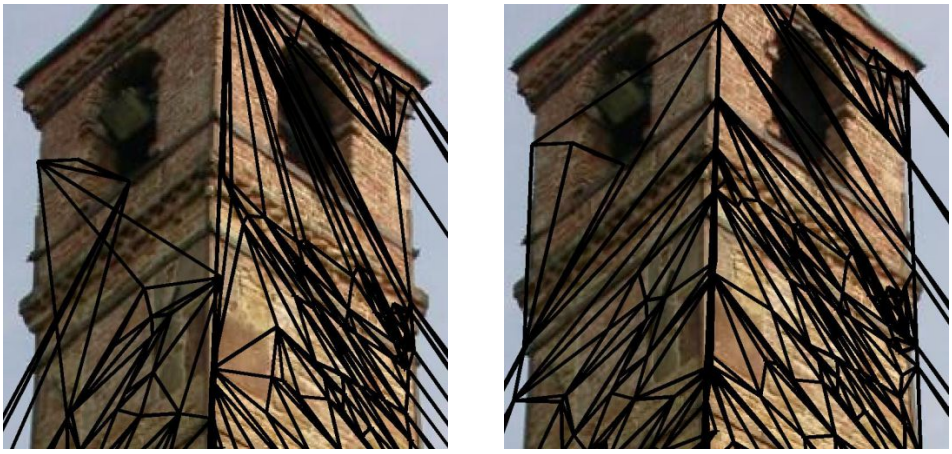

Fig. 5.3. Detail of the triangulation before (left) and after (right) augmentation with boundary points.

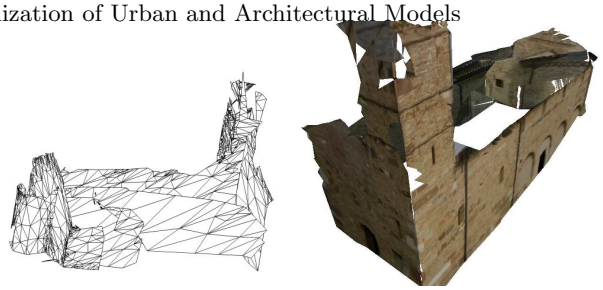

Fig. 5.4. The final triangulated model for the "Pozzoveggiani" example and its textured version.

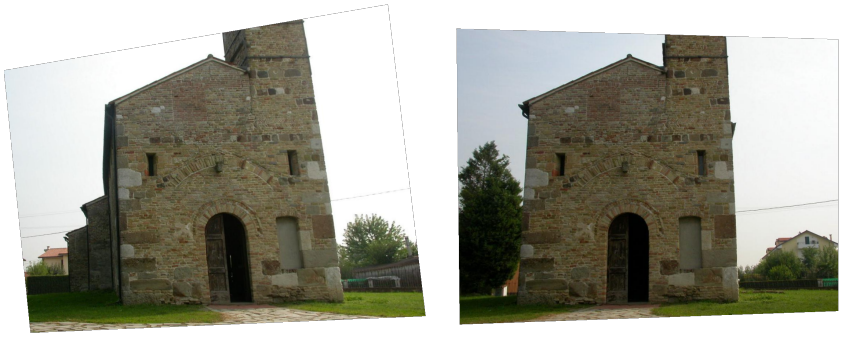

Fig. 5.5. Rectified images used for the recovery of the front façade.

first rectified, discarting during the process the couples with excessive distorsions. Conflicts is depth arising from different couples are resolved taking the median of the estimates.

Once disparity has been obtained recovering bump, normal and displacement maps is straighforward; these data enables the simulation of fine geometry and the use of modern rendering algorithm such as [53] and its more recent derivations.

Figure 5.5 shows two views after homographical rectification [34] and Fig. 5.6 the color and normal maps resulting from the relief map extraction. The extracted map encodes both fine geometry and architectural features, modeling wall extrusions as well as windows and arches.

A detail of the façade is analyzed in Fig. 5.7 where we compare side by side the dense regular mesh generated by the matching process with two renderings composed of a single polygon. The effect of parallax mapping, enabled in the last image, are particularly noticeable in correspondance of the door extrusion.

5.2 Texturing fitted surfaces

In this section we explore the possibility of texturing architectural models in a completely automated way. In the preceding part, scene and texture boundaries were defined by the recovered triangulation or by the valid regions of computed disparity maps. We show here how to compute texture atlases and masks from the results of a structure and motion pipeline.

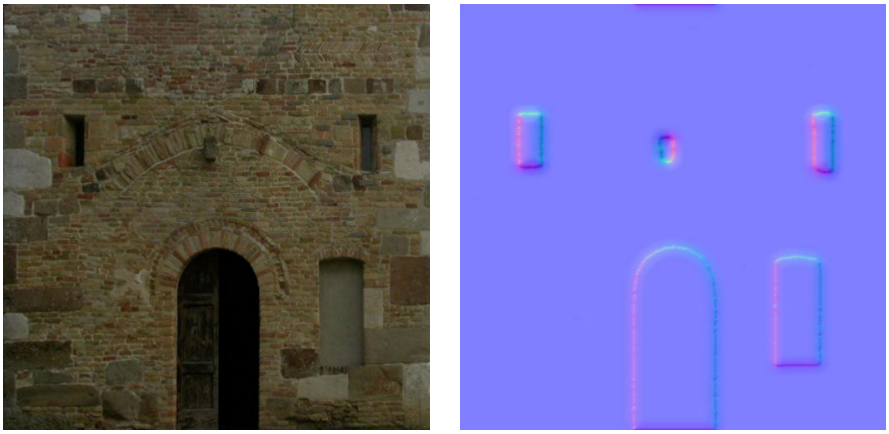

Fig. 5.6. Color and normal textures automatically generated for the front of the church.

Fig. 5.7. From left to right, the dense mesh generated from stereo matching, a single quad textured without fine geometry and the same quad with parallax mapping.

The two main goals in this case are the scalability and fidelity of the obtained rendering and its capability to easily support user navigation. The problem here is to give each picture context, to guide the exploration. We have already shown in the preceding parts how this is possible using triangulated models; we tried also rendering the point cloud through point splatting, or using oriented disks as shown in figure 5.8.

Of course using the scene geometry as a proxy for projecting the captured imagery isn't the only solution capable of achieving the aforementioned goals. A different philosophy is employed in the Photosynth software (http://photosynth.net): only a single picture is ever shown at full resolution from a vantage point; other pictures that can be related to the reference one by a homography are used to provide the context for the user to understand and navigate the collection. An example of this approach produced within our system is shown in Fig 5.9.

Such a representation has a number of interesting properties: it supports spatial navigation and provides excellent visual fidelity, since the reference picture is always seen from the position from which it was shoot, and augmented with a

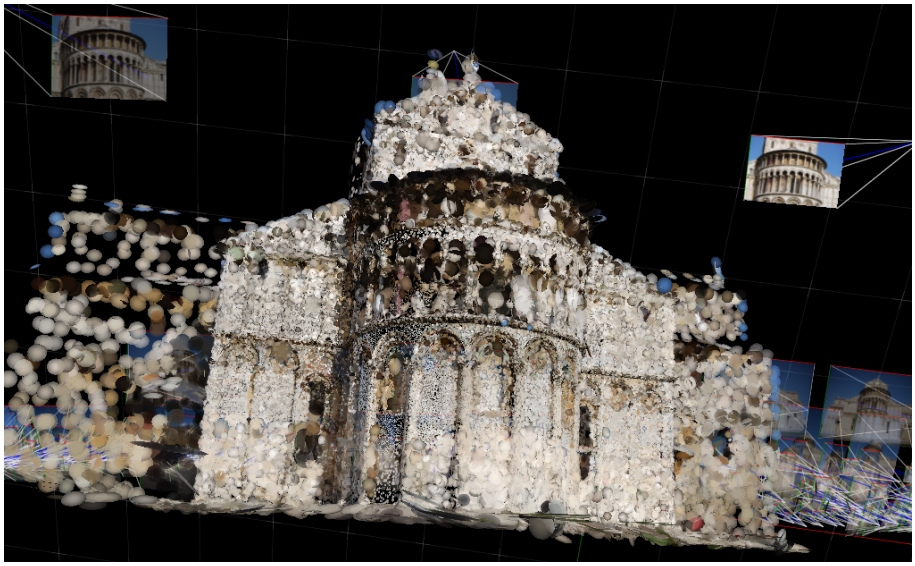

Fig. 5.8. Rendering with oriented disks.

Fig. 5.9. A planar stitch of pictures in 3D.

relevant context. On the other hand however, it is structurally unable to capture image relashionships that can't be modeled with a collinearity.

These problems can be overcomed by using the high level models as a proxy for the scene geometry, and rendering the photo collection against them. To be scalable and effective however, this approach must be coupled with a way to select from a arbitrary position in 3D the subset of the available views which maximizes the visual fidelity while containing the computational workload. We will see how in the following section.

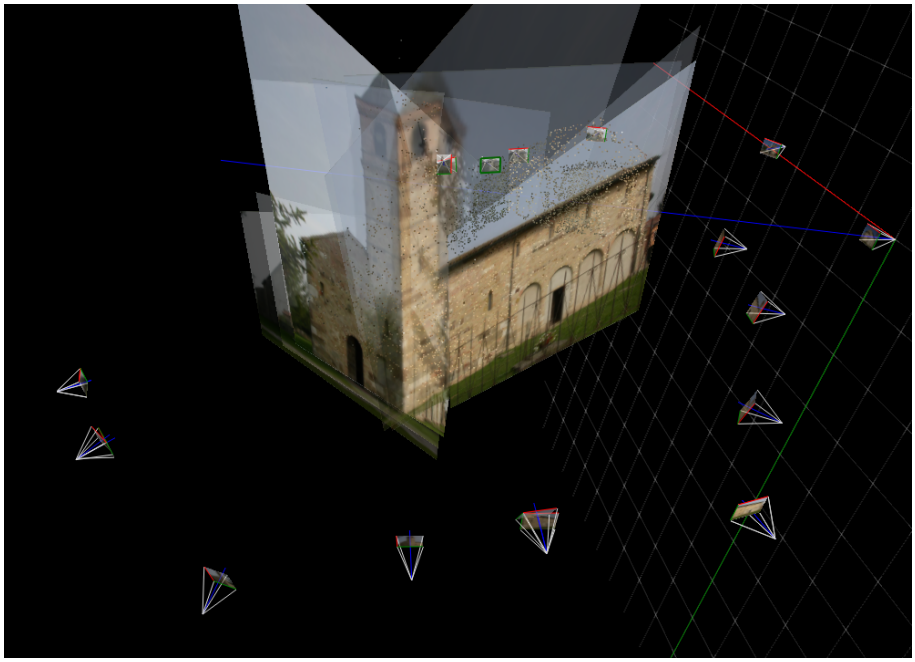

Fig. 5.10. Unmasked rendering on the recovered primitives.

5.2.1 View-model affinity

We first consider the problem of selecting, given a reference view, a number of additional views that will provide the context for the reference one. This can be done in several ways: one possible solution is to simply select the nearest neighbours induced by a distance function on the camera parameters, like the following one:

$$d(G_{ref}, G) = \min(\log(||G_{ref}^{-1}G||), \log(||GG_{ref}^{-1}||))$$

where G are the extrinsic parameters of the considered camera. This metric usually gives good result when the scale and the intrinsic parameters of the cameras are roughly the same.

In the general case however, selecting views that contain a large number of common 3D features has shown itself a much more stable heuristic, capable of automatically coping with scale changes and camera tilt. Such characteristics are important for selecting a range of images with sufficient variability. The same criteria can be used also to evaluate the affinity between a collection of high level primitives and a view.

When realizing that a arbitrary position and direction in space specified by a virtual camera is akin to a regular view, it becomes possible to select both the models and cameras that have affinity with a arbitrary point in space.

With these data, each selected view can then be rendered using projective texture mapping on the proxy geometry that the high level primitives constitute. If needed, the fine details lost in the primitive extraction can be encoded in displacement or relief maps, as suggested earlier.

5.2.2 Mask creation

The process described in the previous section however is not sufficient to guarantee an artifact-free rendering, as Fig. 5.10 clearly shows. These effect can be avoded masking the projection over each recovered primitive.

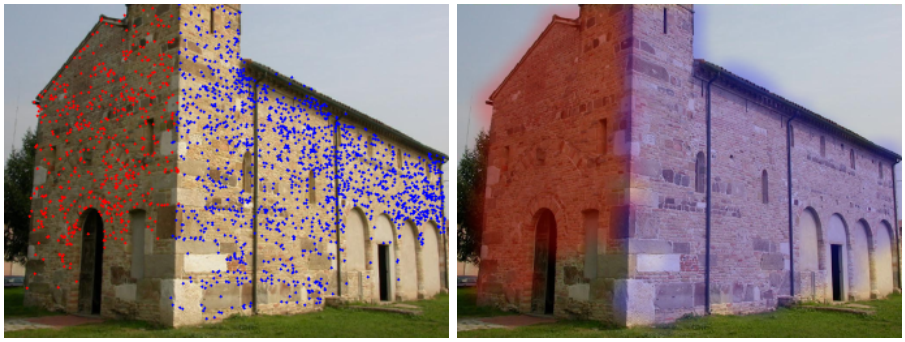

Fig. 5.11. Points on two different planes and their recovered masks.

The problem can be solved creating the mask for each primitive back-projecting its points onto the image plane, and extracting a 2D neighbourhood of the obtained points. We found that just thresholding a low pass filtered binary image containing the point projections gave reasonable results.

Figure 5.11 shows the masks obtained from two planar surfaces: as it can be seen the recovered mask follow quite closely the underlying structure. This approach works well when the three dimensional features are evenly distributed: in that case, we obtain surfaces without connectivity problems.

The potential issues with color bleeding on the boundaries between primitives could be further corrected by constraining mask borders to align with the models intersections. In our experience however, the perceived effect of bleeding was unnoticeable.

5.2.3 Results on the Pozzoveggiani dataset

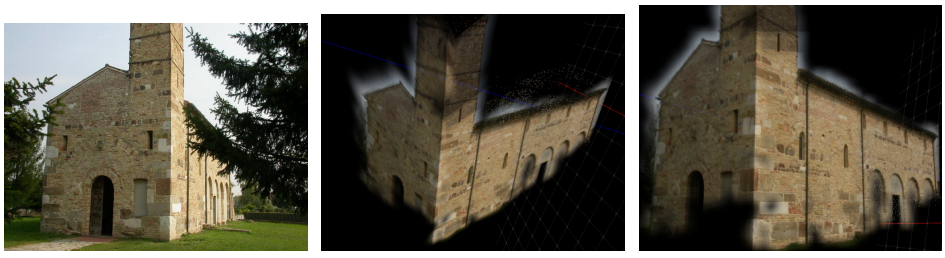

Fig. 5.12. One of the original photo and two novel virtual views of the Pozzoveggiani model from our interactive visualizer.

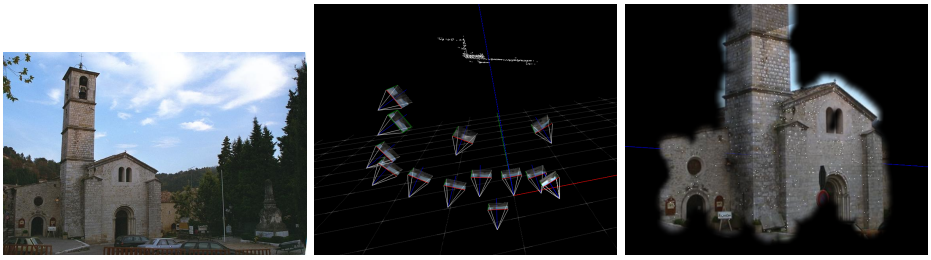

Fig. 5.13. Results for the Valbonne dataset.

The "Pozzoveggiani" dataset portraits a small church near Padua (IT) that had been used before in [39] to test photogrammetric reconstruction. It has a simple planimetry: the perimeter is composed of straight walls, with a bell tower and a slanted roof covered with bent tiles. A cylindrical apse protrudes from the back; several arches and slit windows open into the well-textured brick walls.

The picture set is composed of 54 images acquired from the ground plane with a consumer camera at a resolution of 1024x768 pixels, at different times and with automatic exposure. This is the dataset that was chosen to illustrate the various step of the algorithm through this paper: as was shown, our pipeline succeeds in recovering and modeling all the perimetral walls. The good properties of the reconstruction can also be assessed by measuring the average angle between orthogonal planes, which is 90.44 degrees.

Two frames from our interactive are shown in Fig. 5.12; as it can be seen, it correctly models the two surfaces visible from the current view, while discarting the background. The missing parts from the side textures are due to a uneven distribution of the 3D features on the walls; when seen in movement, the model faithfully captures the expected appearance of the scene, guiding the user in the exploration.

5.2.4 Results on the Valbonne dataset

The church of Valbonne is another small church located in France, and extensively used in the computer vision literature. Its stone walls are organized into two dominant, orthogonal directions.

This experiment comprises fifteen photos: the dataset is recorded at a resolution of 768x512 pixels, in varying condition of illumination and occlusion. Again – as shown in Fig. 5.13 – our system successfully recovers all dominant planes and cameras, with the front façade assimilating the contributes of the two protrusions at its sides.

6

Conclusions

In this thesis we have described several improvements to the current state of the art in the context of uncalibrated Structure and Motion from images. Our first result was a hierarchical framework for Structure and Motion, which was demonstrated to best the sequential approach both in computational complexity and with respect to the overall containement of error. Our proposal constitutes the first truly scalable approach to the problem of reconstruction from images, showing a sub-linear complexity in the number of points and cameras.

We then described a novel self-calibration approach, which coupled with our hierarchical pipeline constitutes the first published example of uncalibrated Structure and Motion for generic datasets not using external, ancillary information. The robustness of our approach has been demostrated on 3D reconstruction datasets; it was also used for the upgrade of single pair of cameras with good results, a task generally considered bad conditioned by the relevant literature.

In the second part of this thesis, we demonstrated various rendering modes for point clouds, with the goal of obtaining a efficient and faithful representation of the scene. By using plane and quadric fitting we obtained also compact, semantically meaningful descriptions which can be interpreted as a crude reverse engineering of the environment.

Future research will continue trying to further improve the computational complexity of our Structure and Motion pipeline. A complete C++ port of the algorithms is already underway; it will be coupled with structural improvements such as the addition of novel cameras from existing ones using the already recovered fundamental matrices and a novel summary procedure for points orthogonal to the one described for views in chapter 2. We will also try to integrate in our pipeline reflectance images coming from laser scans, to integrate existing photogrammetric workflows with the Computer Vision approach. We are going to develop sub-linear search strategies for the self-calibration part, which are enabled by the particular structure of the cost functions we employed. Finally, most of the rendering algorithm described here will be updated contextually with the aforementioned C++ port.

References

1. Sameer Agarwal, Noah Snavely, Ian Simon, Steven M. Seitz, and Richard Szeliski. Building rome in a day. In *International Conference on Computer Vision*, Kyoto, Japan, 2009.
2. S. Baker, S. Roth, D. Scharstein, M.J. Black, J.P. Lewis, and R. Szeliski. A database and evaluation methodology for optical flow. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8, 2007.
3. P. Beardsley, A. Zisserman, and D. Murray. Sequential update of projective and affine structure from motion. *Int. Journal of Computer Vision*, 23(3):235–259, 1997.
4. Stan Birchfield and Carlo Tomasi. A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(4):401–406, April 1998.
5. Benoît Bocquillon, Adrien Bartoli, Pierre Gurdjos, and Alain Crouzil. On constant focal length self-calibration from multiple views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007.
6. S. Bougnoux. From projective to Euclidean space under any practical situation, a criticism of self-calibration. In *Proceedings of the International Conference on Computer Vision*, pages 790–796, Bombay, 1998.
7. M. Brown and D. Lowe. Recognising panoramas. In *Proceedings of the 9th International Conference on Computer Vision*, volume 2, pages 1218–1225, October 2003.
8. Matthew Brown and David 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.
9. Matthew Brown and David G. Lowe. Unsupervised 3D object recognition and reconstruction in unordered datasets. In *Int. Conf. 3DIM*, June 2005.
10. http://phototour.cs.washington.edu/bundler/.
11. Manmohan Chandraker, Sameer Agarwal, Fredrik Kahl, David Nister, and David Kriegman. Autocalibration via rank-constrained estimation of the absolute quadric. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 0, pages 1–8, 2007.
12. Manmohan Chandraker, Sameer Agarwal, David Kriegman, and Serge Belongie. Globally optimal affine and metric upgrades in stratified autocalibration. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.
13. Ondřej Chum, Tomáš Pajdla, and Peter Sturm. The geometric error for homographies. *Computer Vision and Image Understanding*, 97(1):86–102, 2005.

14. D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619, 2002.
15. O. Cooper, N. Campbell, and D. Gibson. Automatic augmentation and meshing of sparse 3D scene structure. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION)*, pages 287–293, Washington, DC, USA, 2005.
16. Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, and Clifford Stein. *Introduction to Algorithms*. The MIT Press, Cambridge, MA, USA, 2001.
17. N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *Proceedings of CVPR*, volume 2, pages 1339–1344, 2006.
18. N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1339–1344, 2006.
19. N. Cornelis, B. Leibe, K. Cornelis, and L. Van Gool. 3D urban scene modeling integrating recognition and reconstruction. *International Journal of Computer Vision*, 78(2-3):121–141, July 2008.
20. P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Holly Rushmeier, editor, *SIGGRAPH Conference Proceedings*, pages 11–20, New Orleans, Louisiana, August 1996.
21. A. R. Dick and et al. Modelling and interpretation of architecture from several images. *IJCV*, 60(2):111–134, 2004.
22. A. R. Dick, P. H. S. Torr, and R. Cipolla. Modelling and interpretation of architecture from several images. *International Journal of Computer Vision*, 60(2):111–134, 2004.
23. Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis*, pages 98–105. John Wiley and Sons, 1973.
24. R.P.W. Duin, E. Pekalska, P. Paclik, and D.M.J. Tax. The dissimilarity representation, a basis for domain based pattern recognition? In *Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings*, pages 43–56, Cambridge, UK, 2004.
25. Geoffrey Egnal, Max Mintz, and Richard P. Wildes. A stereo confidence metric using single view imagery with comparison to five alternative approaches. *Image Vision Comput.*, 22(12):943–957, 2004.
26. M. Farenzena and A. Fusiello. 3D surface models by geometric constraints propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage (Alaska), June 24-26 2008.
27. M. Farenzena, A. Fusiello, and R. Gherardi. Efficient visualization of architectural models from a structure and motion pipeline. In *Proceedings of Eurographics - short papers*, Crete, Greece, april 14-18 2008.
28. Michela Farenzena, Andrea Fusiello, Riccardo Gherardi, and Roberto Toldo. Towards unsupervised reconstruction of architectural models. In *Proceedings of Vision, Modeling, and Visualization 2008*, pages 41–50, Konstanz, Germany, October 8-10 2008.
29. O. Faugeras. Stratification of 3-D vision: projective, affine, and metric representations. *Journal of the Optical Society of America A*, 12(3):465–484, 1994.
30. Paul D. Fiore. Efficient linear solution of exterior orientation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(2):140–148, 2001.
31. M. A. Fischler and R. C. Bolles. Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

32. 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.
33. 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.
34. A. Fusiello, E. Trucco, and A. Verri. A compact algorithm for rectification of stereo pairs. *Machine Vision and Applications*, 12(1):16–22, 2000.
35. Simon Gibson, Jon Cook, Toby Howard, Roger Hubbold, and Dan Oram. Accurate camera calibration for off-line, video-based augmented reality. *Mixed and Augmented Reality, IEEE / ACM International Symposium on*, 2002.
36. Michael Goesele, Brian Curless, and Steven M. Seitz. Multi-view stereo revisited. In *CVPR (2)*, pages 2402–2409, 2006.
37. Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of ICCV*, October 14-20 2007.
38. Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of the International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 14-20 2007.
39. Alberto Guarnieri, Antonio Vettore, and Fabio Remondino. Photogrammetry and ground-based laser scanning: Assessment of metric accuracy of the 3D model of pozzoveggiani church. In *FIG Working Week. TS on "Positioning and Measurement Technologies and Practices II - Laser Scanning and Photogrammetry"*, Athens, Greece, 22-27 May 2004.
40. F.R. Hampel, P.J. Rousseeuw, E.M. Ronchetti, and W.A. Stahel. *Robust Statistics: the Approach Based on Influence Functions.* Wiley Series in probability and mathematical statistics. John Wiley and Sons, 1986.
41. R. Hartley, E. Hayman, L. de Agapito, and I. Reid. Camera calibration and the search for infinity. In *Proceedings of the International Conference on Computer Vision*, 1999.
42. R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision.* Cambridge University Press, 2003.
43. R. I. Hartley. Estimation of relative camera position for uncalibrated cameras. In *Proceedings of the European Conference on Computer Vision*, pages 579–587, Santa Margherita L., 1992.
44. R. I. Hartley and P. Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, November 1997.
45. Richard I. Hartley. Chirality. *Int. J. Comput. Vision*, 26(1):41–61, 1998.
46. A. Hilton. Scene modelling from sparse 3d data. *Image Vision Computing*, 23(10):900–920, 2005.
47. Heiko Hirschmüller and Daniel Scharstein. Evaluation of cost functions for stereo matching. In *CVPR*, 2007.
48. H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH Conference Proceedings*, pages 295–302, New York, USA, 1994.
49. Arnold Irschara, Christopher Zach, and Horst Bischof. Towards wiki-based dense city modeling. In *Proceedings of the 11th International Conference on Computer Vision*, pages 1–8, 2007.
50. G. Kamberov and et al. 3D geometry from uncalibrated images. In *2nd Intl. Symposium on Visual Computing*, Springer Lecture Notes in Computer Science, 2006.

51. 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*, Springer Lecture Notes in Computer Science, November 6-8 2006.
52. Y. Kanazawa and H. Kawakami. Detection of planar regions with uncalibrated stereo using distributions of feature points. In *Proceedings of the British Machine Vision Conference*, pages 247 – 256, September 7-9 2004.
53. T. Kaneko, T. Takahei, and et al. Detailed shape representation with parallax mapping. In *Proceedings of ICAT*, pages 205–208, 2001.
54. T. Kaneko, T. Takahei, M. Inami, N. Kawakami, Y. Yanagida, T. Maeda, and S. Tachi. Detailed shape representation with parallax mapping. In *Proceedings of the ICAT 2001*, pages 205–208, 2001.
55. Kuk-Jin Yoon; In-So Kweon. Locally adaptive support-weight approach for visual correspondence search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 924–931, 2005.
56. K. Levenberg. A method for the solution of certain non-linear problems in least squares. *Quarterly Journal of Applied Mathmatics*, II(2):164–168, 1944.
57. M.I.A. Lourakis and A.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.
58. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
59. Q.-T. Luong and O. D. Faugeras. The fundamental matrix: Theory, algorithms, and stability analysis. *International Journal of Computer Vision*, 17:43–75, 1996.
60. Q.-T. Luong and T. Viéville. Canonical representations for the geometries of multiple projective views. *Computer Vision and Image Understanding*, 64(2):193–229, 1996.
61. R. Manduchi and C. Tomasi. Distinctiveness maps for image matching. *Image Analysis and Processing, 1999. Proceedings. International Conference on*, pages 26–31, 1999.
62. S. J. Maybank and O. Faugeras. A theory of self-calibration of a moving camera. *International Journal of Computer Vision*, 8(2):123–151, 1992.
63. P. Mordohai and et al. Real-time video-based reconstruction of urban environments. In *Workshop 3D-ARCH 2007*, July 12-13 2007.
64. D. D. Morris and T. Kanade. Image-consistent surface triangulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 332–338, June 2000.
65. David M. Mount and Sunil Arya. Ann: A library for approximate nearest neighbor searching. In *http://www.cs.umd.edu/ mount/ANN/*, 1996.
66. 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.
67. Atsutada Nakatsuji, Yasuyuki Sugaya, and Kenichi Kanatani. Optimizing a triangular mesh for shape reconstruction from images. *IEICE - Transactions on Information and Systems*, E88-D(10):2269–2276, 2005.
68. G. N. Newsam, D. Q. Huynh, M. J. Brooks, and H. p. Pan. Recovering unknown focal lengths in self-calibration: An essentially linear algorithm and degenerate configurations. In *In Int. Arch. Photogrammetry and Remote Sensing*, pages 575–580, 1996.
69. Kai Ni, Drew Steedly, and Frank Dellaert. Out-of-core bundle adjustment for large-scale 3D reconstruction. In *Proceedings of the International Conference on Computer Vision*, pages 1–8, 2007.

70. 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.
71. P. Mordohai et al. Real-time video-based reconstruction of urban environments. In *3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures*, July 12-13 2007.
72. J.-S. Perrier, G. Agam, and P. Cohen. Image-based view synthesis for enhanced perception in teleoperation. In J. G. Verly, editor, *Enhanced and Synthetic Vision (Proceedings SPIE)*, volume 4023, pages 213–224, June 2000.
73. S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 2:1–61, 2002.
74. M. Pollefeys, L.V. Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision 2002*, number 34 in International Archive of Photogrammetry and Remote Sensing., page 252 258, 2002.
75. 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.
76. 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.
77. M. Pollefeys, F. Verbiest, and L. Van Gool. Surviving dominant planes in uncalibrated structure and motion recovery. In *Proceedings of the European Conference on Computer Vision*, pages 837–851, 2002.
78. Till Quack, Bastian Leibe, and Luc 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.
79. Radim Sara. Finding the largest unambiguous component of stereo matching. *Proceedings 7th European Conference on Computer Vision (ECCV2002)*, 2:900–914, may 2002.
80. Frederik Schaffalitzky and Andrew 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.
81. D. Scharstein and R. Szeliski. A taxomomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision, IJCV*, 47(1):7–42, april 2002.
82. Konrad Schindler and Joachim Bauer. A model-based method for building reconstruction. In *Proceedings of HLK'03*, page 74, Washington, DC, USA, 2003. IEEE Computer Society.
83. Konrad Schindler and Joachim Bauer. A model-based method for building reconstruction. In *Proceedings of the First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling And Motion Analysis*, page 74, Washington, DC, USA, 2003.
84. Yongduek Seo, Anders Heyden, and Roberto Cipolla. A linear iterative method for auto-calibration using the dac equation. In *CVPR (1)*, pages 880–. IEEE Computer Society, 2001.
85. Yongduek Seo, Anders Heyden, and Roberto 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.

86. J. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1–3):86–95, 2002.
87. Heung-Yeung Shum, Qifa 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.
88. Ian Simon, Noah Snavely, , and Steven M. Seitz. Scene summarization for online image collections. In *Proceedings of the Internation Conference on Computer Vision*, 2007.
89. 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.
90. Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH Conference Proceedings*, pages 835–846, NY, USA, 2006.
91. Drew Steedly, Irfan Essa, and Frank Dellaert. Spectral partitioning for structure from motion. In *Proceedings of the International Conference on Computer Vision*, pages 649–663, 2003.
92. P. Sturm. On focal length calibration from two views. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume II, pages 145–150, Kauai, USA, 2001.
93. T. Thormählen, H. Broszio, and A. Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In Tomás Pajdla and Jiri Matas, editors, *Proceedings of the European Conference on Computer Vision*, volume 3021 of *Lecture Notes in Computer Science*, pages 523–535, 2004.
94. R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *Proceedings of the European Conference on Computer Vision*, Nice, FR, October 2008.
95. Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with j-linkage. In *Proceedings of the European Conference of Computer Vision*, volume 1, pages 537–547, Marseille, France, October 12-18 2008.
96. 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.
97. 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.
98. 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.
99. M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Machione Vision and Applications*, 17(6):411–426, 2006.
100. Maarten Vergauwen and Luc Van Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006.
101. Liang Wang, Mingwei Gong, Minglun Gong, and Ruigang Yang. How far can we go with local optimization in real-time stereo matching. *3D Data Processing, Visualization, and Transmission, Third International Symposium on*, pages 129–136, June 2006.
102. L. Xu, E. Oja, and P. Kultanen. A new curve detection method: randomized Hough transform (RHT). *Pattern Recognition Letters*, 11(5):331–338, 1990.
103. Kuk-Jin Yoon and In So Kweon. Stereo matching with the distinctive similarity measure. *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–7, 14-21 Oct. 2007.

104. Kuk-Jin Yoon and In-So Kweon. Locally adaptive support-weight approach for visual correspondence search. *IEEE Conf. CVPR*, 2:924–931 vol. 2, 20-25 June 2005.
105. Wei Zhang and Jana Kosecká. Nonparametric estimation of multiple structures with outliers. In *WDV*, pages 60–74, 2006.
106. Wei Zhang and Jana Kosecká. Nonparametric estimation of multiple structures with outliers. In *Workshop on Dynamic Vision, European Conference on Computer Vision 2006*, volume 4358 of *Lecture Notes in Computer Science*, pages 60–74, 2006.
107. 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.
108. M. Zuliani, C. S. Kenney, and B. S. Manjunath. The multiRANSAC algorithm and its application to detect planar homographies. In *Proceedings of the IEEE International Conference on Image Processing*, Genova, IT, September 11-14 2005.
109. Marco Zuliani. *Computational Methods for Automatic Image Registration*. PhD thesis, University of California, Santa Barbara, Dec 2006.