3D Geometry from Uncalibrated Images

George Kamberov[1], Gerda Kamberova[2], O. Chum[3], Š. Obdržálek[3], D. Martinec[3], J. Kostková[3], T. Pajdla[3], J. Matas[3], and R. Šára[3,⋆]

[1] Stevens Institute of Technology
Hoboken, NJ 07030, USA
gkambero@stevens.edu
[2] Hofstra University,
Hempstead, NY 11549, USA
cscglk@hofstra.edu
[3] Center for Machine Perception
Department of Cybernetics
Faculty of Electrical Engineering
Czech Technical University
Prague 6, Czech Republic
{sara, martid1, pajdla, chum, matas}@cmp.felk.cvut.cz

Abstract. We present an automatic pipeline for recovering the geometry of a 3D scene from a set of unordered, uncalibrated images. The contributions in the paper are the presentation of the system as a whole, from images to geometry, the estimation of the local scale for various scene components in the orientation-topology module, the procedure for orienting the cloud components, and the method for dealing with points of contact. The methods are aimed to process complex scenes and non-uniformly sampled, noisy data sets.

1 Introduction

In this paper we present a collection of computational methods and a pipeline system which extracts the geometric structure of a scene and makes quantitative measurements of the geometric properties of the objects in the scene from a set of images taken in real-life conditions at different unknown discrete instances over some period of time, without the luxury of performing calibration and parameter estimation during the acquisition. The pipeline does not require a human in the loop. This work was motivated by many applications where, due to communications, synchronization, equipment failures, and environmental factors, an intelligent system is forced to make decisions from an unorganized set of images offering partial views of the scene. The presented pipeline has three principal stages: (i) an image-to-3D-point-and-tangent-plane pipeline; (ii) an orientation-topology module which orients the cloud, assigns topology, and partitions it into

⋆ This research was supported by The Czech Academy of Sciences under project 1ET101210406 and by the EU projects eTRIMS FP6-IST-027113 and DIRAC FP6-IST-027787.

G. Bebis et al. (Eds.): ISVC 2006, LNCS 4292, pp. 802–813, 2006.

connected manifold components; and (iii) a geometry pipeline which recovers the local surface geometry descriptors at each surface point.

The contributions in the paper are the presentation of the system as a whole, from images to geometry, the estimation of the local scale for various scene components in the orientation-topology module, the procedure for orienting the cloud components, and a method of dealing with points of contact.

The results on two different sets of unorganized images are illustrated through out the paper.

Previous Research. The data acquisition method implied by our scenario could lead to reconstructions of exceptional geometric complexity. The point clouds could be sparse and nonuniform; both positions and tangent plane estimates are expected to be noisy. Because of noise, but also because of the freedom to collect and combine different views, we might have objects that touch each other – such touching objects should be separated. Furthermore, many of the reconstructed surfaces will not be closed; they will have boundary points. There are numerous surface reconstruction methods [1,2,3,10,11,12] which work well for closed surfaces and clouds that satisfy some reasonable sampling conditions (for example, uniformly sampled clouds, evenly sampled clouds, or densely sampled clouds). Such methods require a human in the loop to deal with boundary points, non uniformly sampled clouds, and touching surfaces. Furthermore, recent studies indicate that the polygonal surfaces and other surface type reconstructions do not lead to gained precision in computing geometric properties [35]. What is needed is a method to find neighborhoods tight in terms of surface distance and a robust method for computing the differential properties at a point. The framework proposed in [18] shows how to do this at the 3D point level without ever spending time on reconstruction of polygonal, parametric, or implicit representations. Here we present novel methods for adaptive scale selection and for rapid orientation of a cloud endowed with tangent planes only (not oriented normals), see the relevant subsection in Section 2.2. The main difference with earlier approaches [22,28] dealing with scale selection and orientation of point clouds is that we do not impose limitations on the sampling density, do not assume known noise statistics, and do not use a set of implicit tuning parameters; compared to the orientation propagation in [14,12] which rely on minimum spanning tree, our approach is more efficient since it does global flips on sets of normals.

Our system differs from the system for 3D reconstruction from video proposed in [30], in two aspects: (i) the 3D reconstruction can be obtained from wide baseline images only [24] and (ii) the scene is automatically segmented into objects.

2 System Overview: From Images to 3D Geometry

The system is organized as a pipeline consisting of three modules: the image-data pipeline, the orientation-topology module, and the geometry pipeline. The input data are photographs of the scene to be reconstructed, taken with a hand-held compact digital camera which is free to move in the scene and to zoom in and out. The images are input into the data pipeline which outputs an unorganized

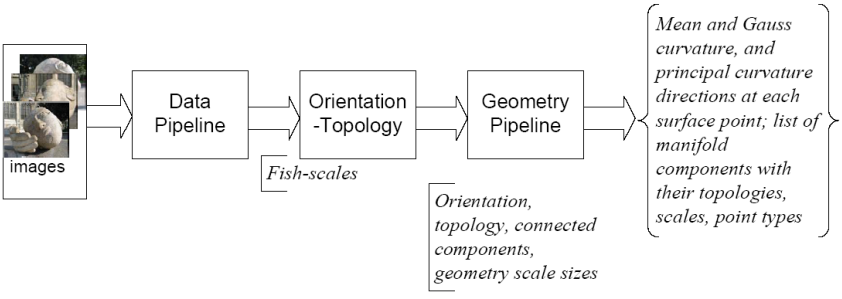

Fig. 1. The system: from images to cloud with 3D geometry descriptors

3D point cloud with fish-scales attached to the points.Each fish-scale encodes a 3D point and an estimate of the plane tangent to the cloud at this point. The fish-scales (3D points with tangent planes) are input into the orientation-topology module which recovers local scale, orientation and topology for the cloud and separates it into connected manifold components. The latter are processed by the geometry pipeline which recovers the geometric descriptors at the surface points.

2.1 The Data Pipeline

The data processing pipeline consists of several steps: (1) wide-baseline matching, (2) structure from motion, (3) dense matching, (4) 3D model reconstruction. The 3D model produced by the data pipeline consist of a collection of fish-scales.

We used the method in [24]. A brief sketch follows. First, sparse correspondences are found across all image pairs. Pairwise image matching is done with Local Affine Frames [26] constructed on intensity and saturation MSER regions, LaplaceAffine and HessianAffine [27] interest points. An epipolar geometry unaffected by a dominant plane is found using [7]. The inliers are used as the pool for drawing samples in calibrated RANSACs. This scheme is applied to the 6-point algorithm [34] as well as to the 5-point algorithm [29]. First the 6-point algorithm is run on all pairs (with some minimal support) and the focal length is estimated as the mean of the estimates from individual pairs. Then, the 5-point algorithm [29] is run using fully known camera internals.

A multi-view reconstruction is estimated given pair-wise Euclidean reconstructions by [29] up to rotations, translations and scales. The partial reconstructions are glued by the following three step procedure: (i) camera rotations consistent with all reconstructions are estimated linearly; (ii) all the pair-wise reconstructions are modified according to the new rotations and refined by bundle adjustment while keeping the corresponding rotations same; (iii) the refined rotations are used to estimate camera translations and 3D points using Second Order Cone Programming by minimizing the L_∞-norm [15].

The method from [24] can be used in extreme cases of missing data, i.e., when each point is visible in two images only in a (sub)set of images. It is capable of

dealing with degenerate situations like dominant planes, pure camera rotation (panoramas) and zooming. The Head2 scene used in this paper is an example in which all these inconveniences appear. No projective-to-metric upgrade is needed as in [23]. Compared to incremental structure-from-motion methods, gluing all pair-wise geometries at the same time has the advantage that the global minimum of an approximation to the reprojection error is achieved [23]. As a consequence, no drift removal [9] is needed. The current limitation of [24] is that the translation estimation using [15] can be harmed by mismatches. This limitation can be removed by using the method in [33] instead. See [25] for a demo with 3D vrml models of difficult data sets.

The next step is pairwise image rectification, which improves matching efficiency. We use Hartley's method [4]. Pairs for dense matching are selected based on the mutual location of the cameras, as described in [8]. After that, dense matching is performed as disparity search along epipolar lines using Stratified Dense Matching [20]. The algorithm has a very low mismatch rate [21], it is fast, robust, and accurate, and does not need any difficult-to-learn parameters. The output from the matching algorithm is one disparity map per image pair admitted for dense matching. By least squares estimation using an affine distortion model the disparity maps are upgraded to sub-pixel resolution [31].

Fig. 2. The Data Pipeline: From an unorganized set of images to fish-scales. A fish-scale is a 3D point with a local covariance ellipsoid centered at it. It encodes the position, a measure of the spatial density and the tangent plane to the cloud at the 3D point.

The disparity maps are used to reconstruct the corresponding 3D points. The union of the points from all disparity maps forms a dense point cloud. An efficient way of representing distributions of points is to use fish-scales [32]. Fish-scales are local covariance ellipsoids that are fit to the points by the K-means algorithm. They can be visualized as small round discs. A collection of fish-scales approximate the spatial density function of the measurement in 3D space.

2.2 Orientation-Topology Module

To deal with the complexity of the data and noise (intermediate point sets recovered in the data pipeline use millions, redundant, noisy points) the points in the final output cloud of the data pipeline are the centers of the fish-scales. This cloud, unorganized set of fish-scale centers, together with the corresponding set of the covariance ellipsoids is the input to the orientation-topology module.

Fig. 3. The Data Pipeline: A series of images and the 3D cloud reconstructed from them. The area around the door is reconstructed at much higher density than the rest of the cloud.

Each covariance ellipsoid defines a tangent plane at the corresponding cloud point. The output of the orientation-topology module is a consistently oriented, topologized point cloud, each point of which is equipped with a normal and a neighborhood of points closest to it in terms of surface distance; the points are classified as isolated, curve, and surface points, and the later points are divided into interior and boundary points; the connected components of the topologized cloud are identified, thus the scene is segmented into objects.

A key parameter in the orientation-topology module is the determination of geometrically meaningful local scales. In our scenario different parts of the scene may be reconstructed at different scales since the unorganized set of input images could consist of images collected at different zoom levels and possibly obtained with different cameras. The orientation-topology module is based on the ideas in [18] combined with novel adaptive scale selection and a method for rapid orientation of the cloud endowed with tangent planes only (not oriented normals), see subsection "Scale, orientation and topology of 3D fish-scales cloud" in Section 2.2. A key ingredient is the method for topologizing oriented point clouds introduced in [17] and outlined below.

Topology of an Oriented Cloud (After [17,18]). An oriented point cloud M is a set of 3D points p with normals $\mathbf{N}$, $M = \left\{(p, \mathbf{N}) \in \boldsymbol{R}^3 \times S^2\right\}$. The neighbors of an oriented point P are chosen based on a proximity measure $L_P : M_\rho(P) \to [0, 1]$, where $M_\rho(P)$ is a ball/voxel centered at p of radius ρ. For every candidate neighbor $Q = (q, \mathbf{N}_q) \in M_\rho(P)$, $L_P(Q)$ expresses the likelihood that Q is nearby P on the sampled surface. The likelihood incorporates three estimators of surface distance: a linear estimator based on Euclidean distance, a quadratic estimator based on the cosine between normals, and a new third order estimator δ_p defined below (the last estimator is crucial in distinguishing points which are far on the surface but close in Euclidean distance),

$$L_P(Q) = \left(1 - \frac{|p-q|}{D_{\text{Eucl}}(P)}\right) t(P,Q), \text{ where } D_{\text{Eucl}}(P) = \max_{Q' \in M_\rho(P)} (|p - q'|),$$

$$t(P,Q) = \frac{1}{2}\left(1 + \langle \mathbf{N}_p \,|\mathbf{N}_q \rangle\right)\left(1 - \frac{\delta_p(Q)}{D_{\text{Surf}}(P)}\right)$$

$$\delta_p(Q) \;=\; |\,\langle \mathbf{N}_q - \mathbf{N}_p \,|\overrightarrow{pq}\rangle + 2\,\langle \mathbf{N}_p \,|\overrightarrow{pq}\rangle\,|, \text{ and } D_{\text{Surf}}(P) = \max_{Q' \in M_\rho(P)} (\delta_p(Q')).$$

Note that $< \cdot|\cdot >$ in the equations above denotes the the Euclidean dot product in $\boldsymbol{R}^3$. Once a scale $\rho > 0$ is chosen, then one can construct a neighborhood $U(P)$ of an oriented point $P = (p, \mathbf{N})$, $U(P) = \{Q_i = (q_i, \mathbf{N}_i) : |p - q_i| < \rho\}$, such that the projection of q_i in the plane through p perpendicular to N is not inside a triangle fan centered at p and whose remaining vertices are projections of the remaining base points in the neighborhood. The last condition implies that each neighbor Q_i carries unique information about the distribution of the surface normals around P. Such a neighborhood is called Δ neighborhood of P and the minimum of $L_P(Q_i)$ is called the likelihood of the neighborhood.

The Δ neighborhoods provide a tool to classify the points in an oriented cloud. The *isolated points* do not have Δ neighborhoods with positive likelihood; the rest of the points are either *curve points*, *boundary surface* points, or *interior surface* points. The point P is a surface point if one can use a Δ neighborhood to estimate the orientation along at least two orthogonal directions emanating from the base point. The point P is an interior surface point if one can use a positive likelihood Δ neighborhood to compute the orientation in a full circle of directions centered at p and perpendicular to $\mathbf{N}$. A point is a curve point if it is not a surface or an isolated point. The surface points are samples of 2D manifolds and we can compute differential properties at each such point.

To build Δ neighborhoods we pre-compute and store in a look-up table the proximity likelihoods for points within an Euclidean distance ρ. After that the construction of the neighborhoods for different points can be done in parallel. There is no additional burden involved in dealing with boundary points – in contrast with the extra steps needed if one uses a Dealunay-Voronoi based approach.

Scale, Orientation and Topology for 3D Fish-Scales Cloud. We use an iterative procedure, `OrientationAndTopology` which simultaneously recovers the *voxel scales* appropriate for the different parts of the cloud, the *orientation and topology*, and partitions the cloud into *connected components*. See Figures 7 and 8.

The iterations correspond to the terms in an increasing sequence of voxel sizes. Each re-entry in the loop processes a current fish-scale cloud equipped with an *entry orientation*.

At the initial entry in the loop: the voxel size is set to equal the largest fish-scale diameter in the fish-scale cloud; and the cloud is assigned an initial orientation by choosing an orientation for each fish-scale plane so that all normals point in the same fixed closed half-space (for example with respect to the $x - y$ plane.

During each iteration a procedure, `FixedScaleOrientationAndTopology` is used to compute orientation and topology adapted to this voxel scale, and the connected manifold components. All sufficiently large 2D manifold components are identified. As a heuristic at the moment we consider components to be large if they contain at least 10% of the fish-scales in the current cloud. (Presently we are developing a density based approach to determining the scale.) The current voxel size is assigned as the corresponding scale for these components, then they are removed from the fish-scale cloud, and are handed to the Geometry pipeline. The entry orientation is updated to equal the normals assigned by `FixedScaleOrientatinAndTopology` and the voxel size is increased linearly.

The loop is reentered until the fish-scales are exhausted.

`FixedScaleOrientationAndTopology` *Procedure:* The input is a voxel scale and a cloud of 3D points with normals. It produces a topology (an adjacency graph) by defining a Δ neighborhood for each point in the cloud, the connected components of the cloud, and a new orientation of the cloud adapted to the Δ neighborhoods. The whole process is organized as an iterative improvement (enlarging connected manifold components from largest to smallest) of some initial orientation and topology for fixed scale. The iteration stops when the components stabilize.

During each iteration we use the current orientation and voxel scale to find a topology by computing Δ neighborhoods. Then the cloud is segmented into connected components by finding the strongly connected components of the adjacency graph defined by the topology. Because the initial normals were chosen up to sign, typically the same geometric object will be split into multiple components (adjacent components with discontinuity in the Gauss map along adjacent component boundaries). Thus we have to synchronize the normals of such components.

The procedure explores the adjacency graph of the connected components (not the whole cloud), always starting at the current largest component that has not been touched previously. Two components are adjacent, if they have a pair of boundary points within a scale unit from each other. The normals are averaged over each of the two boundary points' neighborhoods. If the angle between the average normals is close to π, a decision whether to synchronize the normals of the smaller component with the larger one is made based on the proximity likelihoods of the Δ neighborhoods of the interior points which are neighbors of the boundary points in question. For interior surface points along boundary regions with "opposing" normals, but otherwise similar geometry, the likelihoods will be highly correlated and similar. We sample the proximity likelihoods of points in the Δ neighborhoods of the interior points adjacent to the boundary pair, thus we construct two sets of samples, one sample set for each component. We do a statistical test for the equality of the population means of the two samples, and if the population means are equal, we reverse all normals of the smaller component and mark the smaller component as "flipped". After all connected

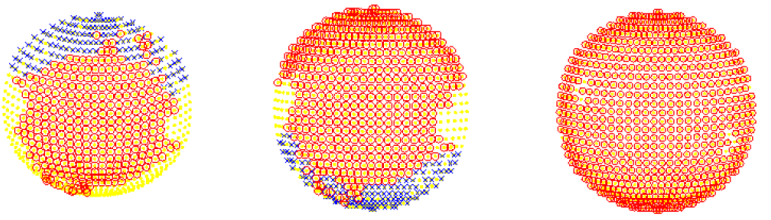

Fig. 4. Growing the topology and orientation. The final stage has a single 2D component which captures 98% of the cloud points. This is a synthetic example using randomly sampled points on a sphere.

Fig. 5. Two tangent surfaces: (Left) The original point cloud. (Right) The two largest connected components: part of a sphere and part of a plane. The two components contain roughly 94% of the cloud.

components with boundaries adjacent to the largest component are examined (and possibly their normals reversed), the topology procedure is run again using the updated normals. The reversal of the normals of a connected component can be implemented very efficiently in almost any programming environment. This orientation propagation amounts to a traversal of a tree with depth not bigger than the number of connected components. This approach allows us to resolve tangency cases as in Figure 5. Then the main loop of the procedure is re-entered until the size of the largest connected component stops to increase. See Figure 4.

2.3 The Geometry Pipeline

The geometry pipeline takes 3D oriented, topologized components and computes the geometric descriptors at surface points (mean curvature, Gauss curvature, and principal curvature directions) following the methods in [14, 13]. These methods use discrete versions of (1)-(3): (1) is a basic identity, [19], used for computing the mean curvature H, $\mathbf{f}$ is a surface parametrization with Gauss map $\mathbf{N}$, and for every tangent vector $\mathbf{v}$, $J(\mathbf{v})$ is the unique tangent vector satisfying $d\mathbf{f}\,(J(\mathbf{v})) = \mathbf{N} \times d\mathbf{f}\,(\mathbf{v})$; the Hopf form, $\omega\,(\mathbf{v})$, defined by (2), is used in computing the symmetric, trace-free coefficient matrix $A = \begin{pmatrix} a & b \\ b & -a \end{pmatrix}$ of the form $< \omega(\cdot)|d\mathbf{f}(\cdot) >$ with respect to an arbitrary positively oriented local basis $(\mathbf{u}, \mathbf{v})$ (note that $< \cdot|\cdot >$ is the Euclidean dot product in $\boldsymbol{R}^3$); and in (3), K is the Gauss curvature, and a and b are the entries of A as defined earlier.

Fig. 6. Geometry Pipeline Output: The two curvature lines foliations extracted on a 2D component in the cloud from Figure 7

$$-Hd\mathbf{f}(\mathbf{v}) = \frac{1}{2}\left(d\mathbf{N}(\mathbf{v}) - \mathbf{N} \times d\mathbf{N}(J(\mathbf{v}))\right) \tag{1}$$

$$\omega(\mathbf{v}) = \frac{1}{2}\left(d\mathbf{N}(\mathbf{v}) + \mathbf{N} \times d\mathbf{N}(J(\mathbf{v}))\right) \tag{2}$$

$$K = H^2 - (a^2 + b^2) \tag{3}$$

The principal curvature vectors of are expressed explicitly in closed form involving the entries of the matrix A. Given an oriented point P and its neighborhood $\mathcal{U}(P) = \{P, P_1, \ldots, P_{k_P}\}$, a directional derivative can be computed along each edge $p_i - p$. Thus, there is a 1-1 correspondence between the neighbors of P and a set of *adapted frames* ϕ_i, defined as $\phi_i = (p, \mathbf{u}_i, \mathbf{v}_i, \mathbf{N})$, where $\mathbf{u}_i$ is a unit vector collinear with the projection of $p_i - p$ in the plane orthogonal to $\mathbf{N}$, and $\mathbf{v}_i = \mathbf{N} \times \mathbf{u}_i$. Redundant computations of the geometry descriptors are done based on each adapted frame ϕ_i, next the extreme values are trimmed and from the rest of the descriptors, and by averaging, estimates of the final geometric descriptors at the point are obtained. Multiple geometry pipelines can be run in parallel for the different components. An extensive study and comparison of the

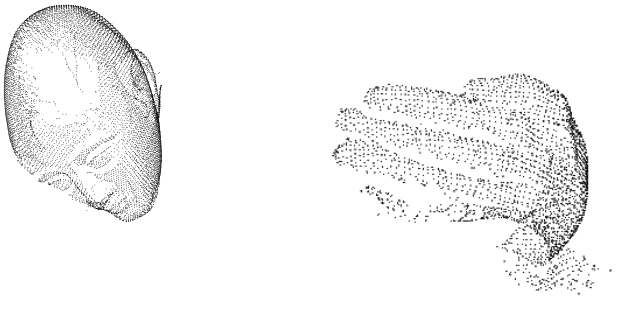

Fig. 7. Orientation-Topology Output: The two largest 2D components of the cloud from Figure 2. Together they contain 12182 points out of the total 12692 points in the scene reconstructed by the Data Pipeline.

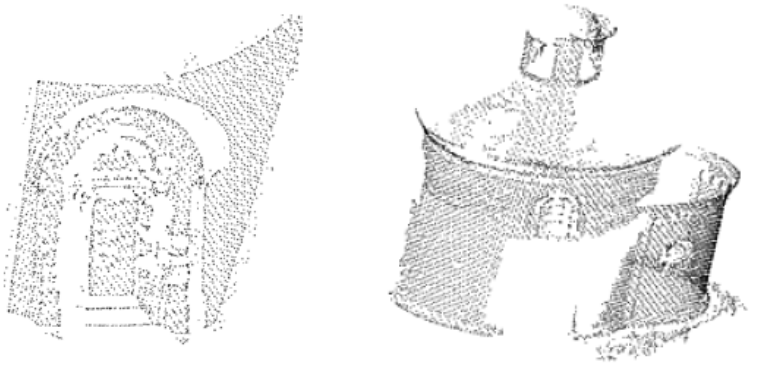

Fig. 8. Orientation-Topology Output: A high-resolution oriented 2D component (left) and a lower resolution component (right) extracted from the cloud in Figure 3. The high resolution component contains 3398 points and the low resolution contains 7012. The total size of the cloud produced by the Data Pipeline is 12499 points.

methods for computing curvatures has been reported [16,35], and here for the purpose of illustration we show two families of principal curvature directions. See Figure 6.

3 Summary

The paper presents a methodology for recovering the geometry of a 3D scene from a set of unordered, uncalibrated images. The images are fed into an automatic data processing pipeline which produces a collection of fish-scales; the fish-scales the are presented as input in an automatic pipeline which assigns orientation and topology, segments the cloud into connected manifold components, and recovers the local surface geometry descriptors at each surface point. The data pipeline, the recovery of the topology of an oriented cloud and the method for computing the geometric surface descriptors have been introduced previously and their performance had been analyzed [8,16,18,24]. The results presented here are for illustrative purposes. The theoretical contributions presented are limited to the iterative procedure that simultaneously recovers orientation, topology and segments the cloud into manifold components. We are conducting a performance evaluation and comparison study focusing on the orientation and scale selection. The results are the subject of a forthcoming paper.

References

1. M. Alexa, J. Behr, D. Cohen-Or, S. Fleishman, D. Levin, C. T. Silva, "Computing and Rendering Point Set Surfaces", *Trans. Vis. Comp. Graph.*, Volume 9(1): 3-15, 2003.
2. Nina Amenta and Yong Joo Kill, "Defining Point-Set Surfaces", ACM Trans. on Graphics, vol. 23(3), 264-270, Special Issue: Proceedings of SIGGRAPH 2004.

3. E. Boyer, Petitjean, S., "Regular and Non-Regular Point Sets: Properties and Reconstruction", Comp. Geometry, vol. 19, 101-131, 2001.
4. R. Hartley, "Theory and Practice of Projective Rectification," Int J Computer Vision, 35(2), 115-127, 1999.
5. R. Hartley, R. and Zisserman, A., "Multiple View GeomeLtry in Computer Vision", Cambridge University Press, 2000.
6. O. Chum, Matas, Jiří, and Kittler, J., "Locally optimized RANSAC," Proc DAGM, 236-243, 2003
7. O. Chum, T. Werner, and J. Matas. Two-view geometry estimation unaffected by a dominant plane. In *CVPR*, vol. 1, pp. 772–779, 2005.
8. H. Cornelius, Śara, R., Martinec, D., Pajdla, T., Chum, O., Matas, J., "Towards Complete Free-Form Reconstruction of Complex 3D Scenes from an Unordered Set of Uncalibrated Images," SMVP/ECCV 2004, vol. LNCS 3247, (2004) pp. 1-12.
9. K. Cornelis, F. Verbiest, and L.J. Van Gool. Drift detection and removal for sequential structure from motion algorithms. *PAMI*, 26(10):1249–1259, October 2004.
10. T. K. Dey, Sun, J., "Extremal Surface Based Projections Converge and Reconstruct with Isotopy", Tech. Rep. OSU-CISRC-05-TR25, Apr. 2005.
11. H. Edelsbrunner, Mucke, E.P., "Three-dimensional alpha shapes," ACM Trans. Graph., Volume 13(10, 43-72, 1994.
12. M. Gopi, Krishnan, S., Silva, C. T., "Surface reconstruction based on lower dimensional localized Delaunay triangulation", EUROGRAPHICS 2000, Computer Graphics Forum, vol 19(3), 2000.
13. J.-Y. Guillemaut, Drbohlav, O., Sara, R., Illingworth, J., "Helmholtz Stereopsis on rough and strongly textured surfaces ", 2nd Intl. Symp. 3DPVT, Greece, 2004.
14. H. Hoppe, DeRose, T., Duchamp, T., McDonald, J., Stuetzle, W. "Surface reconstruction from unorganized points", *Comp. Graph. (SIGGRAPH '92 Proceedings)*, 26, 71–78, (1992)
15. F. Kahl. Multiple view geometry and the l_∞-norm. In *ICCV05*, pp. II: 1002–1009, 2005.
16. G. Kamberov, G. Kamberova, "Conformal Method for Quantitative Shape Extraction: Performance Evaluation", ICPR 2004, Cambridge, UK, August, 2004, IEEE Proceedings Series, 2004
17. G. Kamberov, G. Kamberova,"Topology and Geometry of Unorganized Point Clouds", 2nd Intl. Symp. 3D Data processing, Visualization and Transmission, Thessaloniki, Greece, IEEE Proc. Series, in cooperation with Eurographics and ACM SIGGRAPH, 2004.
18. G. Kamberov, Kamberova, G., Jain, G., "3D shape from unorganized 3D point clouds", Intl. Symposium on Visual Computing, Lake Tahoe, NV, Springer Lecture Notes in Computer Science, 2005.
19. G. Kamberov, P. Norman, F. Pedit, U. Pinkall, "Quaternions, Spinors, and Surfaces", American Mathematical Society , 2002, ISBN 0821819283.
20. J. Kostková and Šára, R. "Stratified Dense Matching for Stereopsis in Complex Scenes," Proc BMVC, Vol. 1, 2003.
21. J. Kostková and Šára, R. "Dense Stereomatching Algorithm Performance for View Prediction and Structure Reconstruction," Proc SCIA, 101-107, 2003.
22. J. Lalonde, R. Unnikrishnan, N. Vandapel, and M. Hebert "Fifth Scale Selection for Classification of Point-sampled 3-D Surfaces," Proc. International Conference on 3-D Digital Imaging and Modeling (3DIM 2005), 2005.
23. D. Martinec, and Pajdla, T. "3D Reconstruction by Fitting Low-rank Matrices with Missing Data," Proc CVPR 2005, Vol. I, pp. 198-205.

24. D. Martinec and T. Pajdla. 3D reconstruction by gluing pair-wise euclidean reconstructions, or "how to achieve a good reconstruction from bad images". In *3DPVT*, p. 8, University of North Carolina, Chapel Hill, USA, June 2006.
25. D. Martinec, demo at http://cmp.felk.cvut.cz/~martid1/demo3DPVT06.
26. J. Matas, Š. Obdržálek, and O. Chum. Local affine frames for wide-baseline stereo. In *ICPR(4)*, pp. 363–366, 2002.
27. K. Mikolajczyk et al. A Comparison of Affine Region Detectors. *IJCV*, 2005.
28. N. Mitra, N. Nguyen, and L. Guibas "Estimating surface normals in noisy point cloud data," International Journal of Computational Geometry & Applications, Vol. 14, Nos. 4&5, 2004, pp. 261–276.
29. D. Nistér. An efficient solution to the five-point relative pose. *PAMI*, 26(6):756–770, June 2004.
30. M. Pollefeys et al. Image-based 3d recording for archaeological field work. *CG&A*, 23(3):20–27, May/June 2003.
31. R. Šára, "Accurate Natural Surface Reconstruction from Polynocular Stereo," Proc NATO Adv Res Workshop Confluence of Computer Vision and Computer Graphics, NATO Science Series No. 84, pp. 69-86, Kluwer, 2000.
32. R. Šára and Bajcsy, R., "Fish-Scales: Representing Fuzzy Manifolds", *Proc. Int. Conf. on Computer Vision*, Bombay, India, Narosa Publishing House, 1998.
33. K. Sim and R. Hartley. Recovering camera motion using l_∞ minimization. In *CVPR*, vol. 1, pp. 1230–1237, New York , USA, June 2006.
34. H. Stewénius, D. Nistér, F. Kahl, and F. Schaffalitzky. A minimal solution for relative pose with unknown focal length. In *CVPR*, vol. 2, pp. 789–794, 2005.
35. T. Surazhsky, Magid, E., Soldea, O., Elber, G., and Rivlin, E., "A comparison of Gaussian and mean curvatures estimation methods on triangular meshes", 2003 IEEE Interntl. Conf. on Robotics & Automation (ICRA2003)