Towards Unsupervised Reconstruction of Architectural Models

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

Dipartimento di Informatica, Università di Verona
Strada Le Grazie 15 - 37134 Verona
Email: farenzena@sci.univr.it

Abstract

Architectural blueprints offer a concise, clear and high-level description of the structure of a building. On the other hand, state of the art reconstruction pipelines can nowadays produce dense point clouds or high-polygon meshes without any human intervention from a set of digital images or video. We present a fully automated structure and motion framework capable of capturing the expressivity of a high-level description without sacrificing the minute details of an accurate reconstruction. Our resulting architectural models, composed of textured high-level geometric primitives, capture the overall structure of a building and give birth to a more tractable and abstract model of the imaged scene, thereby narrowing the semantic gap in 3D reconstruction. Several examples display our system in action.

1 Introduction

While the current state of the art in urban three-dimensional reconstruction (3D) has focused on the recovery of dense and accurate representations of objects imaged through pictures or video, the sustained interest in accessible architectural modeling software is a strong evidence of an untapped general need for compact, abstract representations of architectural objects.

What separates dense triangulated reconstructions from higher-level renditions of an architectural model is a *semantic gap*, which must be bridged exploiting additional information. First-order knowledge that can be injected includes meshing, surfaces, ordering, occlusion, and parallelism and orthogonality of structures. Increasing levels of abstraction can then be obtained progressing to recognition of scene elements or entire architectural scenes.

In this paper we present a complete 3D reconstruction pipeline capable of producing compact descriptions composed of textured quadric surfaces using as its sole input a sparse collection of digital images.

High-level primitives such as planes and generalized cones are ideal descriptors for architectural buildings and manufactured articles in general. They enables direct abstract reasoning about attributes like parallelism [7], perimeter or planimetry, and the extraction of high level properties (such as symmetry, or function) and unseen geometry.

The process leverages models from unorganized point clouds to editable, CAD-friendly representations that narrow the gap between acquisition and manipulation of architectural models.

Our integrated approach is composed of a front-end section constituted by a Structure and Motion (SaM) pipeline specifically tailored for robustness, that is able to automatically reconstruct 3D points and cameras from uncalibrated views. The resulting unorganized point cloud is subsequently augmented by fitting its elements with geometrical primitives such as planes and cylinders. The back-end is a surface recovery procedure that exploits the segmentation from the previous stage and image contraints to produce a textured triangular mesh. This mesh can be optionally augmented with a relief map that recovers the fine geometry discarted in the previous steps.

The final system brings seamlessly together previous art and novel solutions in an unsupervised framework which needs relatively few assumptions or restrictions.

The rest of the paper is organized as follows. In Sec. 2 we will survey the literature most closely related to our work. Sec. 3 will describe our SaM pipeline, while Sec. 4 outlines the surface recovery stage. Several experimental results validating our approach are reported in Sec. 5. Conclusions are drawn in Sec. 6.

2 Related work

The approaches covered in the literature for solving the problem of architectural/urban reconstruction can be categorized in two main branches: a first one [30, 34, 2, 17] is composed of the *Structure and Motion* (SaM) pipelines that are able to handle the reconstruction process making no assumptions on the imaged scene and without manual intervention.

These methods usually share a common structure and produce as output, along with camera parameters, an arbitrarily dense but ultimately unorganized point cloud which fails to model surfaces ([11] being the notable exception).

The second category comprises the methods specifically tailored for urban environmentsand engineered to be mounted on survey vehicles [24, 4]. These systems usually rely on a host of additional information, such as GPS and inertial sensors, and output dense polygonal maps using stereo triangulation.

Both approaches produce large amounts of data, making it difficult to store, render, analyze or disseminate the results. The most scalable approach was shown in [4], developed for compact visualization on consumer navigation products. Road ground and building façades were forced to lie on textured, mutually-orthogonal, gravity-aligned, geo-located planes.

The recovery of the semantic structure of urban elements, in order to produce simpler and more tractable models, has been tackled by fewer researchers. In this respect, the two most similar articles to the work presented here are [6] and [28]. In [6] is described a system that specializes in creating architectural models from a limited number of images. Initially a coarse set of planes is extracted by grouping point features; the models are subsequently refined by casting the problem in a Bayesian framework where priors for architectural parts such as doors and windows are incorporated or learnt. A similar deterministic approach is developed in [28] where dominant planes are recovered using a orthogonal linear regression scheme: façade features, which are modeled as shaped protrusions or indentations, are then selected from a set of predefined templates. Both methods rely on a large amount of prior knowledge to operate, either implicitly or explicitly, and make strict assumption on the imaged scene.

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

While the problem of fitting quadric primitives has been extensively investigated in literature (see [26] for a survey of the topic) most of published material is designed to be applied to dense point clouds produced by laser scanners or to already triangulated meshes. Common assumptions include uniform sampling and negligible acquisition noise; such methods can't therefore be used for processing 3D clouds produced by Structure and Motion pipelines which don't provide connectivity, are unevenly sampled and corrupted by a comparatively large signal-to-noise ratio.

3 Structure and Motion

Given a collection of uncalibrated images of the same scene, with constant intrinsic parameters, the SaM pipeline outputs camera parameters, pose estimates and a sparse 3D points cloud of the scene. Our SaM pipeline is made up of state-of-the-art algorithms and follows an incremental greedy approach, similar to [30] and [27]. The most efforts have been made in the direction of a robust and automatic approach, avoiding unnecessary parameters tuning and user intervention. A sample output is shown in Fig. 1.

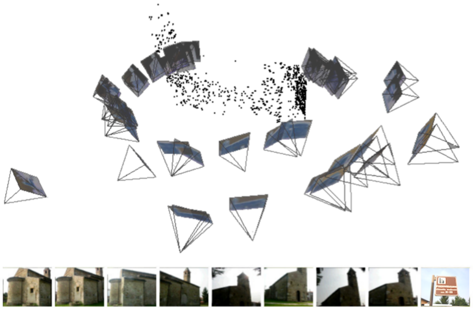


Figure 1: Reconstruction of the ”Pozzoveggiani” dataset.

3.1 Multimatching

Initially, keypoints are extracted and matched over different images. This is accomplished using SIFT [21] for detection and description of local point features. Matching follows a nearest neighbor approach [21], with rejection of those keypoints for which the ratio of the nearest neighbor distance to the second nearest neighbor distance is greater than 2.0.

Homographies and fundamental matrices between pairs of images are then computed using RANSAC [9]. At this point we have a set of matches that are considered inliers for a certain model. However, in order to increase the robustness of the method further, we apply an outlier rejection rule, called X84 [12]. Let e_i be the residuals, a robust noise scale estimator is the Median Absolute Deviation (MAD):

$$\sigma^* = 1.4826 \operatorname{med}_i |e_i - \operatorname{med}_j e_j|. \quad (1)$$

The robustified inliers[35] are those points such that $e_i < 3.5\sigma^*$. The model parameters are eventually re-estimated via least-squares minimization of the (first-order approximation of) geometric error [13].

The best-fit model (homography or fundamental matrix) is selected according to the Geometric Robust Information Criterion (GRIC) [33]:

$$\mathrm{GRIC} = \sum \rho(e_i^2) + nd \log(r) + k \log(rn) \quad (2)$$

$$\rho(e) = \min\left(\frac{e^2}{\sigma^2}, 2(r-d)\right) \quad (3)$$

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

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

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

3.2 Autocalibration

The intrinsic parameters K of the camera are constant but unknown. A globally convergent autocalibration algorithm [10] is employed to recover them automatically from the set of fundamental matrices calculated during the matching phase. In short, the algorithm uses Interval Analysis to minimize the following cost function:

$$\chi(K) = \sum_{i,j} w_{ij} \frac{2\operatorname{tr}(E_{ij}{E_{ij}}^\mathsf{T})^2 - \operatorname{tr}^2(E_{ij}{E_{ij}}^\mathsf{T})}{\operatorname{tr}^2(E_{ij}{E_{ij}}^\mathsf{T})} \quad (4)$$

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

3.3 Initialization

Once the intrinsic parameters are known, the position of each view as well as the 3D location of the tracks is recovered using an incremental approach that entails to start from a seed reconstruction, made up of two calibrated views and the relative 3D points in a Euclidean frame. The extrinsic parameters of two given views is obtained by factorizing the essential matrix, as in [14]. Then 3D points are reconstructed by intersection (via the midpoint algorithm [1]) and pruned using X84 on the reprojection error. Bundle adjustment (BA) [20] is run eventually to improve the reconstruction.

The choice of the two views for initialization turns out to be critical [31]. It should be a compromise between distance of the views and the number of keypoints in common. We require that the matching points must be well spread in the two images, and that the fundamental matrix must explain the data far better than other models (namely, homography), according to the GRIC, as in [27]. This should ensure that the baseline between the two images is large, and that the fundamental matrix correctly captures the structure of the scene, so that triangulation is well-conditioned and the estimation of the starting 3D structure is reliable. The heuristic adopted in practice is then:

$$\mathcal{S}_{i,j} = \frac{CH_i}{A_i} + \frac{CH_j}{A_j} + \frac{\operatorname{gric}(F_{i,j})}{\operatorname{gric}(H_{i,j})}, \quad (5)$$

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

Structure and motion pipeline

1. Multimatching:
   (a) Extract keypoints in each image;
   (b) Match keypoints between each pair of images;
   (c) Find the best-fit model using RANSAC and GRIC;
   (d) Reject outliers using X84 rule on distance to the best-fit model;
   (e) Link keypoints into tracks.
2. Autocalibration, using the fundamental matrices;
3. Initialization:
   (a) Select two views according to (5);
   (b) Compute their extrinsic parameters via factorization of essential matrix.
4. Incremental Step Loop:
   (a) Compute 3D points with intersection and run X84 on the reprojection error;
   (b) Add new 3D points to the reconstruction;
   (c) Run BA on the current reconstruction;
   (d) Select the next view;
   (e) Initialise camera pose with RANSAC and linear exterior orientation;
   (f) Add the camera to the reconstruction;
   (g) Run BA on the current reconstruction;
   (h) Select new tracks;

3.4 Incremental Step Loop

After initialization, a new view at a time is added until there are no remaining views. The next view to be considered is the one that contains the largest number of tracks whose 3D position has already been estimated. This gives the maximum number of 3D-2D correspondences, that are exploited to solve an exterior orientation problem via a linear algorithm [8]. The algorithm is used inside a RANSAC iteration, in order to cope with outliers. The extrinsic parameters are then refined with BA.

Afterwards, the 3D structure is updated by adding new tracks, if possible. Candidates are those tracks that have been seen in at least one of the cameras in the current reconstruction. 3D points are reconstructed by intersection (midpoint algorithm), and successively pruned using X84 on the reprojection error. As a further caution, 3D points for which the intersection is ill-conditioned are discarded, using a threshold on the condition number of the linear system.

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

4 Surface reconstruction

The recovery of the geometric structure in the form a 3D-point cloud does not produce a model of the object's surface. However, even describing the surface of the imaged object as a triangulated mesh is not enough: to convey meaning, structure must be *constructively specified*, for example linking together parts from a architectural structure database or using constructive solid geometry. We chose to build our models out of geometric primitives representing their surfaces.

4.1 High-level primitive fitting

The first stage is fitting simple geometric primitives such as planes, cylinder or spheres to the data. We developed a specific approach that enables data self-organization and copes naturally with multiple structures. Given a distribution of points corrupted by outliers, the algorithm generates a set of model hypotheses by repeatedly drawing at random the minimal required number of samples for each desired structure. Then data is transformed into its *conceptual representation*: each data point is represented with the characteristic function of the set of models preferred by that point. Multiple models are revealed as clusters in the conceptual space.

A specific agglomerative clustering procedure, called J-linkage, for it is based on the Jaccard distance, has been developed [32]. The Jaccard distance measures the degree of overlap of the two sets and ranges from 0 (identical sets) to 1 (disjoint sets). In formulae, given two sets A and B, the Jaccard distance is defined as:

$$d_{\mathrm{J}}(A, B) = \frac{|A \cup B| - |A \cap B|}{|A \cup B|} \tag{6}$$

The J-Linkage procedure is summarized below.

J-linkage

1. Put each point in its own cluster.
2. Define the cluster's PS 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.

Each cluster of points defines (at least) one model. The final model for each cluster of points is estimated by least squares fitting.

Model selection is subsequently used first to merge different model instances of the same type (*intra-model selection*) and then to determine the best-fit model among different ones (*inter-model selection*), namely planes, spheres or cylinders. The best-fit model is determined using the Bayesian Information Criterion (BIC), which proved to produce better results in our experiments. Eventually, each inlier data point belongs to one (and only one) model. An example is shown in Fig. 2.

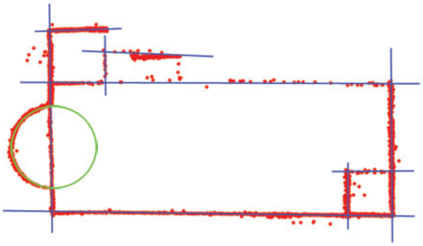


Figure 2: Automatically recovered planes and cylinders from the 3D point cloud (top view).

4.2 Image-consistent triangulation

Estimating a sound triangulation on the output of a structure and motion pipeline is inherently difficult because the recovered 3D information suffer from uneven sampling and reconstruction errors. This inhibits the use of a large part of algorithms for recovering meshes from unorganized point clouds like for example [16]. Therefore, we turn our attention to *image-consistent* triangulation algorithms, i.e., algorithms that uses information from the images to guide the triangulation of 3D points.

Following [3] we first augment our point cloud by adding points along the intersections between the recovered primitives, provided that these points projects onto actual image edges. As a result the model's boundaries are better preserved, as seen in Fig. 3.

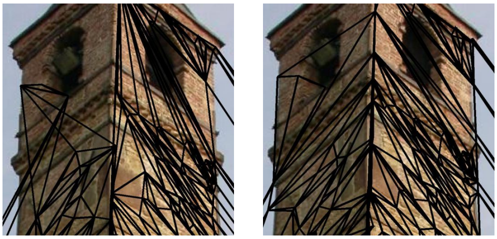


Figure 3: Detail of the triangulation before (left) and after (right) augmentation with boundary points.

The initial triangulation is calculated by projecting the recovered 3D points to their belonging surface and applying the 2D Delaunay triangulation algorithm. This approximation contains *spurious triangles* that does not correspond to a planar patch in 3D. They may arise because a single surface has been fitted to data consisting of actually two distinct surfaces, separated by a gap. Delaunay triangulation subsequently links all the points of the unique surface, resulting in triangles that spans over the gap. Moreover, Delaunay triangulation is convex by construction, therefore it adds spurious triangles along the concavities of the boundary of the object (see Fig. 4).

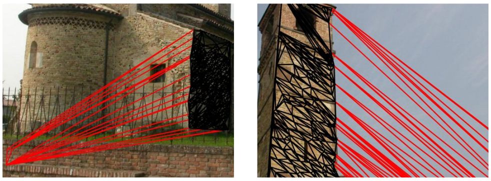


Figure 4: Examples of spurious triangles covering textured (left) and uniform (right) areas.

Firstly, a test is performed taking visibility into account: triangles are projected onto the original views, and those covering visible points are removed, as in [15].

If the spurious triangle covers a textured area in the image it can be detected by applying a check based on the appearance of the underlying surface, like suggested in [22, 25]. The rationale behind those methods is that under the assumptions that surfaces are planar and Lambertian, all the different views of the same triangle are similar or *photo-consistent*. As customary, we used the Sum of Squared Differences (SSD) of the pixels intensity to measure photo-(in)consistency and X84 to automatically reject inconsistent triangles, assuming that the majority of them are indeed consistent (see Fig. 5).

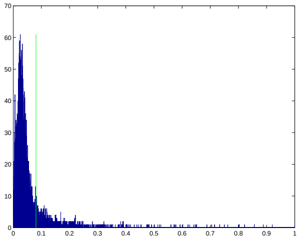


Figure 5: Histogram of SSD values over corresponding triangles in all views. The vertical line is the X84 threshold.

This procedure works well in textured areas but fails to detect spurious triangles that cover a uniform area like sky or grass patches. Therefore, along the same line as [23], a different strategy is adopted to cope with these triangles.

The idea is to focus on edges on boundary the triangulation: such edges must coincide with intensity edges, if the triangle is correct. Therefore, spurious triangles are detected and removed by checking image gradient along each boundary edge, starting from the outer triangles and proceeding inward until no triangle needs to be removed.

Finally, the inner triangles are substituted with fewer and less skinny ones (see Fig. 6). This is achieved thanks to a constrained conforming Delaunay triangulation [29] with minimum angle set to 20 deg.

The final triangular mesh is shown in Fig. 7.

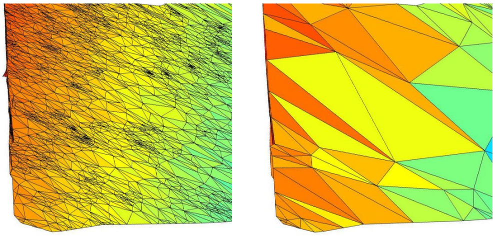


Figure 6: Detail of the triangulation before (left) and after (right) simplification.

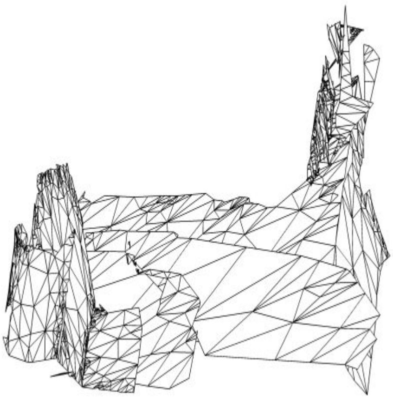


Figure 7: The triangulated model for the ”Pozzoveggiani” example.

4.3 Relief map extraction

Having obtained a compact model from the original pictures we can optionally augment it with relief textures, thus recording also the fine geometry lost during the primitives extraction, as in [5]. We obtained the preliminary results shown in Fig. 8, developing a simplified version of a recent stereo algorithm based on gestalt principles [19]. While based on local methods, it can achieve good performance by employing large disparity neighbourhoods. The problem usually associated with large correlation windows are minimized by weighting the stereo cost function with a measure of similarity and proximity between candidate matches, thus mimicking the behaviour of stereo algorithms based on explicit segmentation.

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

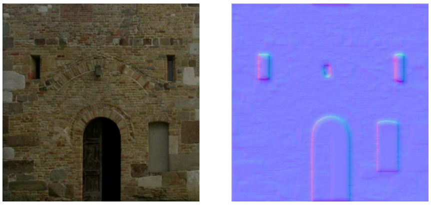


Figure 8: Color and normal textures automatically generated for the front of the church.

process the pairs with excessive distortions. Conflicts in depth arising from different couples are resolved taking the median of the estimates. Once disparity has been obtained recovering bump, normal and displacement maps is straightforward; these data enables the simulation of fine geometry and the use of modern rendering algorithm such as [18] and its more recent derivations.

5 Experiments

Our technique was tested on several, large architectural models. We will show three of them portraying a small medieval church (“Pozzoveggiani”), the well-known Valbonne dataset and a section of a stronghold (“Controporta”). All pictures were taken in uncontrolled settings and environment, and no additional information was used.

Pozzoveggiani. The dataset is composed of 54 images acquired from the ground plane with a consumer camera at a resolution of 1024x768 pixels, at different times and with automatic exposure (Fig. 9(a)). Photos contain occlusions, scale changes, uneven brightness, sun flares and an outlier that we inserted purposely to verify its rejection.

The church itself has a fairly simple planimetry: the perimeter is composed of straight walls, with a bell tower and a slanted roof covered with bent tiles. A cylindrical apse protrudes from the back; several arches and slit windows open into the well-textured brick walls.

In Fig. 9(b) the complete point cloud generated from the SaM section of the pipeline (described in Sec. 3) is shown. It displays good continuity properties and a remarkable accuracy in modeling the perimetric walls but also a uneven density (see Fig. 1) caused by the different number of pictures imaging each side: for this reason the roof, hardly visible from the ground plane, remains unmodeled.

The model extraction correctly recovers all perimetric planes (the average angle between orthogonal planes is $90.44\,\deg$) and fits a cylinder to the apse as expected (Fig. 9(c)). The subsequent triangulation is shown in Fig. 9(d) and 9(e). Most of the surface is correctly reconstructed, with only few missing triangles in correspondence with loosely sampled locations and strong border interference.

Controporta. Our second example is composed of twelve pictures of a massive ruined fortification on a grassy plane. Imaged at a resolution of 1280x960 pixels it is composed of straight sections of stone wall.

Most detected tracks lie uniformly on the flat surfaces (Fig. 9(g)), which are all correctly identified in the primitive extraction step (Fig. 9(h)). A small cluster of 3D points, localized on the grass plot next to the camera, enables the recovery of the ground plane. The gap that can be seen in Figures 9(i) and 9(j) between the back and foreground geometry is structural, because of the occluding grass patch.

Valbonne. The last experiment uses 15 photos of the Valbonne church, extensively used in calibration literature. The dataset is recorded at a resolution of 768x512 pixels, in varying condition of illumination and occlusion.

Recovered tracks (Fig. 9(l)) cover the front and side faces of the building. Three main dominant planes are recovered, as seen in Fig. 9(m), with the front face assimilating the contributes of the two protrusions at its sides.

6 Conclusions

We presented a complete reconstruction pipeline for large architectural scenes capable of automatically recovering from a set of sparse pictures a compact and meaningful representation composed of textured high-level geometric primitives. This format, which conveys the semantic structure of the imaged environment, has obvious advantages when compared with unorganized point clouds or overly dense meshes produced by competing approaches. As such, it has the potential to narrow the current gap existing between acquisition, editing and visualization of urban scenes.

Acknowledgments. The use of the "Triangle" software by J. R. Shewchuk is here acknowledged.This project was supported by PRIN 2006 project 3-SHIRT. At the time of writing, M. Farenzena is affiliated with ASMEA, UMR6602 CNRS, Clermont (F).

References

[1] P. Beardsley, A. Zisserman, and D. Murray. Sequential update of projective and affine structure from motion. *Int. Journal of Computer Vision*, 23(3):235–259, 1997.

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

[3] O. Cooper, N. Campbell, and D. Gibson. Automatic augmentation and meshing of sparse 3D scene structure. In *Proceedings of the Seventh IEEE Workshops on Application of Computer Vision (WACV/MOTION)*, pages 287–293, Washington, DC, USA, 2005.

[4] N. Cornelis, K. Cornelis, and L. Van Gool. Fast compact city modeling for navigation pre-visualization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 1339–1344, 2006.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In Holly Rushmeier, editor, *SIGGRAPH Conference Proceedings*, pages 11–20, New Orleans, Louisiana, August 1996.

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

[7] M. Farenzena and A. Fusiello. 3D surface models by geometric constraints propagation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Anchorage (Alaska), June 24-26 2008.

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

[9] M. A. Fischler and R. C. Bolles. Random Sample Consensus: a paradigm model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, June 1981.

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

[11] Michael Goesele, Noah Snavely, Brian Curless, Hugues Hoppe, and Steven M. Seitz. Multi-view stereo for community photo collections. In *Proceedings of the International Conference on Computer Vision*, Rio de Janeiro, Brazil, October 14-20 2007.

[12] 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.

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

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

[15] A. Hilton. Scene modelling from sparse 3d data. *Image Vision Computing*, 23(10):900–920, 2005.

[16] H. Hoppe, T. DeRose, T. Duchamp, M. Halstead, H. Jin, J. McDonald, J. Schweitzer, and W. Stuetzle. Piecewise smooth surface reconstruction. In *SIGGRAPH Conference Proceedings*, pages 295–302, New York, USA, 1994.

[17] G. Kamberov, G. Kamberova, O. Chum, S. Obdrzalek, D. Martinec, J. Kostkova, T. Pajdla, J. Matas, and R. Sara. 3D geometry from uncalibrated images. In *Proceedings of the 2nd International Symposium on Visual Computing*, Springer Lecture Notes in Computer Science, 2006.

[18] T. Kaneko, T. Takahei, M. Inami, N. Kawakami, Y. Yanagida, T. Maeda, and S. Tachi. Detailed shape representation with parallax mapping. In *Proceedings of the ICAT 2001*, pages 205–208, 2001.

[19] Kuk-Jin Yoon; In-So Kweon. Locally adaptive support-weight approach for visual correspondence search. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, pages 924–931, 2005.

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

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

[22] D. D. Morris and T. Kanade. Image-consistent surface triangulation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 1, pages 332–338, June 2000.

[23] Atsutada Nakatsuji, Yasuyuki Sugaya, and Kenichi Kanatani. Optimizing a triangular mesh for shape reconstruction from images. *IEICE - Transactions on Information and Systems*, E88-D(10):2269–2276, 2005.

[24] P. Mordohai et al. Real-time video-based reconstruction of urban environments. In *3D-ARCH 2007: 3D Virtual Reconstruction and Visualization of Complex Architectures*, July 12-13 2007.

[25] J.-S. Perrier, G. Agam, and P. Cohen. Image-based view synthesis for enhanced perception in teleoperation. In J. G. Verly, editor, *Enhanced and Synthetic Vision (Proceedings SPIE)*, volume 4023, pages 213–224, June 2000.

[26] S. Petitjean. A survey of methods for recovering quadrics in triangle meshes. *ACM Computing Surveys*, 2:1–61, 2002.

[27] M. Pollefeys, L.V. Gool, M. Vergauwen, K. Cornelis, F. Verbiest, and J. Tops. Video-to-3d. In *Proceedings of Photogrammetric Computer Vision 2002*, number 34 in International Archive of Photogrammetry and Remote Sensing., page 252 258, 2002.

[28] Konrad Schindler and Joachim Bauer. A model-based method for building reconstruction. In *Proceedings of the First IEEE International Workshop on Higher-Level Knowledge in 3D Modeling And Motion Analysis*, page 74, Washington, DC, USA, 2003.

[29] J. Shewchuk. Delaunay refinement algorithms for triangular mesh generation. *Computational Geometry: Theory and Applications*, 22(1–3):86–95, 2002.

[30] Noah Snavely, Steven M. Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3D. In *SIGGRAPH Conference Proceedings*, pages 835–846, NY, USA, 2006.

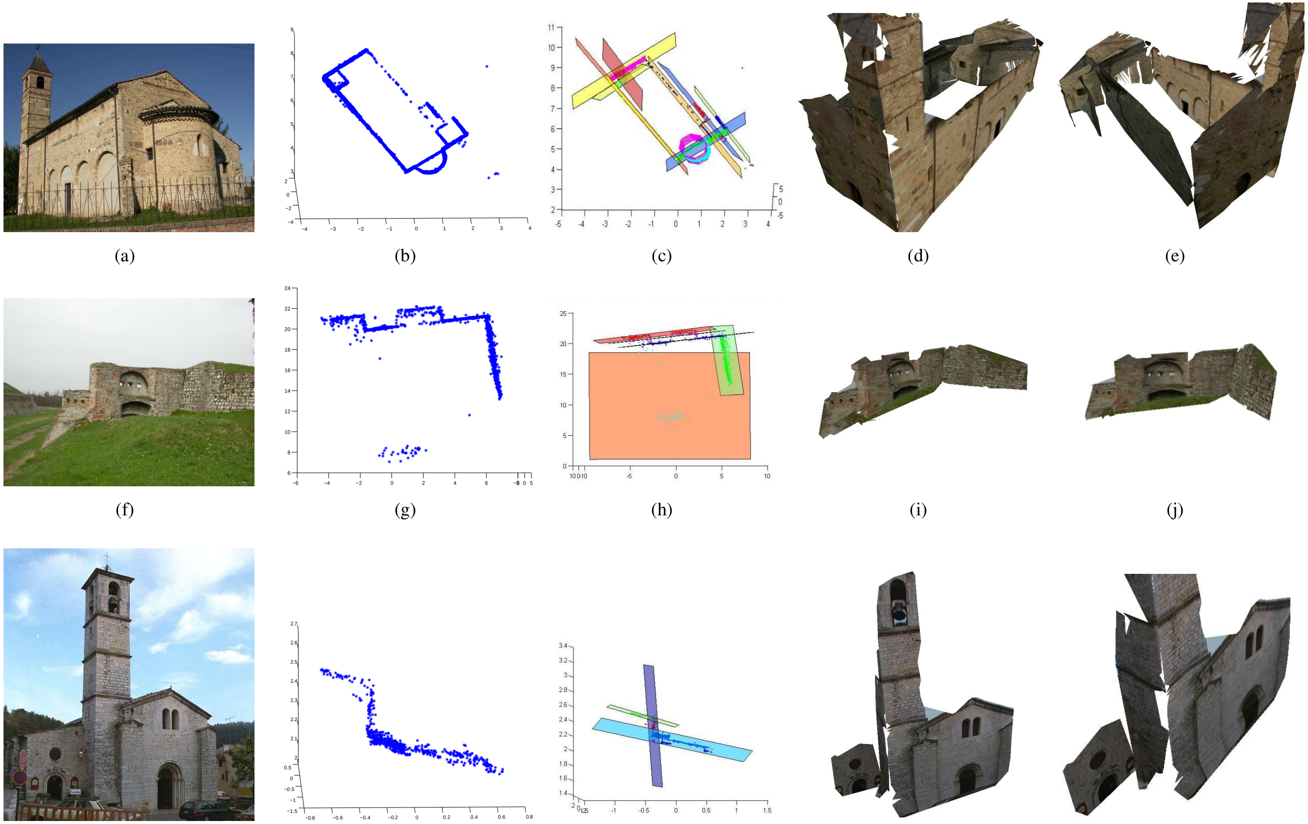
[31] T. Thormählen, H. Broszio, and A. Weissenfeld. Keyframe selection for camera motion and structure estimation from multiple views. In Tomás Pajdla and Jiri Matas, editors, *Proceedings of the European Conference on Computer Vision*, volume 3021 of *Lecture Notes in Computer Science*, pages 523–535, 2004.

[32] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-linkage. In *Proceedings of the European Conference on Computer Vision*, Nice, FR, October 2008.

[33] 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.

[34] M. Vergauwen and L. Van Gool. Web-based 3d reconstruction service. *Machione Vision and Applications*, 17(6):411–426, 2006.

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



(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) Original image (l) Recovered 3D points (m) Recovered structures (n) Triangulated model (o) Another view

Figure 9: The “Pozzoveggiani”, “Controporta” and “Valbonne” experiments.