SAMANTHA: TOWARDS AUTOMATIC IMAGE-BASED MODEL ACQUISITION

R. Gherardi, R. Toldo, M. Farenzena, A. Fusiello

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15, 37134 Verona (Italy)
name.surname@univr.it

Abstract

In this paper we describe a Structure and Motion pipeline from images which is both more robust and computationally cheaper than current competing approaches. Pictures are organized into a hierarchical tree which has single images as leaves and partial reconstructions as internal nodes. The method proceeds bottom up until it reaches the root node, corresponding to the final result. This framework is one order of magnitude faster than sequential approaches, inherently parallel, less sensitive to the error accumulation causing drift and truly uncalibrated, not needing EXIF metadata to be present in pictures. We have verified the quality of our reconstructions both qualitatively producing compelling point clouds and quantitatively, comparing them with laser scans serving as ground truth. We also show how to automatically extract a meaningful collection of planar patches obtaining a compact, stable representation of scenes.

Keywords: Structure and Motion, Autocalibration, J-Linkage, Modelization.

1 Introduction

Three dimensional content is pervasise in most forms of digital media production, fueling the need for ubiquitous, high quality acquisition of 3D data. In this article we describe a complete, robust procedure for the reconstruction of the shape of the environment and camera poses from a unconstrained set of digital images. Picture datasets are easy to capture, process and update. They have better resolution, contrast, definition of the video that can be produced with equally priced equipment. Pictures have also inferior requirements for storage and globally lower costs for production, maintenance and processing. Images are therefore the preferred way for ubiquitous, low cost acquisition of quality three dimensional data.

Relevant literature comprises several 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 [4, 18, 32, 38, 17].

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

A class of solutions that have been proposed are the so-called *partitioning methods* [10]. They reduce the reconstruction problem into smaller and better conditioned subproblems which can be effectively optimized. The subproblems can be selected analytically as in [33], where spectral partitioning has been applied to SaM, or they can emerge from the underlying 3D structure of the problem, as described in [23]. 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.

A 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 [10], using a balanced tree of trifocal tensors over a video sequence. The approach was subsequently refined by [24], adding heuristics for redundant frames suppression and tensor triplet selection. In [29] 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 [13], 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 paper [31] that works with sparse dataset describes a way to select a subset of images whose reconstruction provably approximates the one obtained using the entire set. This considerably lowers the computational requirements by controllably removing redundancy from the dataset. Even in this case, however, the images selected are processed incrementally. Moreover, this method does not avoid computing the epipolar geometry between *all* pairs of images.

A third solution is 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.

Our proposal is a hierarchical and parallelizable scheme for SaM. The images are organized into a hierarchical cluster tree, the reconstruction proceeding from leaves to the root. Partial reconstructions correspond to internal nodes, whereas images are stored in the leaves (see fig. 1). This scheme provably cuts the computational complexity by one order of magnitude (provided that the dendrogram is well balanced) and achieves scalability by partitioning the problem into smaller instances and combining them hierarchically in a inherently parallelizable way. It is also less sensible to typical problems of sequential approaches, namely sensitivity to initialization [34] and drift [6]. This approach has some analogy with [28], 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.

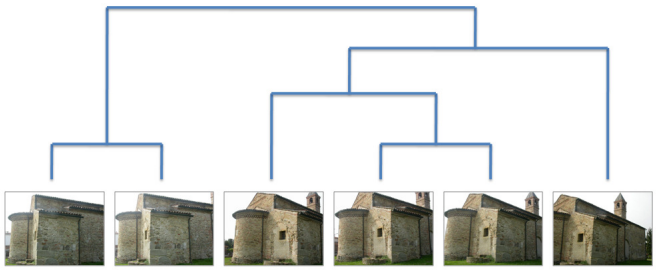


Figure 1: An example of dendrogram for a 6 views set.

Most existing pipelines either assumes known internal parameters [4, 17], or constant internal parameters [38, 18], or relies on EXIF data plus external informations (camera CCD dimensions) [32]. Our second contribution is to endow the SaM pipeline with the capability of dealing with uncalibrated images with varying internal parameters and no ancillary information, using a novel auto-calibration procedure robust enough to be applied in a Structure and Motion context.

Our third and final proposal is to extract photo-consistent planar patches from the obtained point cloud. Planar patches are a very compact and stable intermediate representation of 3D scenes, and a good starting point for a complete automatic reconstruction of surfaces. They can also be regarded as a rough semantic representation of the scene and as a one-step-forward through the automatic scene understanding. The filling of the *semantic gap* that separates unstructured cloud of points from higher level renditions, is a one of the most challenging research area in Computer Vision. Our approach differs from the current state of the art algorithms that works only with simple noise-free scenes [21, 2] or produce a dense points cloud as output [11]. Our solution does not employ any prior knowlodge and it is completely automatic. Furthermore it is efficient, since it works directly on the sparse points cloud produced by a SaM, without the need of densifying it.

The remainder of this article is organized as follows. The next section outlines the matching stage, then Sec. 3 describes the way the hierarchical cluster tree is built. Section 4 presents the hierarchical approach to structure and motion recovery, whereas the autocalibration strategy is explained in Sec. 5. We will then describe the photo-consistent planar patch extraction phase. Experimental detailed in Sec. 7, and finally conclusions are drawn in Sec. 8.

2 Keypoint Matching

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

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 [3]. SIFT [19] keypoints are extracted in all n images. In this culling phase we consider only a constant number of descriptors in each image (300 in our experiments, 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 $\ell = 8$). 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 [22]). 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.

Matching follows a nearest neighbor approach [19], 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 [37]. Let e_i be the residuals after MSAC, following [39], the final set of inliers are those points such that

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

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

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

This outlier rejection rule is called X84 in [14].

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 [20, 5].

Figure 2: An example of one image (top left) from "Piazza Bra" and its six closest neighbors according to the affinity defined in Eq. 3.

The more likely model (homography or fundamental matrix) is selected according to the Geometric Robust Information Criterion (GRIC) [36]. Finally, if the number of remaining matches between two images is less than a threshold (computed basing on a statistical test as in [3]) then they are discarded.

Keypoints matching in multiple images are connected into *tracks*, rejecting as inconsistent those tracks in which more than one keypoint converges [32] and those shorter than three frames.

3 Views Clustering

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

Algorithms for image views clustering have been proposed in literature in the context reconstruction [28], panoramas [3], image mining [26] and scene summarization [30]. The distance being used and the clustering algorithm are application-specific.

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

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

where S_i and S_j are the set of matching keypoints in image I_i and I_j respectively, $CH(\cdot)$ is the area of the convex hull of a set of points and A_i (A_j) is the total area of the image. Figure 2 shows an example of the neighborhood defined by this affinity.

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

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

This procedure allows to decrease the computational complexity with respect to a sequential SaM pipeline, from $O(n^5)$ to $O(n^4)$ in the best case (see [12] for a complete proof), i.e. when the tree is well balanced (n is the number of views). If the tree is unbalanced this computational gain vanishes. It is therefore crucial to enforce the balancing of the tree and this is the goal of the technique that we shall describe in this section.

In order to produce better balanced trees and approximate best-case complexity, we modify 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. 7) that a good compromise is $\ell = 5$. An example is shown in 3. The height of the tree is reduced from 14 to 9 and more initial pairs are present in the dendrogram on the right. Computational complexity decrease accordingly.

Extra care must be taken when building clusters of cardinality two. These are pair of images from which the reconstruction will start, hence pairs related by homographies should be avoided. This is tantamount to say that the fundamental model must explain the data far better than an homography, and this can be implemented by considering the GRIC, as in [25]. We therefore modify the linkage strategy so that two views i and view j are allowed to merge in a cluster only if:

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

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

4 Hierarchical Structure and Motion

The dendrogram produced by the clustering stage imposes a hierarchical organization of the views that will be followed by

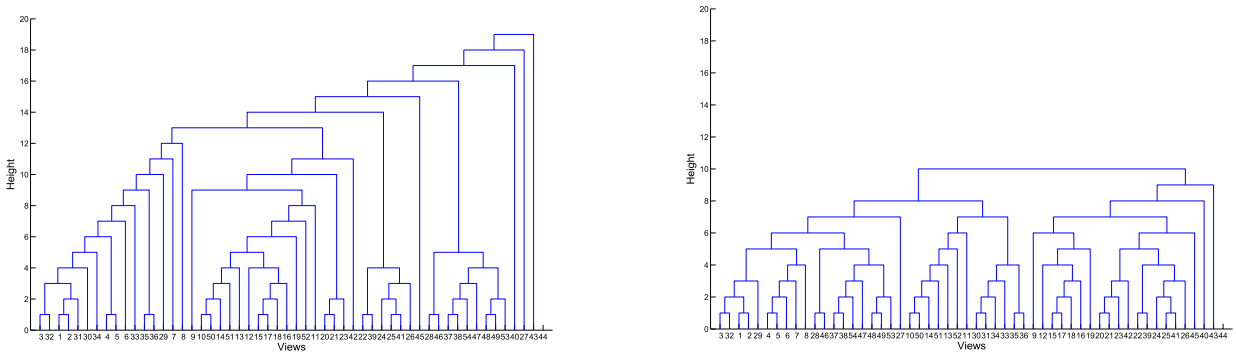


Figure 3: Two dendrograms produced on a 52-views set. The left one was produced using the standard simple linkage rule, the right using the modified rule, with $\ell = 5$.

our SaM pipeline. At every node in the dendrogram an action must be taken, that augment the reconstruction (cameras + 3D points): a two views reconstruction is performed when a cluster is first created, then there can be the addition of a single view to an existing cluster or the merging of two clusters. The first two are the typical operations of a sequential pipeline, whereas the latter is unique to the hierarchical pipeline.

Each node is upgraded, when possible, to a Euclidean frame. Autocalibration with known skew and aspect ratio requires a minimum of $m = 4$ views to work; when this condition is not met, or in degenerate cases, a quasi-Euclidean upgrade will suffice.

4.1 Two-views reconstruction.

The reconstruction from two views proceeds from the fundamental matrix. It is well known that the following two camera matrices:

$$P_1 = [I \mid \mathbf{0}] \quad \text{and} \quad P_2 = [[\mathbf{e}_2]_\times F \mid \mathbf{e}_2], \tag{5}$$

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

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

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

4.1.1 Triangulation.

Triangulation (or intersection) is performed by the iterated linear LS method [16]. Points are pruned by analyzing the condition number of the linear system and the reprojection error. The first test discards ill-conditioned 3D points, using a threshold on the condition number of the linear system (10^4, in our experiments). The second test applies the so-called X84 rule [14], that establishes that, if e_i are the residuals, the inliers are those points such that

$$|e_i - \text{med}_j\, e_j| < 5.2\, \text{med}_i\, |e_i - \text{med}_j\, e_j|. \tag{6}$$

4.2 One-view addition.

The reconstructed 3D points that are visible in the view to be added provides a set of 3D-2D correspondences, that are exploited to glue the view to the cluster. This can be done by resection with DLT [15], using MSAC [37] to cope with outliers. The view that has been glued might have brought in some new tracks, that are triangulated as described before (Sec. 4.1.1). Finally, bundle adjustment is run on the current reconstruction.

4.3 Clusters merging.

When two clusters merge the respective reconstructions live in two different reference systems, that are related by a a projectivity of the space (which is a similarity when both are properly calibrated). 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. An homography of the projective space is sought that brings the second onto the first, thereby obtaining the correct basis for the second. Once the cameras are registered, the common 3D points are re-computed by triangulation (Sec. 4.1.1), and the tracks obtained after the merging as well. The new reconstruction is eventually refined with bundle adjustment.

5 Autocalibration

To be able to traverse the hierarchical tree without calibration information, we strive to enforce Euclidean structure inside each node. This is of course not always possible, in particular for nodes at the lowest level of the hierarchy, composed by a low number of views (for example, autocalibration with known skew and aspect ratio requires a minimum of 4 views to obtain

a unambiguous solution). For these nodes, a quasi-Euclidean upgrade will suffice until the minimum number of views or a unambiguous configuration is reached.

Our approach is based on a novel method for the estimation of the plane at infinity given an estimate for the internal parameters of at least two cameras. Equipped with such procedure, we can then explore exaustively the space of valid calibration parameters (which is naturally bounded because of the finiteness of acquisition devices) while looking for the best rectifying homography.

The canonical pair of camera matrices

$$P_1 = [I \mid \mathbf{0}] \quad \text{and} \quad P_2 = [Q_2 \mid \mathbf{e}_2], \tag{7}$$

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

$$H = \begin{bmatrix} K_1 & \mathbf{0} \\ \mathbf{v}^\top & 1 \end{bmatrix}. \tag{8}$$

Given reasonable assumptions on internal parameters of the cameras K_1 and K_2, the upgraded, metric versions of the perspective projection matrices are equal to:

$$P_1^E = [K_1 \mid \mathbf{0}] \simeq P_1 H \tag{9}$$

$$P_2^E = K_2 \left[R_2 | \mathbf{t}_2\right] \simeq P_2 H = \left[Q_2 K_1 + \mathbf{e}_2 \mathbf{v}^\top | \mathbf{e}_2\right] \tag{10}$$

The rotation R_2 can therefore be equated to the following:

$$R_2 \simeq K_2^{-1} \left(Q_2 K_1 + \mathbf{e}_2 \mathbf{v}^\top\right) = K_2^{-1} Q_2 K_1 + \mathbf{t}_2 \mathbf{v}^\top \tag{11}$$

in which it is expressed as the sum of a 3 by 3 matrix and a rank 1 term. Let R^* be the rotation such that: $R^* \mathbf{t}_2 = [\|\mathbf{t}_2\| \; 0 \; 0]^\top$. Left multiplying it to Eq. 11 yields:

$$R^* R_2 \simeq \overbrace{R^* \; K_2^{-1} Q_2 K_1}^{W} + [\|\mathbf{t}_2\| \; 0 \; 0]^\top \mathbf{v}^\top \tag{12}$$

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

$$R^* \; R_2 = \begin{bmatrix} \mathbf{w}_1{}^\top + \|\mathbf{t}_2\| \mathbf{v}^\top \\ \mathbf{w}_2{}^\top \\ \mathbf{w}_3{}^\top \end{bmatrix} / \|\mathbf{w}_3\| \tag{13}$$

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

$$\mathbf{v} = \left(\mathbf{w}_2 \times \mathbf{w}_3 / \|\mathbf{w}_3\| - \mathbf{w}_1\right) / \|\mathbf{t}_2\| \tag{14}$$

With the described procedure, we can enumerate through all possible matrices of intrinsics of two cameras K_1 and K_2 checking for the best upgrading homography, which can finally be refined through non-linear optimization.

In order to sample the space of calibration parameters we can safely assume, as customary, 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 length values rather than the image center. Thus, we can iterate just over focal lengths f_1 and f_2 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. The search space is therefore reduced to a bounded region of $\mathbb{R}^2$.

To score each sampled point (f_1, f_2), we consider the aspect ratio, skew and principal point location of the resulting transformed camera matrices and aggregate their respective value into a single cost function:

$$\{f_1, f_2\} = \arg\min_{f_1, f_2} \sum_{\ell=2}^{n} \mathcal{C}^2(K_\ell) \tag{15}$$

where K_ℓ is the intrinsic parameters matrix of the ℓ-th camera after the Euclidean upgrade determined by (f_1, f_2), and

$$\mathcal{C}(K) = \overbrace{w_{sk}|k_{1,2}|}^{\text{skew}} + \overbrace{w_{ar}|k_{1,1} - k_{2,2}|}^{\text{aspect ratio}} + \overbrace{w_{u_o}|k_{1,3}| + w_{v_o}|k_{2,3}|}^{\text{principal point}} \tag{16}$$

where $k_{i,j}$ denotes the entry (i, j) of K and w are suitable weights, computed as in [25]. The first term of (16) takes into account the skew, which is expected to be 0, the second one penalizes cameras with aspect ratio different from 1 and the last two weigh down cameras where the principal point is away from the image centre.

Operatively, self calibration procedure is triggered when any of the internal camera parameters of the reconstructed cameras in the current node after the merge step lie outside the valid parameter space. Along with bundle adjustment, this rule ensure a proper Euclidean framework for all nodes for which the self-calibration problem is well-posed.

6 Photo Consistent Planar Patches

What separates unstructured cloud of points from higher-level renditions of an architectural model is a *semantic gap*, which should be bridged exploiting additional information. When no prior knowledge is assumed or user intervention is not available, bottom-up methods are employed. They start directly from raw three-dimensional data points trying to aggregate them in progressively higher level structures, possibly using also the information coming from the images. In this section we will explain how to extract planar image-consistent planar patches. More details can be found in [27]. Planar patches are a very compact and stable intermediate representation of 3D scenes, as they are a good starting point for a complete automatic reconstruction of surfaces. The method integrates several constraints inside J-linkage[35], a robust algorithm for multiple models fitting. It makes use of information coming both from the 3D structure and the images.

6.1 Overview of the J-linkage algorithm

In this section the J-linkage algorithm will be briefly overviewed. More details can be found in [35].

The method is based on random sampling, like RANSAC. Each minimal sample set (MSS) defines a tentative model. Imagine to build a $N \times M$ matrix (Fig. 6.1) where entry (i, j) is 1 if point i is closer to model j than a threshold ε. Each column of that matrix is the characteristic function of the *consensus set* of a model. Each row is the characteristic function of the *preference set* (PS) of a given point, i.e., indicates which models a points has given consensus to. Points belonging to the same structure will have similar PS, in other words, they will cluster in the conceptual space $\{0, 1\}^M$. This is a consequence of the fact that models generated with random sampling cluster in the hypothesis space around the true models.

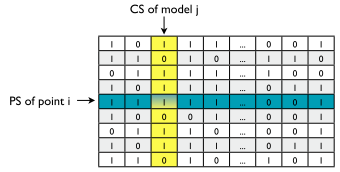


Figure 4: An example of consensus/preference matrix. Columns are consensus sets (CS), rows are preference sets (PS).

Models are extracted by agglomerative clustering data points in the conceptual space, where each point is represented by its PS. The distance between two elements (point or cluster) is computed as the *Jaccard* distance between the respective preference sets. The PS of a cluster is defined as the intersection of the preference sets of its points. Given two sets A and B, the Jaccard distance is

$$d_J(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}. \qquad (17)$$

The Jaccard distance measures the degree of overlap of A and B and ranges from 0 ($A = B$) to 1 ($A \cap B = \emptyset$).

The algorithm proceeds by linking elements with distance smaller than 1 and stops as soon as there are no such elements left. This can be performed efficiently using an heap data structure. As a result, clusters have the following properties:

- for each cluster there exists at least one model that is in the PS of all its points;
- one model cannot be in the PS of *all* the points of two distinct clusters;

The final model parameters for each cluster of points is estimated by least squares fitting.

6.2 Constraints integration

A planar patch associated to a cluster of points is the surface delimited by the convex hull given by the projection of the points onto the fitting plane. Making use of the solely spatial information is not enough for our goal (see Fig. 5). In order for a planar patch to represent an actual surface, it must satisfy a number of constraints, beside coplanarity, that will be described later. This section will concentrate on how these constraints can be seamlessly integrated inside J-linkage.

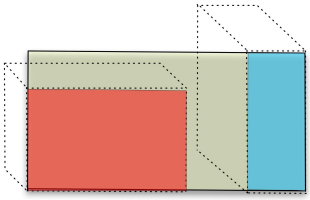


Figure 5: A single plane (yellow) contains several patches (blue and red).

J-linkage extracts models in an incremental way, by merging smaller structures at each step. In the case on planar patches, two patches can merge only if the result is a set of coplanar points (to some extent). The validation of additional constraints can be integrated at this level so that two clusters can be merged if and only if the new cluster does not violate additional rules, i.e., the additional constraints must hold for the planar patch associated to the new cluster.

More in detail, the constraints will be formulated and tested on triangles, since any planar polygon can be triangulated. When two patches are being considered for possible merging, a new patch is computed as the convex hull of the union of the points. By inductive hypothesis the two original patches satisfy the constraints, whereas the new triangles that are created must be tested against the constraints. If a single triangle fails the merging is rejected. A graphical explanation of this step is shown in the Fig. 6.

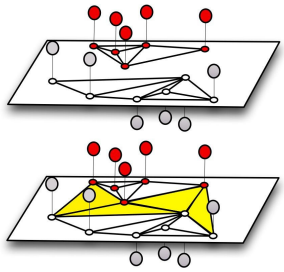


Figure 6: Incremental step. The constraints are assumed to be valid for each patch (top). When two patches are merged (bottom) the constraints needs to be checked only for the new triangles (yellow).

Three kind of constraints are enforced:

- **Photo-Consistency Constraint**: the projections of a triangle on the images where it is visible should be photo-consistent.

- **Visibility Constraint**: a triangle must not to occlude any visible point.
- **Non Intersection Constraint**: a triangle must not intersect any previously defined surface.

6.2.1 Photo-Consistency Constraint

A patch in space is *image-consistent* if all its projections onto the images where it is visible contain conjugate points. Image-consistent patches are attached to actual object surfaces in the scene. Image-consistency can be checked through *photo-consistency*, the property that the projections of a patch are equal up to a projective transformation and photometric differences.

Let us first define a set of compatible images as the ones where the vertices of a given triangle are visible. Among them, the one where the projected triangle exhibits the maximum area is chosen as the reference. All the triangles in the compatible images are projectively warped onto the triangle in the reference image and compared to it through normalized cross-correlation (NCC):

$$\frac{1}{n-1}\sum_{x,y}\frac{(R(x,y)-\overline{R})(C(x,y)-\overline{C})}{\sigma_R\sigma_C}. \qquad (18)$$

where R is the region inside the reference triangle, C is the region inside the triangle in a compatible image, n is the number of pixels inside the triangle, $\overline{x}$ denotes the mean and σ_x denotes the standard deviation. The final photo-consistency of the 3D triangle is obtained as the average of the NCC scores of its projections (the value ranges from -1 to 1), and its is considered photo-consistent if this value is below a fixed threshold.

6.2.2 Visibility Constraint

A Structure and Motion pipeline generally outputs the *visibility* of the points, i.e. the cameras from which a point is visible. This information derives from the initial stage of the pipeline where keypoints are extracted and matched with other keypoints from different images. We can thus formulate a simple yet powerful constraint: a surface patch must not occlude a 3D point from the view where it is visible.

Mathematically, this translates into a segment-triangle intersection test. The segment ranges from the optical center of the view to the 3D point that is being examined. The intersection test can be performed efficiently at constant time. However, in the worst case - i.e. when no intersections with the current triangle were found - one need to run the test for each view and for each visible point from that view. In order to speed up the process, we precompute the axis aligned bounding box (AABB) for each view that contains every visible points and the optical center. We also compute and update an AABB that contains every point of a patch. A prior intersection test is made between the AABB of the patch and the AABB of a view: if no intersection occurs we are assured that no triangle of the patch will intersect a segment in that view. The intersection test between two AABB also takes constant time.

6.2.3 Non Intersection Constraint

During the patch growing, it may happen that patches end up intersecting each other in their interior. It would be desirable to deal with tridimensional meshes that are not self-intersecting. This derives from the fact that surfaces are assumed to be manifolds. We embedded the non intersection constraint directly in the J-linkage.

When creating a new patch we check that it is not intersecting any previously defined patch. This translates into a triangle-triangle intersection test among all the triangles of two patches. The triangle-triangle intersection test can be computed in constant time. However, when dealing with surfaces composed by many triangles, it may require many checks. We thus perform a prior AABB-AABB intersection test: if the bounding boxes do not overlap, we are assured that the surfaces are not intersecting each other and we do not need any further testing.

6.3 Filling the gaps

During the agglomerative clustering of J-linkage, it is sufficient that a single triangle does not satisfy a constraint to discard the entire merge, because it is inductively assumed that patches are *convex*. As a result, triangles that fulfill the constraints are discarded, thereby leaving gaps in the surfaces between neighbouring patches (Fig. 7). This issue is solved *a-posteriori*, by a gap-filing heuristics that relaxes the convexity assumption.

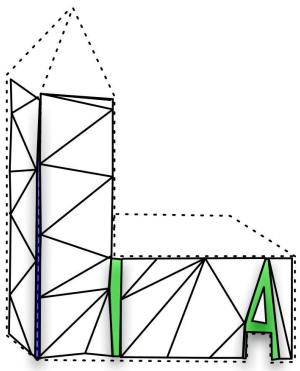


Figure 7: Green regions are gaps between adjacent patches that are to be filled. Blue regions are gaps between orthogonal patches.

Two patches are said to be *adjacent* if at least one of the points of one patch contains a point of the other patch in his k-neighborhood. We can distinguish two cases of adjacent

patches: *coplanar*, when the angle between the respective support planes is less than 30 degrees, and *orthogonal*, when the angle lays between 60 and 120 degrees. A graph of connection between the patches can thus be inferred. First, we fill the gaps between orthogonal patches. By construction, a point can belong to only one patch. We identify the points compatible, by means of the inlier threshold, to both the orthogonal patches. The points are then added to both patches if the constraints defined before are valid for the newly computed patches.

Finally, we fill the gaps between coplanar patches by testing each one connecting triangles between the patches using the same methods and constraints defined before.

7 Results

In this section we will first show the potentialities of our approach by processing a uncalibrated urban dataset composed of pictures collected in the city of Verona, Italy. We will then analyze the accuracy of our reconstructions comparing them to laser scans, taken as ground truth data. Finally we will present the result of the planar patches extraction phase and the resulting compact models.

No comparative results will be presented here; hierarchical framework was already demonstrated to be faster [8] and more precise [12] in previous literature.

7.1 Verona dataset

Our first experiments consisted in processing a dataset of 1129 images composed fusing together five different image collections captured in the city of Verona, Italy, as shown in table 1. The total running time for the cumulative bundle adjustment phase for this dataset was slightly over 6 hours, producing a cloud of over 370000 points. The final results is shown in figure 8 where the obtained point cloud is shown superimposed to a map of the city. The cloud is not composed from a single cluster but is partitioned into the four connected components, corresponding to the overlapping picture groups. Three of them are shown as colored point clouds in figure 8.

Dataset	# images
Piazza Bra	380
Piazza Erbe	333
Dante	40
Via Roma	326
Castevecchio	120
Total	1129

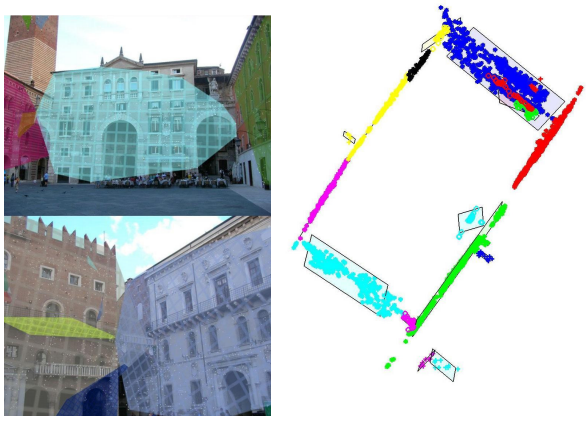
Table 1: Dataset "Verona" was obtained joining five different datasets of sightseeing location in Verona, Italy.

7.2 Duomo dataset

Our second experiment will test the accuracy of our proposal. We will evaluate dataset "Duomo", composed of 309 uncalibrated pictures of the cathedral of Pisa, Italy. Thanks to the availability of ground truth for 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 (fig. 9). The residual distances between closest pairs are measured and their average – the reconstruction accuracy – is about 15cm over an area of more than 70 meters as the crow flies.

7.3 Planar patches models

The first set - "Dante" - is composed of 39 images and 2971 points. The results are shown in Fig. 10. In the second test the subject is a church. The images involved are 54 and the cloud of points is composed of 11094 points. The last test is computationally more challenging. The subject is "Piazza Bra" (Verona). The images are 380 and the points 52024 (obtained by subsampling the original 104047 points). The final extracted patches with our approach are 302. Please note that the boundaries of the patches seldom do not coincide with the actual edges of the façades, because points were detected by SIFT, which tends to keep away from corners. However, these planar patches must be considered only as a initial step toward the extraction of an high-level model. Several heuristics can be deployed to expand the regions up to their natural boundaries. More detailed results are are shown in [27].



(a) Planar Patches. (b) Supporting planes.

Figure 10: "Dante" dataset.

A textured version of our results are shown in Fig. 11.

(a) "Dante". (b) "Pozzoveggiani". (c) "Piazza Bra".

Figure 11: Textured examples.

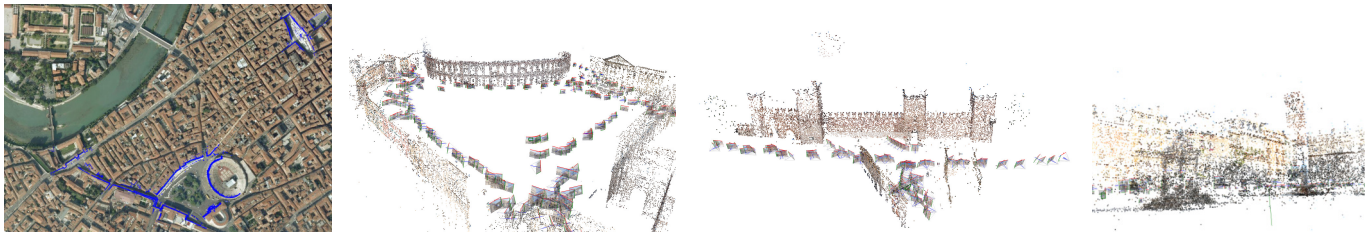


Figure 8: A top view of the reconstruction of dataset "Verona" (Italy). The three perspective views are respectively of the sightseeing locations Piazza Bra, Castevecchio and Piazza Erbe.

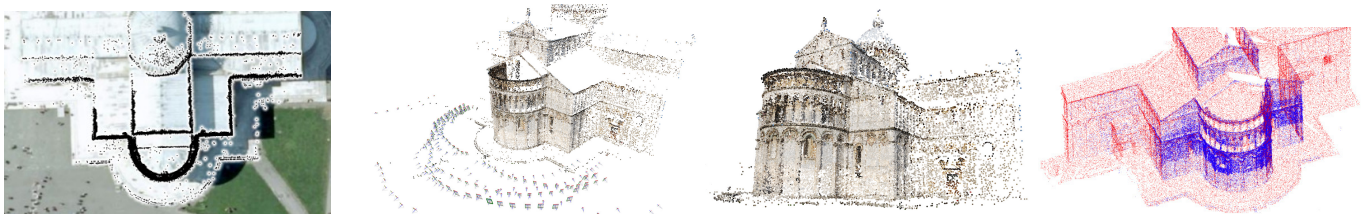


Figure 9: A top view and two perspective views of the reconstruction of "Duomo" (Pisa, Italy). In the last picture the point cloud (blue) is shown superimposed to the ground truth (red).

8 Conclusions

We presented a Structure and Motion pipeline that improves on the state of the art thanks to a hierarchical scheme based on views clustering. Our proposal is more efficient than sequential approaches, boosting the computational efficiency by one order of magnitude, more effective, because it is insensitive to initialization and copes better with drift problems, and more general, because able to process uncalibrated picture datasets.

The point clouds were augmented with planar photo-consistent patches, seamlessly combining information coming from the images and the SaM pipeline. The final result is a very compact and stable intermediate representation, and can be regarded as a starting point for a complete automatic reconstruction of scene surfaces. Future work will aim at bridging further the semantic gap.

Data and additional material are available from from http://profs.sci.univr.it/∼fusiello/demo/samantha/.

Acknowledgements

The laser data of the "Duomo di Pisa" comes from the "Cattedrale Digitale" project (http://vcg.isti.cnr.it/cattedrale), while the photo set is courtesy of the Visual Computing Lab (ISTI-CNR, Pisa). The use of VLFeat by A. Vedaldi and B. Fulkerson, ANN by David M. Mount and Sunil Arya, SBA by M. Lourakis and A. Argyros is gratefully acknowledged.

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] A. Bartoli. A random sampling strategy for piecewise planar scene segmentation. *Computer Vision and Image Understanding*, 105(1):42–59, 2007.

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

[4] 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.

[5] Ondřej Chum, Tomáš Pajdla, and Peter Sturm. The geometric error for homographies. *Computer Vision and Image Understanding*, 97(1):86–102, 2005.

[6] 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.

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

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

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

[10] 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.

[11] Y. Furukawa, B. Curless, S.M. Seitz, R. Szeliski, and R. Szeliski. Reconstructing building interiors from images. In *Proc. Int. Conf. on Computer Vision*, pages 80–87, 2009.

[12] R. Gherardi, M. Farenzena, and A. Fusiello. Improving the efficiency of hierarchical structure-and-motion. In *Proceedings*

of the European Conference on Computer Vision (ECCV), 2010. To appear.

[13] 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.

[14] 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 & Sons, 1986.

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

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

[17] 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.

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

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

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

[21] D.D. Morris and T. Kanade. Image-consistent surface triangulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1. IEEE Computer Society; 1999, 2000.

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] R.Toldo and A.Fusiello. Photo-consistent planar patches from unstructured cloud of points. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010. To appear.

[28] 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.

[29] 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.

[30] Ian Simon, Noah Snavely, , and Steven M. Seitz. Scene summarization for online image collections. In *Proceedings of the Internation Conference on Computer Vision*, 2007.

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

[32] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH: International Conference on Computer Graphics and Interactive Techniques*, pages 835–846, 2006.

[33] 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.

[34] Thorsten Thormählen, Hellward Broszio, and Axel Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In *Proceedings of the European Conference on Computer Vision*, pages 523–535, 2004.

[35] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 537–547. Springer, 2008.

[36] 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.

[37] 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.

[38] Maarten Vergauwen and Luc Van Gool. Web-based 3D reconstruction service. *Machine Vision and Applications*, 17(6):411–426, 2006.

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