Image-consistent patches from unstructured points with J-linkage

Roberto Toldo[1], Andrea Fusiello[2]

Dipartimento di Informatica, Università di Verona,
Strada Le Grazie, 15 - 37134 Verona, Italy

Abstract

Going from unstructured cloud of points to surfaces is a challenging problem. However, as points are produced by a structure-and-motion pipeline, image-consistency is a powerful clue that comes to the rescue. In this paper we present a method for extracting planar patches from an unstructured cloud of points, based on the detection of image-consistent planar patches with J-linkage, a robust algorithm for multiple models fitting. The method integrates several constraints inside J-linkage, optimizes the position of the points with regard to image-consistency and deploys a hierarchical processing scheme that decreases the computational load. With respect to previous work this approach has the advantage of starting from sparse data. Several results show the effectiveness of the proposed approach.

Keywords: robust model fitting, surface reconstruction, plane fitting, 3D modeling

1. Introduction

Although the current state of the art in three-dimensional (3D) reconstruction from images addresses the recovery of dense and accurate models [1, 2, 3], there is still an unfulfilled need for compact, abstract representations of objects.

What separates unstructured cloud of points from higher-level renditions of a model is a *semantic gap*, which could be bridged by leveraging additional information, such as meshing, surfaces, ordering, occlusion, parallelism and orthogonality of structures. Increasing levels of abstraction can then be obtained progressing to recognition of scene elements or entire architectural scenes.

Very recent works based on multiview stereo coupled with a structure-and-motion pipeline [1, 2, 3] produce visually compelling results, but they do not address the semantic gap issue at all, since the output is a dense, non-compact representation of the scene.

Three main approaches can be recognized that aims at bridging the semantic gap: interactive, top-down and bottom-up.

Interactive approaches require user intervention to recognize higher level structures, usually basing on the 3D information previously extracted [4, 5, 6, 7].

Top-down or model-based approaches start from the prior knowledge of the set of potential parametric models and try to infer the best fitting one along with its parameters [8, 9, 10, 11]. Potentially, only one image could be employed if the prior knowledge is enough to derive the 3D model [12, 13].

Bottom-up methods start directly from 3D data points trying to aggregate them in progressively higher-level structures, possibly using the reflectance information coming from the images, namely *image-consistency*. This paper falls in this category: The aim is to leverage models from unorganized point clouds to an intermediate representation made of planar patches as they are a good starting point for a complete automatic reconstruction of surfaces.

Some methods try to optimize an initial triangulation using visibility [14] or image-consistency [15, 16] only. They work only with very simple convex polyhedra objects, and they assume all points visible in at least one view. Others [17, 18] extract the planes underlying the scene using RANSAC (or MSAC [19]) with spatial or image-consistency information. However, the sequential application of an algorithm designed for *single* model extraction is not suitable for large, noisy datasets. In fact, the experiments reported in those papers involve extremely simple objects.

In this paper we describe a general robust technique designed to fit *multiple* model instances, and then specialize it to the task of fitting planar patches to large unorganized point clouds, by integrating in a seamless way image-consistency and visibility constraints. Part of the work reported here has been published in [20, 21].

A releted stream of work is those referred to as "planar stereo" [22, 23, 24], where one aims at obtaining a piecewise planar representation of scene starting from *dense* multiview stereo. In this work, instead, we face a more difficult problem, for we start with the *sparse* output of structure-and-motion, thereby avoiding the multiview stereo stage.

The rest of the paper is organized as follows. Section 2 introduces the J-linkage algorithm, whereas Sec. 3 describes how geometrical and image constraints have been

Email addresses: roberto.toldo@3dflow.net (Roberto Toldo), andrea.fusiello@uniud.it (Andrea Fusiello)

[1]R.T. is now with 3Dflow s.r.l. Verona, Italy

[2]A.F is now with the University of Udine - DIEGM

added to address the patch fitting problem. The post-processing is illustrated in Sec. 4. Section 5 presents the hierarchical processing. Experiments are reported in Sec. 6 and, finally, conclusions are drawn in Sec. 7.

2. J-linkage

A widespread problem in Computer Vision is fitting a model to noisy data: The RANSAC algorithm [25] is the common practice for that task. It works reliably when data contains measurements from a single structure corrupted by gross outliers.

When multiple instances of the same structure are present in the data, the problem becomes tough, as the robust estimator must tolerate both gross outliers and *pseudo-outliers*. The latter are defined in [26] as "outliers to the structure of interest but inliers to a different structure". The difficulty arises because most robust estimators, including RANSAC, are designed to extract a single model. Mode finding in parameter space and Randomized Hough Transform (RHT) [27], on the contrary, copes naturally with multiple structures, but cannot deal with high percentage of gross outliers, especially as the number of models grows and the distribution of inliers per model is uneven.

In the same spirit of [28] we analyze the distribution of residuals of individual data points with respect to the hypotheses (generated by random sampling and fitting) instead of studying the distribution of residuals per each hypothesis. It turns out that the modes of the residuals distribution reflects the model instances, because hypotheses generated with random sampling tend to cluster around the true models – a fact that is also exploited by RHT. However, finding modes ends up to be cumbersome, as proved in our experiments. One reason is that the peak corresponding to a given model becomes less localized as the point-model distance increases. As a result, the rightmost modes in the histogram are usually drowned in the noise.

For this reason we do not work in the residual space as [28], nor in the parameter space, which is at the root of the shortcoming of RHT. We adopt instead a *conceptual representation*: each data point is represented with the characteristic function of the set of models preferred by that point[3]. Multiple models are revealed as clusters in the conceptual space.

In [30] a method based on the related notion of "preference analysis" is proposed, where a point is represented by the permutation that arranges the models in order of ascending residuals. A kernel is then defined, based on this representation, such that in the corresponding RKHS inliers to multiple models and outliers are well separated. This allows remove outliers, then the clean data is over-clustered with kernel-PCA and spectral clustering and the resulting structures are merged with a sequential ad-hoc scheme that incorporates model selection criteria. In [31] this last stage have been refined with kernel optimization. Residual information is also exploited in [32], a single-model estimation technique based on random sampling, where the inlier threshold is not required.

A more classical model selection approach is taken in [33] and [34], where the cost function to be minimized is composed by a data term that measures goodness of fit and a penalty term which weigh model complexity (cfr. GRIC [35]). In both papers the authors focus on sophisticated and effective minimization techniques, but the relative magnitude of the penalty term with reference with the data term is not obvious to define.

J-linkage starts with random sampling: M model hypothesis are generated by drawing M minimal sets of data points necessary to estimate the model, called minimal sample sets (MSS). Then the consensus set (CS) of each model is computed, as in RANSAC. The CS of a model is the set of points such that their distance from the model is less than a threshold ε.

Imagine to build a $N \times M$ matrix where entry (i, j) is 1 if point i belongs to the CS of model j, 0 otherwise. Each column of that matrix is the characteristic function of the CS of a model hypothesis. Each row indicates which models a points has given consensus to, i.e., which models it prefers. We call this the *preference set* (PS) of a point. Figure 1 shows an example of such a matrix in a concrete case.

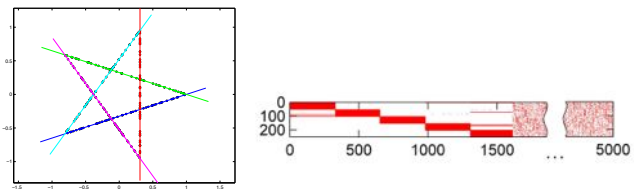


Figure 1: Right: the data consist of 250 points on five segments forming a star. Left: Preference matrix. The rows are points (ordered by cluster), the columns are models (ordered by cluster size)

The characteristic function of the preference set of a point can be regarded as a conceptual representation of that point. Points belonging to the same structure will have similar conceptual representations, in other words, they will cluster in the *conceptual space* $\{0, 1\}^M$. This is, again, a consequence of the fact that models generated with random sampling cluster in the hypothesis space around the true models.

2.1. Random sampling

Minimal sample sets are constructed in a way that neighboring points are selected with higher probability, as suggested in [36, 37]. Namely, if a point $\mathbf{x}_i$ has already

[3] According to [29] the posterior probabilities of an object x given C classes form a *similarity conceptual representation*: $[P(x|\text{ class } 1) \cdots P(x|\text{ class } C)]$.

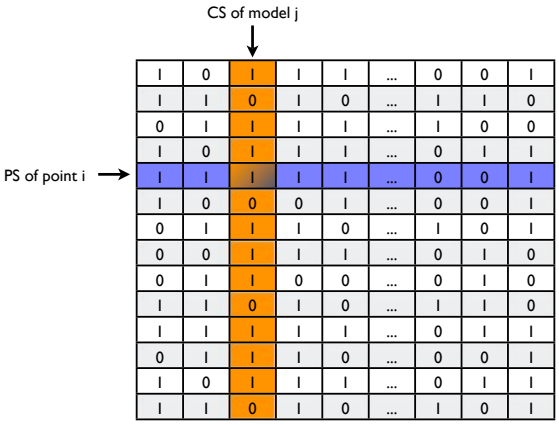


Figure 2: An example of consensus/preference matrix. Columns are consensus sets (CS), rows are preference sets (PS).

been selected, then $\mathbf{x}_j$ has the following probability of being drawn:

$$P(\mathbf{x}_j|\mathbf{x}_i) = \begin{cases} \frac{1}{Z}\exp -\frac{||\mathbf{x}_j-\mathbf{x}_i||^2}{\sigma^2} & \text{if } \mathbf{x}_j \neq \mathbf{x}_i \\ 0 & \text{if } \mathbf{x}_j = \mathbf{x}_i \end{cases} \quad (1)$$

where Z is a normalization constant and σ is chosen heuristically (set to two times the inlier threshold in our experiments).

Then for each points its preference set is computed, as the set of models such that the distance from the point is less than the inlier threshold ε (same as RANSAC).

The number M of MSS to be drawn is related to the percentage of outlier and must be large enough so that a certain number (at least) of outlier-free MSS are obtained with a given probability for all the models. Please note that if this condition is verified for the model with less inliers, it is automatically verified for all the other models.

Let S be the number of inliers for a given model and N be the total number of points. The probability of drawing a MSS of cardinality d composed only of inliers is given by:

$$p = P(E_1)P(E_2|E_1)\dots P(E_d|E_1, E_2 \dots E_{d-i}) \quad (2)$$

where E_i is the event "extract an inlier at the i-th drawing". In the case of uniform sampling $P(E_i|E_1, E_2 \dots E_{i-1}) = \frac{S-i+1}{N-i+1}$. In our case, the first point is sampled with uniform probability, hence $P(E_1) = S/N$, while the others are sampled with the probability function (1), therefore, after expanding the normalization constant Z, the conditional probability can be approximated as

$$P(E_i|E_1, E_2 \dots E_{i-1}) = \frac{(S-i+1)e^{-\alpha^2/\sigma^2}}{(N-S-i+1)e^{-\omega^2/\sigma^2} + (S-i+1)e^{-\alpha^2/\sigma^2}} \quad i = 2 \dots d \quad (3)$$

where α is the average inlier-inlier distance, and ω is the average inlier-outlier distance. If $S \gg d$ then

$$p \simeq \delta \left(\frac{\delta \exp -\frac{\alpha^2}{\sigma^2}}{(1-\delta)\exp -\frac{\omega^2}{\sigma^2} + \delta \exp -\frac{\alpha^2}{\sigma^2}} \right)^{d-1}. \quad (4)$$

where $\delta = S/N$ is the inlier fraction for a given model. Therefore, assuming that ω is larger than α, the sampling strategy increases the probability of extracting an outlier-free MSS, as the intuition would also suggests.

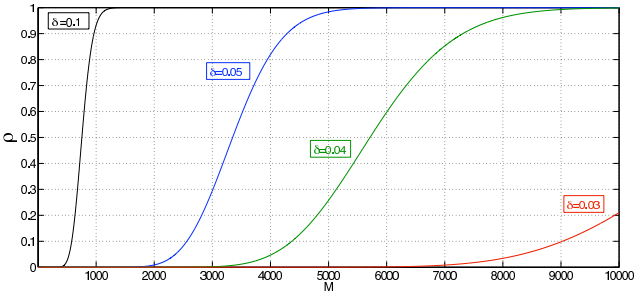


Figure 3: Plot of ρ vs M for different values of δ with $d = 3$, $K = 25, \alpha = \sqrt{0.5}\,\sigma, \omega = \sqrt{3.0}\,\sigma$.

Finally, the probability of drawing at least K outlier-free MSS out of M, for a given model, is given by [28]:

$$\rho = 1 - \sum_{k=0}^{K-1} \binom{M}{k} p^k (1-p)^{M-k}. \quad (5)$$

This equation is used to compute the required number of samples M for a given confidence ρ and a given K. Values of ρ vs M are shown in Figure 3. The value of δ in (4) must be set to the smallest inliers fraction among all the models. This is likely to lead to a pessimistic assumption, which translates in an overestimation of M.

The benefit of local sampling can be appreciated in Figure 4, where, in the case of a single model estimation, it is clear that localized sampling avchieves the same ρ with a smaller M than the classical uniform sampling.

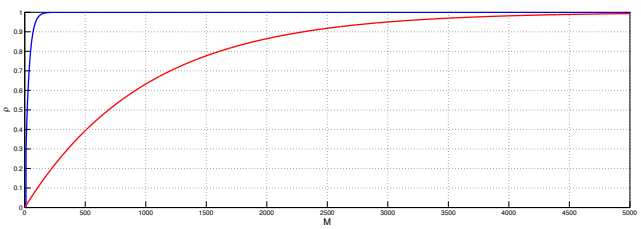


Figure 4: Plot of ρ vs M for uniform sampling with $\delta = 0.1$, $d = 3$ (blue line) and for localized sampling (red line) with ρ vs M for $\delta = 0.1$ with $d = 3$, $K = 1, \alpha = \sqrt{0.5}\,\sigma, \omega = \sqrt{3.0}\,\sigma$.

2.2. J-linkage clustering

Models are extracted by agglomerative clustering of data points in the conceptual space, where each point is represented by (the characteristic function of) its preference set.

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 [38].

We propose a variation that fits very well to our problem, called *J-linkage* (see Algorithm 1). First the preference set of a cluster is computed as the *intersection* of the preference sets of its points. Then the distance between two elements (point or cluster) is computed as the *Jaccard distance* between the respective preference sets. Given two sets A and B, the Jaccard distance is

$$d_{\mathrm{J}}(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|}.$$

The Jaccard distance measures the degree of overlap of the two sets and ranges from 0 (identical sets) to 1 (disjoint sets).

The cut-off value is set to 1, which means that the algorithm will only link together elements whose preference sets overlap. Please note that the cut-off distance is not data dependent, but defines a characteristics behavior of the J-linkage algorithm. Indeed, as a result, clusters of points have the following properties:

- for each cluster there exist at least one models that is in the PS of all the points (i.e., a model that fits all the points of the cluster)
- one model cannot be in the PS of *all* the points of two distinct clusters (otherwise they would have been linked).

Each cluster of points defines (at least) one model. If more models fit all the points of a cluster they must be very similar. The final model for each cluster of points is estimated by least squares fitting.

The algorithm can be summarize as follows:

Algorithm 1 J-LINKAGE

Input: the set of data points, each point represented by its preference set (PS)
Output: clusters of points belonging to the same model

1. Put each point in its own cluster.
2. Define the PS of a cluster as the *intersection* of the PSs of its points.
3. Among all current clusters, pick the two clusters with the smallest Jaccard distance between the respective PSs.
4. Replace these two clusters with the union of the two original ones.
5. Repeat from step 3 while the smallest Jaccard distance is lower than 1.

Outliers ends up in small clusters: Let us order the clusters by cardinality. Depending on the application, one may set different rejection thresholds:

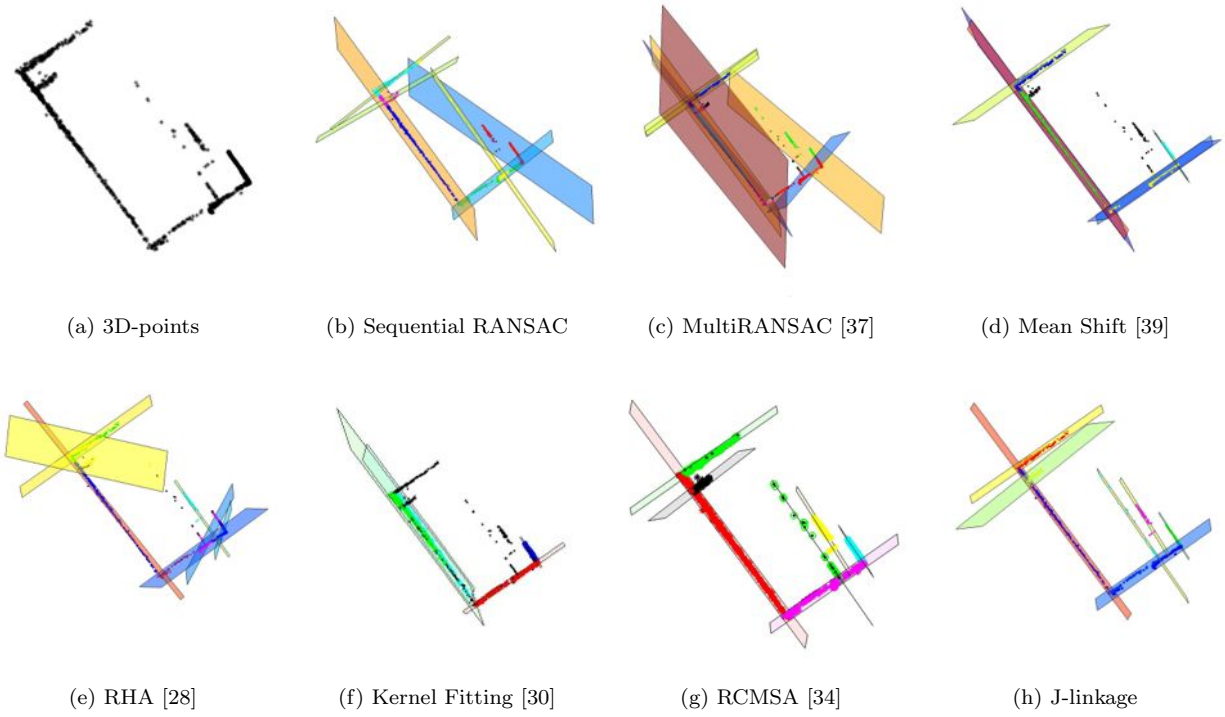
- If the percentage of outliers is known or can be estimated (as it is assumed in RANSAC), one may reject all the smallest clusters up to the number of outliers.
- If the models are know to have *almost* the same cardinality, one may find the point where the cardinality of clusters drops, and reject below that point.
- If the number k of models is known, one may keep the largest k clusters.

J-linkage is a general tool for fitting multiple model instances to data corrupted by outliers. With respect to similar competing methods like [30, 31, 34], J-linkage has the drawback of requiring the inlier threshols, as RANSAC, and some additional knowledge or processing is needed to determine the number of models. However, it must be noted that:

- The inlier threshold is usually an educated guess, and however some work has been done in the direction of its automatic estimation [40];
- Although the number of models is not estimated by J-linkage, this information is not involved in the processing. In other words, J-linkage does not need this piece of information to produce its results, which can then be refined with an educated guess or with an automatic model selection procedure, as in [31].

Moreover, in J-Linkage is easier to introduce additional domain-dependent constraints in the aggregation stage, as will be discussed in Section 3.

These remarks concur to define J-linkage as a basic building block, whose simplicity is its more remarkable feature, that can be extended along several directions [41],



(a) 3D-points (b) Sequential RANSAC (c) MultiRANSAC [37] (d) Mean Shift [39]

(e) RHA [28] (f) Kernel Fitting [30] (g) RCMSA [34] (h) J-linkage

Figure 5: "Pozzoveggiani" dataset. 3D planes extracted by J-linkage and other algorithms, viewed from the zenith.

completed with other modules, and inserted in meta-schemata [40]. It is however out of the scope of this paper to describe these developments.

Our motivation for proposing J-linkage derives from the problem of fitting planes to 3D points, therefore, as an example, Figure 5 shows the results of fitting planes to a cloud of 3D points with J-linkage and some competing methods (references in the figure captions). The 3D points shown in Figure 5(a) have been produced by a structure-and-motion pipeline [42] fed with a set of images of the church of Pozzoveggiani (Italy). Despite the fact that gross outliers are absent, pseudo-outliers and the uneven distribution of points among the models (ranging from 9 to 1692) challenges any model fitting algorithm.

As a matter of fact J-linkage and RCMSA are the only ones that produces the correct result, although RCMSA required to guess the correct scale of penalty term (set to 1000) by trial and error. On the contrary, the knowledge that the data were almost outliers-free let us select all cluster of cardinality greater than three in J-linkage.

The bad performance of Kernel Fitting on this example (although it must be said it is the only method that was not given any additional information) is due to the absence of outliers, as pointed out also in [41].

A more systematic experimental validation of J-linkage is reported in [20, 41].

3. Constraints integration

Fitting planes, however, does not solve the problem of exacting planar patches, for a patch is a *region* of the plane, and the same plane may contain more patches (see Figure 6).

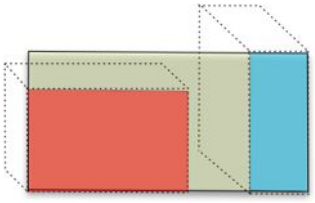


Figure 6: A single plane may contain several patches (blue and red).

This section describes how to leverage the J-linkage algorithm to fit planar patches to a cloud of 3D points that are considered as samples of actual surfaces in the observed scene.

The planar patch associated to a set of coplanar points is the *convex hull* of the projection of the points onto the supporting plane[4]. In order for a 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.

J-linkage extracts models in an incremental way, by merging smaller structures at each step. In the case of planes, two clusters can merge only if the result is a set

[4]According to this definition patches are convex. This requirement will be relaxed in the post-processing step (see Sec. 4)

of coplanar points (within the inlier threshold ε): Coplanarity is the invariant property for plane fitting.

In the case of planar patches, other constraints can be enforced as invariant properties, so that two patches can be merged if and only if the resulting patch does not violate these constraints. When two patches are being considered for possible merging, a new tentative patch is computed as the convex hull of the union of the points. Consider the triangulation of this convex hull: by the inductive hypothesis the triangles belonging to the two original patches satisfy the constraints, whereas the other triangles belonging to the "seam" between the two patches must be tested against the constraints. The "seam" triangles are defined as those belonging to the convex hull of the union of the two patches but not the union of the two patches.

If a single triangle fails the merging is rejected. A graphical explanation of this incremental step is shown in the Figure 7.

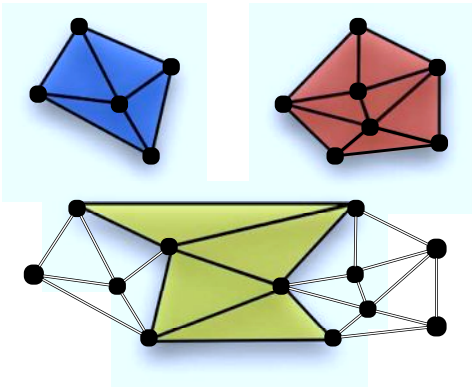


Figure 7: Incremental step. Top: the two patches that are to be merged (blue and red). Bottom: the "seam" triangles (in yellow) that need to be checked against constraints violation.

More in detail, two constraints are considered:

- **Visibility**: a triangle must not to occlude any visible point (See. Sec. 3.1).
- **Image-Consistency**: the projections of a triangle onto the images where it is visible must consist of conjugate points (See. Sec. 3.2).

Sometimes the image-consistency test fails because of small imprecision in the localization of 3D points. In this case, adjusting the 3D points so that to optimize photo-consistency (See. Sec. 3.3) of the region around them could cure the problem, beside improving the overall quality of the 3D reconstruction.

3.1. Visibility Constraint

A structure-and-motion pipeline typically produces the *visibility* of each point $V(P)$, i.e. the views from which point P is visible. This information can be exploited to formulate a simple yet powerful constraint: a triangle must not occlude a 3D point from the view where it is visible.

Mathematically, this translates into a segment-triangle intersection test between the triangle and the line segments joining P and every camera in $V(P)$. The intersection test can be performed efficiently at constant time.

This test must be performed 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.

Please note that in the presence of outliers, valid triangles can be discarded. However, structure-and-motion pipelines are very restrictive about including rogue points, and also discarding clusters with cardinality smaller or equal to three helps cleaning the results.

3.2. Photo-Consistency Constraint

A patch in space is *image-consistent* if all its projections onto the images where it is visible consist of conjugate points. Image consistent patches are attached to actual object surfaces in the scene (see Figure 8). 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 nuances.

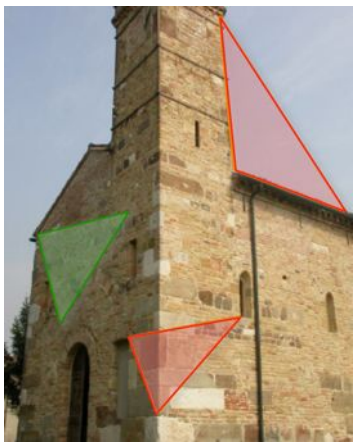


Figure 8: The green triangle is image-consistent, the red ones are not, because they are not part of an actual surface.

Let us consider $V(\tau)$, the set of images where the vertices of a given triangle τ are visible Among them, the one where the projected triangle exhibits the maximum area is chosen as the reference. All the triangles in $V(\tau)$ are projectively warped onto the triangle in the reference image and compared to it through normalized cross-correlation (NCC). 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 (set to 0 in our experiments).

3.3. Photo-consistent adjustment

Points coming from a structure-and-motion pipeline derive their position from triangulation and bundle adjustment based on correspondences among image keypoints. After these keypoints (SIFT, in the case at issue) have been extracted in the early stage of the pipeline, the photometric information is not taken into account any more, as all the processing is purely geometric. Therefore, the photo-consistency of those points might be less than optimal. This is the rationale for the photo-consistent adjustment (or *photo-adjustment*) step that will be described.

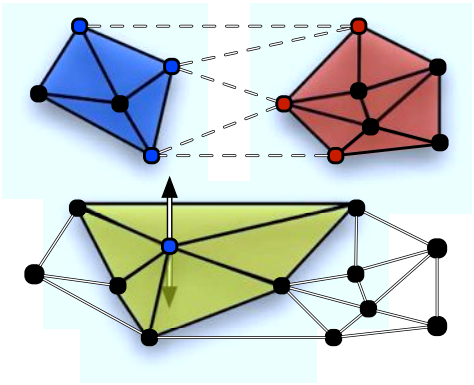


Figure 9: Photo-adjustment. Top: The points on the boundary of two patches that are to be merged (red and blue respectively) are those that needs to be photo-adjusted before the constraints check. Bottom: A point gets displaced along its normal using the fan of triangles (yellow) around it.

The photo-consistency of a point P with fan[5] $\mathcal{F}$ is defined as:

$$\phi(P,\mathcal{F}) = \frac{1}{|V(\mathcal{F})|-1} \sum_{i \in V(\mathcal{F}) \setminus r} \mathrm{ncc}(\Pi_r(\mathcal{F}), T_{i\to r}(\Pi_i(\mathcal{F}))) \quad (6)$$

where $V(\mathcal{F})$ is the set of images where $\mathcal{F}$ is visible, r is a reference image (we choose the one where the area of the projection of $\mathcal{F}$ is larger), $T_{i\to r}$ is the projectivity mapping the plane passing through $\mathcal{F}$ from image i to image r, Π_i is the operator that projects onto view i, and ncc is the normalized cross correlation.

The point P is adjusted by moving along the normal $\mathbf{n}$ to the plane it belongs to, so as to maximize photo-consistency:

$$\max_{d \in [d-\theta,\, d+\theta]} \phi(P + \mathbf{n}d, \mathcal{F}) \quad (7)$$

[5] A *fan* is a set of connected triangles that share one vertex

It is customary in surface multiresolution decomposition (cfr. [43]) to consider a base surface and a normal displacement vector. The tangential component can be taken into account by changing the point on the base surface.

Photo-adjustment takes place *during* the J-linkage clustering: when two patches are going to be merged, for every new triangle that is instantiated its vertices are photo-adjusted (see Figure 9). This guarantees that a point is photo-adjusted before the photo-consistency of the triangle it belongs to is checked.

The algorithm can now be summarized as follows:

Algorithm 2 Convex Planar Patches

Input: cloud of 3D points
Output: clusters of points belonging to the same convex planar patch

1. Compute PS of points with plane models;
2. Put each point in its own cluster;
3. Let C_1 and C_2 the two clusters with the smallest Jaccard distance between the respective PSs;
4. Compute the convex hull of $C_1 \cup C_2$, and identify the set of "seam" triangles S;
5. Do photo-adjustment on vertices of S;
6. If visibility and image-consistency constraints are satisfied by every triangle in S, replace C_1 and C_2 with $C_1 \cup C_2$.
7. Repeat from step 3 while the smallest Jaccard distance is lower than 1.

In our tests we observed that the ratio of triangles that would have failed photo-consistency without the adjustment to the total number of triangles, ranges from 1.8% to 3.5%.

4. Post-processing

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 neighboring patches. Gaps arises also between non-coplanar patches because in J-linkage a point can belong to only one patch (or plane): as a result non-coplanar triangles cannot share a common edge (Figure 10). This issues are solved *a-posteriori*, by a gap-filling heuristics that relaxes the convexity assumption and the uniqueness of point assignment.

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 its k-neighborhood (we used $k = 10$ in our experiments).

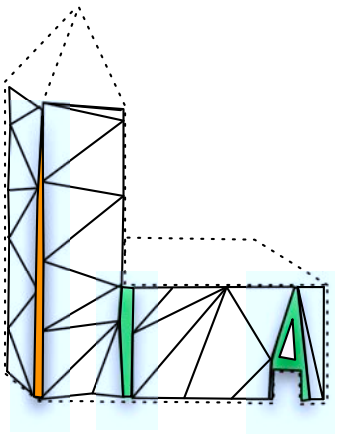


Figure 10: Green regions are gaps between adjacent patches that are to be filled. Blue regions are gaps between orthogonal patches.

Adjacent patches can be *quasi-coplanar*, if the angle between the respective support planes is less than 30 degrees, and *quasi-orthogonal*, if the angle lies between 60 and 120 degrees.

First, the algorithm extracts clusters of quasi-coplanar and mutually adjacent patches, with agglomerative clustering using the angle as distance. A patch can be added to a cluster if it is quasi-coplanar to all the patches of the cluster and adjacent to at least one. Eventually, a 2D Delaunay triangulation of the support plane of the whole cluster is run and the new triangles that are thereby created – which cover the regions that connects different patches – are tested against photo-consistency and visibility, as inside the J-linkage (see Figure 11.) The resulting clusters are the new planar patches, which are possibly larger than the original and non-convex.

Second, the gaps between quasi-orthogonal patches are filled. The algorithm first identifies points compatible with two quasi-orthogonal patches, using the inlier threshold ε, then it tries to add these points to the two patches, checking all the appropriate constraints.

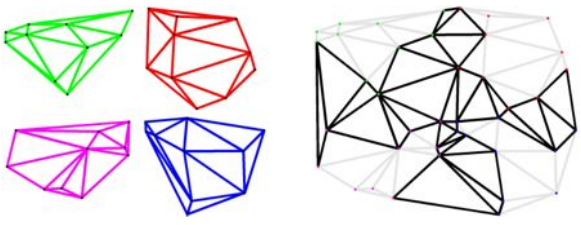


Figure 11: Merge of several quasi-coplanar patches into one new patch. The black triangles on the left are those tested against photo-consistency, visibility and non intersection. If any of them fails the patch becomes non convex.

Finally, the resulting patches are cleaned using a procedure that resembles the morphological opening (erosion followed by dilation). First, in the "erosion" step, triangles that have one or zero neighbor are deleted, together with those having an angle smaller than 10 degrees ("skinny" triangles). Then, in the "dilation step", holes, are filled by triangulation.

Figure 12 shows the application of the opening procedure on a mesh portion: isolated and skinny triangles are removed and small holes are filled.

Figure 13 shows the overall effect of the post processing step. The gaps between coplanar and orthogonal patches are filled with photo-consistent triangles.

The effect of the post-processing is to fill gaps, simplify the results and lighten over-segmentation in a general and problem-independent way. Other obvious problem-driven heuristics could have been implemented in this stage.

5. Hierarchical processing

In this section we leverage the hierarchical partitioning of data (camera and points) provided by Samantha [44] to obtain a hierarchical patch fitting procedure which is more computationally efficient, inherently parallel and suitable for out-of-core processing. Samantha is a structure-and-motion pipeline that first runs an agglomerative clustering on the set of images and then processes them following the dendrogram. As a result, a hierarchy of contained cluster of points is created, where the final reconstruction rests at the root.

Therefore, instead of processing all the points at the same time, the algorithm presented in the previous section is run on partial reconstructions and then the results are merged at the father node. In particular, in every node Samantha can perform two operations: (a) add new points to an existing reconstruction (possibly empty) and/or (b) merge two reconstructions.

In case (a) the planar patch fitting algorithm is run on new points and the resulting patches are appended; in case (b) the two set of patches are joined and common points are assigned to the biggest planar patch. As for step a), in order to link the patches the fit the partial reconstruction with the new points, the support planes of the existing patches are forced into the hypothesis pool of J-linkage. Moreover, we implemented a lazy update strategy, i.e., the processing is triggered only when a certain number of points k are waiting to be added.

The hypothesis generation is performed at each merging step using only the k novel points. Experimentally, we found that 1500 is a good and conservative value for the number M of hypothesis to be generated. This can be obtained from Equation 4 with $\rho = 0.999$, $\delta = 0.1$, $d = 3$, $K = 25, \alpha = \sqrt{0.5}\,\sigma, \omega = \sqrt{3.0}\,\sigma$.

Figure 14 compares the running time of this method (without photo-adjustment) with the sequential approach (as implemented in [21]): It clearly appears that the speed up gained by this approach is remarkable.

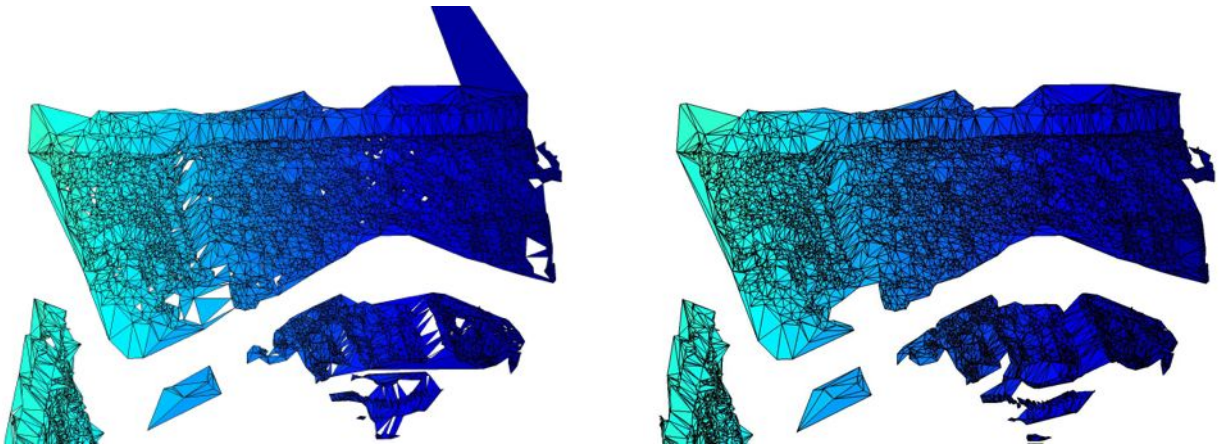


Figure 12: Example of the effect of the "opening" procedure. Input mesh on the left, output mesh on the right.

5.1. Computational complexity

This processing scheme follows a tree, so the total complexity is reduced, with respect to the straightforward sequential approach, which takes $O(n^2 \log n)$ time. In this case, we solve n/k instances of the original problem at the leaves of the tree, with a cost of $n/k\ O(k^2 \log k)$, plus the cost of the internal nodes (merge) which is linear in the number of points to be merged: $n + 2n/2 + 4n/4 + ...hn/h = nh$, where $h = \log(n/k)$ is the height of the tree. Summing up, the time complexity is $n/k\ O(k^2 \log k) + n \log(n/k)$, which is equal to $n \log(n)$ asymptotically, being k constant. Please note, however, that there is a concrete advantage as soon as $k \ll n$.

6. Results

Experiments were performed on publicly available data and other lab-made datasets. The 3D points were produced by Samantha [44], together with the hierarchical partitioning of the data.

Four datasets, namely "Piazza Brà[6]", "Campidoglio", "Duomo[6]" and "Castle-P30[7]" have an architectural scale, whereas "Bas-relief" and "Small sculpture" represent smaller objects captured closer by. The latter, in particular, is not piecewise planar and has a complex topology, compared to the others.

The output produced by our method is shown in Figure 15, Figure 16, and Figure 17. It can be noticed that, besides some minor imprecisions, a meaningful triangulation is extracted from every dataset. On "CastleP-30" and "Campidoglio", the scene is mostly composed by planar regions, and our approach is able to detect them correctly, despite the sparsity of the points in some spots. The "Duomo" dataset is the most challenging among architectural ones. Some minor imprecisions are noticeable in the semicircular apse, but overall the method is able to decompose it in piecewise planar patches, that could be aggregated in higher level primitives by further processing. "Bas-relief" was a fairly easy job, considering the fact that the object is planar and the points cloud is very dense, due to the high resolution (2808x1972) of the images. Finally, in "Small sculpture", although the object is non-planar, our method was able to decompose the surface into piecewise planar regions. Some minor imprecisions are present between these regions, since in most cases they are neither orthogonal nor coplanar.

A more comprehensive visualization of the results can be achieved by watching the videos available on the web[8]. Running times on entry level PC with a single core 2.4Ghz are reported in Table 1.

The reader might notice that the time required to process *Bas-reliefs* is larger than those required to process *Piazza Brà* even though the former has less points (this corresponds to the kink in the line at the bottom of Figure 14). This can be explained by keeping in mind that time complexity does not depend only on the number of points, but also on other variables such as the number of cameras, their overlap and the image resolution (which impacts on the photo consistency step). In this particular case, *Bas-reliefs* has higher resolution images than *Piazza Brà* with a nearly complete visibility graph (every image overlaps almost every other).

Finally, thanks to the availability of ground truth for "Duomo" and "Piazza Brà" datasets, we have been able to assess the improvement in accuracy brought in by the photo-adjustment. The laser data had been subsampled 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 average residual distances between closest pairs was taken as reconstruction accuracy. After photo-adjustment the figure improves by about 4%.

[6] www.diegm.uniud.it/fusiello/demo/samantha/
[7] cvlab.epfl.ch/~strecha/multiview/denseMVS.html
[8] www.diegm.uniud.it/fusiello/demo/jlk/

Figure 13: Some examples of problems cured by post processing (left-before, right-after). The triangles are colored (in transparency) according to the patch they belong to.

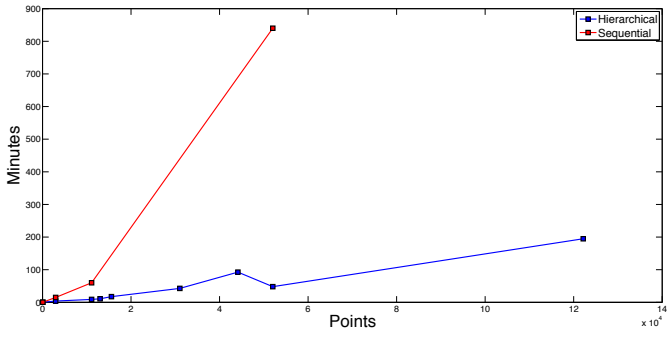


Figure 14: Running times of our method compared with [21]. The blue line plots the same data reported in Tab. 1.

Table 1: Running times.

Dataset	# points	# images	Time [min]
Duplo	72	5	0.3
Dante	2971	39	4.2
Pozzo Veggiani	11094	54	8.9
Small sculpture	12994	58	10.8
Campidoglio	15571	51	17.5
CastleP-30	31030	30	42.8
Bas-relief	44147	45	92.5
Piazza Brà	52024	380	48.0
Duomo	122159	309	194.8

7. Discussion

We presented a method for extracting a triangle mesh from an unstructured cloud of points, based on the detection of image-consistent planar patches with J-linkage. Our approach copes with *sparse* and *real* data, optimizes the position of the points with regard to image-consistency (photo-adjustment) and follows a hierarchical processing scheme that cuts the computing time from $O(n^2 \log n)$ to $O(n \log n)$. Moreover, we proposed a new post processing step that improves the quality of the final mesh. These results, obtained with no manual intervention, are a good starting point for further processing, that could include the user in the loop.

Acknowledgments

This research have been supported by the University of Verona and Gexcel s.r.l. under the "Joint Projects" scheme. The laser scanning of "Piazza Brà" have been conducted by Gexcel s.r.l. with the EU JRC - Ispra and the permission of the municipality of Verona. The laser data of the "Duomo di Pisa" comes from the "Cattedrale Digitale[9]" project, while the photo set is courtesy of the Visual Computing Lab (ISTI-CNR, Pisa).

[9] vcg.isti.cnr.it/cattedrale

References

[1] M. Goesele, N. Snavely, B. Curless, H. Hoppe, S. M. Seitz, Multi-view stereo for community photo collections, in: Proceedings of the International Conference on Computer Vision, 2007.

[2] Y. Furukawa, B. Curless, S. Seitz, R. Szeliski, R. Szeliski, Towards Internet-scale Multi-view Stereo, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, p. 12.

[3] P. F. C. Strecha, T. Pylvanainen, Dynamic and Scalable Large Scale Image Reconstruction, in: Proceedings of 23rd IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[4] C. Taylor, P. Debevec, J. Malik, Reconstructing polyhedral models of architectural scenes from photographs, Lecture Notes in Computer Science 1065 (1996) 659–670.

[5] R. Cipolla, D. Robertson, E. Boyer, Photobuilder–3d models of architectural scenes from uncalibrated images, in: IEEE Int. Conf. on Multimedia Computing and Systems, Vol. 1, 1999, pp. 25–31.

[6] M. Wilczkowiak, P. Sturm, E. Boyer, Using geometric constraints through parallelepipeds for calibration and 3D modeling, IEEE transactions on pattern analysis and machine intelligence (2005) 194–207.

[7] A. van den Hengel, A. Dick, T. Thormählen, B. Ward, P. Torr, VideoTrace: rapid interactive scene modelling from video, in: Proceedings of the SIGGRAPH conference, Vol. 26, 2007.

[8] K. Schindler, J. Bauer, A model-based method for building reconstruction, in: IEEE International Workshop on Higher-Level Knowledge in 3D Modeling and Motion Analysis., 2003, pp. 74–82.

[9] A. Dick, P. Torr, R. Cipolla, Modelling and interpretation of architecture from several images, International Journal of Computer Vision 60 (2) (2004) 111–134.

[10] C. Brenner, N. Ripperda, Extraction of facades using rjM-CMC and constraint equations, Photogrammetric Computer Vision (2006) 155–160.

[11] Y. Tiana, Q. Zhub, M. Gerkea, G. Vosselmana, Knowledge-Based Topological Reconstruction for Building Façade Surface Patches, in: 3D Virtual Reconstruction and Visualization of Complex Architectures (3D-ARCH), Vol. 18, 2009.

[12] F. Han, S. Zhu, Bayesian reconstruction of 3d shapes and scenes from a single image, in: Workshop on High Level Knowledge in 3D Modeling and Motion, Vol. 2, 2003.

[13] P. Muller, G. Zeng, P. Wonka, L. Van Gool, Image-based procedural modeling of facades, ACM Transactions on Graphics 26 (3) (2007) 85.

[14] A. Hilton, Scene modelling from sparse 3D data, Image and Vision Computing 23 (10) (2005) 900–920.

[15] D. Morris, T. Kanade, Image-consistent surface triangulation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Vol. 1, 2000.

[16] A. Nakatsuji, Y. Sugaya, K. Kanatani, Optimizing a triangular mesh for shape reconstruction from images, IEICE Transactions on Information and Systems (2005) 2269–2276.

[17] O. Cooper, N. Campbell, D. Gibson, Automatic augmentation and meshing of sparse 3D scene structure, in: Seventh IEEE Workshops on Application of Computer Vision., Vol. 1, 2005.

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

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

[20] R. Toldo, A. Fusiello, Robust multiple structures estimation with J-linkage, in: Proceedings of the European Conference on Computer Vision, 2008, pp. 537–547.

[21] R. Toldo, A. Fusiello, Photo-consistent planar patches from unstructured cloud of points, in: Proceedings of the European Conference on Computer Vision, 2010, pp. 589–602.

[22] A.-L. Chauve, P. Labatut, J.-P. Pons, Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured

Figure 15: Screenshots of the results: points belonging to different patches are shown in different colors (same color is used more than once when there is no ambiguity). From left to right: "Small sculpture","Campidoglio", "CastleP-30", "Piazza Brà", "Bas-relief", and "Duomo".

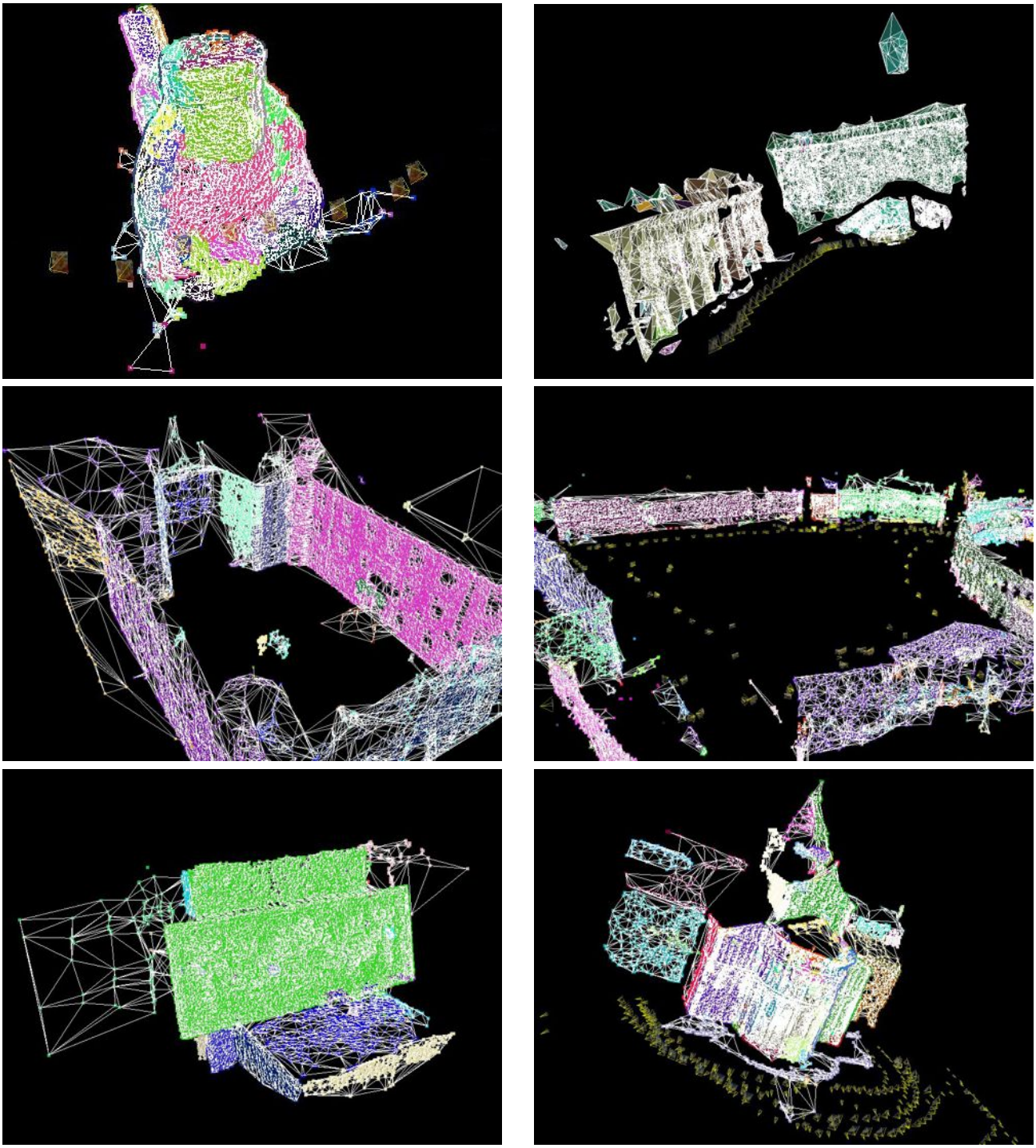


Figure 16: Screenshots of the results shown as triangle meshes. Triangles are colored according to the patch they belong to. From left to right: “Small sculpture”,“Campidoglio”, “CastleP-30”, “Piazza Brà”, “Bas-relief”, and “Duomo”.

Figure 17: Screenshots of the results. The colored triangle mesh (as in Fig. 16) is shown projected onto one of the images.

point data, in: Proceedings of 23rd IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1261 –1268.
[23] C. Hane, C. Zach, B. Zeisl, M. Pollefeys, A patch prior for dense 3d reconstruction in man-made environments, in: Second Joint 3DIM/3DPVT Conference: 3D Imaging, Modeling, Processing, Visualization and Transmission, IEEE, 2012, pp. 563–570.
[24] D. Gallup, J.-M. Frahm, M. Pollefeys, Piecewise planar and non-planar stereo for urban scene reconstruction., in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1418–1425.
[25] M. A. Fischler, R. C. Bolles, Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography, Communications of the ACM 24 (6) (1981) 381–395.
[26] C. V. Stewart, Bias in robust estimation caused by discontinuities and multiple structures, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (8) (1997) 818–833.
[27] L. Xu, E. Oja, P. Kultanen, A new curve detection method: randomized Hough transform (RHT), Pattern Recognition Letters 11 (5) (1990) 331–338.
[28] W. Zhang, J. Kosecká, Nonparametric estimation of multiple structures with outliers, in: R. Vidal, A. Heyden, Y. Ma (Eds.), Workshop on Dynamic Vision, European Conference on Computer Vision 2006, Vol. 4358 of Lecture Notes in Computer Science, Springer, 2006, pp. 60–74.
[29] R. Duin, E. Pekalska, P. Paclik, D. Tax, The dissimilarity representation, a basis for domain based pattern recognition?, in: L. Goldfarb (Ed.), Pattern representation and the future of pattern recognition, ICPR 2004 Workshop Proceedings, Cambridge, UK, 2004, pp. 43–56.
[30] T. Chin, H. Wang, D. Suter, Robust fitting of multiple structures: The statistical learning approach, in: Proceedings of the International Conference on Computer Vision, 2009, pp. 413–420.
[31] T.-J. Chin, D. Suter, H. Wang, Multi-structure model selection via kernel optimisation, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 3586–3593.
[32] R. Raguram, J.-M. Frahm, Recon: Scale-adaptive robust estimation via residual consensus, in: Proceedings of the International Conference on Computer Vision, 2011, pp. 1299–1306.
[33] A. Delong, O. Veksler, Y. Boykov, Fast fusion moves for multi-model estimation, in: Proceedings of the 12th European Conference on Computer Vision, 2012, pp. 370–384.
[34] T.-T. Pham, T.-J. Chin, J. Yu, D. Suter, The random cluster model for robust geometric fitting, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2012, pp. 710–717.
[35] P. H. S. Torr, An assessment of information criteria for motion model selection, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (1997) 47–53.
[36] Y. Kanazawa, H. Kawakami, Detection of planar regions with uncalibrated stereo using distributions of feature points, in: British Machine Vision Conference, 2004, pp. 247 – 256.
[37] M. Zuliani, C. S. Kenney, 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, 2005.
[38] R. O. Duda, P. E. Hart, Pattern Classification and Scene Analysis, John Wiley and Sons, 1973, pp. 98–105.
[39] R. Subbarao, P. Meer, Nonlinear mean shift for clustering over analytic manifolds, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, York, USA, 2006, pp. 1168–1175.
[40] R. Toldo, A. Fusiello, Automatic estimation of the inlier threshold in robust multiple structures fitting, in: Proceedings of the 15th International Conference on Image Analysis and Processing (ICIAP), Vol. 5716 of Lecture Notes in Computer Science, Springer, Vietri sul Mare, Italy, 2009, pp. 123–131.
[41] D. F. Fouhey, Multi-model estimation in the presence of outliers, Master's thesis, Middlebury College, Vermont, adviser: D. Scharstein (2011).
[42] M. Farenzena, A. Fusiello, R. Gherardi, Efficient Visualization of Architectural Models from a Structure and Motion Pipeline, in: Eurographics 2008 - Short Papers, Eurographics Association, Crete, Greece, 2008, pp. 91–94.
[43] L. Kobbelt, J. Vorsatz, H. Seidel, Multiresolution hierarchies on unstructured triangle meshes, Computational Geometry 14 (1-3) (1999) 5–24.
[44] R. Gherardi, M. Farenzena, A. Fusiello, Improving the efficiency of hierarchical structure-and-motion, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2010, pp. 1594 – 1600.