

Surface Reconstruction of Scenes Using a Catadioptric Camera

Shuda Yu and Maxime Lhuillier

LASMEA UMR 6602, UBP/CNRS, Campus des Cézeaux, 63177 Aubière, France
<http://www.lasmea.univ-bpclermont.fr/Personnel/Maxime.Lhuillier/>

Abstract. This paper presents a method to reconstruct a surface from images of a scene taken by an equiangular catadioptric camera. Such a camera is convenient for several reasons: it is low cost, and almost all visible parts of the scene are projected in a single image. Firstly, the camera parameters and a sparse cloud of 3d points are simultaneously estimated. Secondly, a triangulated surface is robustly estimated from the cloud. Both steps are automatic. Experiments are provided from hundreds of photographs taken by a pedestrian. In contrast to other methods working in similar experimental conditions, ours provides a manifold surface in spite of the difficult (passive and sparse) data.

1 Introduction

The automatic 3d modeling of scenes from image sequence is still an active area of research. In our context where the scene is an environment and the view points are in the neighborhood of the ground (not aerial images), the use of a wide view field camera is a natural choice since we should reconstruct everything which is visible around the view points. There are two steps: geometry estimation and surface estimation. The former estimates the camera parameters in conjunction with a sparse cloud of 3d points of the scene surface. The camera parameters are the successive 6D poses (rotation + translation) where the images are taken, and intrinsic parameters which define the projection function of the camera. The latter estimates a list of textured triangles in 3d from the images. In the ideal case, the triangle list is a manifold surface which approximates the scene surface. “Manifold surface” means that every point in the list has a neighborhood which has the disk topology. Thus the triangle list has neither hole nor self-intersection and it partitions the 3d space in “inside” and “outside” regions.

Now, our approach is compared with previous work. An exhaustive survey is outside the paper scope and we focus on the most close approaches. In contrast to the majority of other reconstruction systems, we don’t try to reconstruct a dense set of points using dense stereo methods. We think that a well chosen and sparse set of features is enough in many cases and for several applications such as interactive walkthrough and vehicle localization. We also note that only a small number of reconstruction systems use a catadioptric camera (a convex mirror of revolution mounted in front of a perspective camera) and nothing more; the

most used acquisition hardware is defined by one or several perspective camera(s) pointing in several directions like the Ladybug [16].

A 3d modeling method of scene using a catadioptric camera was developed. Firstly, the camera parameters (poses and intrinsic parameters) and a sparse cloud of 3d points are estimated [11]. Then a quasi-dense reconstruction is done and approximated by triangles [12]. However, these triangles are not fully connected and the resulting 3d models have holes. Comparing with this method, ours uses the same geometry estimation but a different surface estimation which provides a manifold surface. We describes these two steps in Sections 2 and 3, respectively.

An accurate surface reconstruction method from fully calibrated camera was developed [7]. A great number of interest points are matched (in this context, a non negligible part of them have low accuracy or even are false positive). Then, a method based on 3d Delaunay of the reconstructed points and on optimization (Graph-Cut) of all point visibilities (similar to [8]) is used to obtain an approximation of the surface. Last, the surface is refined by minimizing a global correlation score in the images. In practice, results are provided on sequences with a reduced number of images (dozens of images). In this work, no information is provided on how to obtain a manifold surface: the second optimization (correlation) needs the manifold property but the first optimization (visibility) does not enforce this property. We also use a 3d Delaunay of the reconstructed points and optimize the point visibilities, but our 3d point cloud is sparser and more accurate (it is provided by bundle adjustment of Harris interest points).

A reconstruction system based on a costly hardware mounted on a car was also developed [17]. It involves several perspective cameras pointing in several directions, accurate GPS/INS (global positioning system + inertial navigation system). The approach is briefly summarized as follows. Firstly, successive poses are estimated using Kalman fusion of visual reconstruction, INS and GPS. Secondly, interest points are detected and tracked in the images; then a sparse cloud of 3d points is obtained. Third, this cloud is used to select planes in 3d, which are used to drive a denser reconstruction. Fourth, the obtained 3d depth maps are merged by blocks of consecutive images. Last, a list of triangles which approximate the dense 3d information is generated. This approach is incremental and real-time, it allows reconstruction of very long video sequences. However, the resulting surface is not manifold since the triangles are not connected. This problem could be corrected by using a merging method such as [4] followed by a marching cube [13], but this is not adequate to large scale scene since it requires a regular subdivision of space into voxels. For this reason (and other reasons mentioned in [8]), an irregular subdivision of space into tetrahedra is more interesting for large scale scene.

A few works renounce to reconstruct a dense cloud of points for 3d scene modeling. In [15] and [14], the main goal is real-time reconstruction. Both papers reconstruct a sparse cloud of points, add them in a 3d Delaunay triangulation, select the “outside” tetrahedra thanks to visibility (a tetrahedron is “outside” if it is between a 3d point and a view point where the point is seen). The

remaining tetrahedra are “inside” and the surface result is the list of triangles between inside and outside tetrahedra. Both works are experimented on very simple scenes and their surfaces are not guaranteed to be manifold.

Note that a lot of Delaunay-based surface reconstruction methods exist [2] to reconstruct a (manifold) surface from an unorganized point cloud, but these methods require a dense enough cloud and ignore the visibility constraints provided by a geometry estimation step. In our context where visibility constraints are used, the list of adequate methods is reduced to [5,8,15,7,14]. Among these works, only [5] gives some details on how to obtain a manifold surface, but this is experimented on very small input cloud.

2 Geometry Estimation

Here we introduce the catadioptric camera and summarize problems to be solved in the catadioptric context: matching, geometry initialization and refinement.

2.1 Catadioptric Camera

Our low cost catadioptric camera is obtained by mounting a mirror of revolution in front of a perspective camera (the Nikon Coolpix 8700) thanks to an adapter ring. We assume that the catadioptric camera has a symmetry axis and the scene projection is between two concentric circles (Fig. 1). The camera model defines

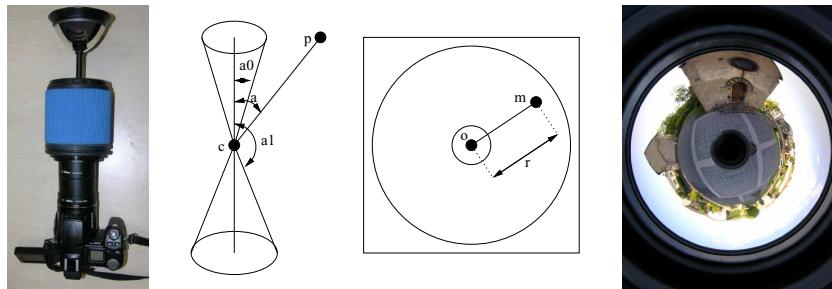


Fig. 1. From left to right: catadioptric camera, camera model (view field and image), and catadioptric image. The camera model has center \mathbf{c} , symmetry axis (vertical line), view field bounded by two angles a_0 and a_1 . Point \mathbf{p} with ray angle a (such that $a_0 \leq a \leq a_1$) is projected to \mathbf{m} with radius r between the two concentric circles.

the projection function from the camera coordinate system to the image. Our model is central (all observation rays of the camera go through a point called the centre) with a general radial distortion function. This distortion function is a polynomial which maps the ray angle a (between the symmetry axis and the line between a 3d point and the centre) to the image radius r (the distance between the point projection and the circle center in the image). This model is a Taylor-based approximation which simplifies the geometry estimation.

During a sequence acquisition, the camera is hand-held and mounted on a monopod such that the symmetry axis is (roughly) vertical. The mirror [1] is designed such that the distortion function is linear (equiangular camera) with a very large view field: 360 degrees in the horizontal plane and about 40 degrees below and above. In practice, the user alternates a step forward in the scene and a shot to get a sequence of photographs. We don't use video (although the acquisition is more convenient) since the video hardware under similar view field and image quality is more costly.

2.2 Matching

Here we explain how to match image points detected in two successive images of the sequence. According to the acquisition process described in Section 2.1, the camera motion is a sequence of translations on the ground and rotations around axes which are (roughly) vertical. In this context, a high proportion of the image distortion (due to camera motion) is compensated by image rotation around the circle center. The Harris point detector is used since it is invariant to such image rotation and it has a good detection stability. We also compensate for the rotation in the neighborhood of the detected points before comparing the luminance neighborhood of two points using the ZNCC score (Zero Mean Normalized Cross Correlation). Here a list of point pairs matched in the two images is obtained. Then the majority of image pixels are progressively matched using match propagation [10]. Last the list is densified: two Harris points satisfying the propagation mapping between both images are added to the list.

2.3 Geometry Initialization

Firstly, the radial distortion function (Section 2.1) which maps the ray angle to the image radius is initialized. On the one hand, approximate values of the two angles which define the view field (above and below the horizontal plane) are provided by the mirror manufacturer. On the other hand, the two circles which bound the scene projection are detected in images. Since the two angles are mapped to the radii of the two circles, we initialize the radial distortion function by the linear function which links these data. Second, the ray directions of the matched Harris points are estimated thanks to the calibration. Third, the 2-view and 3-view geometries (camera poses and 3d points) of consecutive images pairs are robustly estimated from the matched ray directions and RANSAC applied on minimal algorithms (more details in [11]). Fourth, we estimate the whole sequence geometry (camera poses and 3d points) using adequate bundle adjustments (BAs) applied in a hierarchical framework [11]. Remind that BA is the minimization of the sum of squared reprojection errors by the (sparse) Levenberg-Marquardt method [6].

2.4 Geometry Refinement

As mentioned in Section 2.1, our camera model is an approximation. Furthermore, the two angles used for the initialization above are also approximations

of the true angles (they depend on the unknown pose between mirror and the perspective camera). For these reasons, our current model should be refined. Therefore the linear radial distortion polynomial is replaced by a cubic polynomial whose the four coefficients should be estimated [11]. Then we apply a last BA to refine simultaneously all camera poses, 3d points and the four coefficients. A 2D point is involved in BA iff its reprojection error is less than a threshold (2 pixels).

3 Surface Estimation

Firstly, the link between the 3d Delaunay triangulation and the surface estimation problem is described. Then we explain how to obtain a manifold surface from the data provided by the geometry estimation in Section 2. These data are the cloud P of 3d points \mathbf{p}_i , the list C of view points \mathbf{c}_j (3d location of images), and the visibility lists V_i of \mathbf{p}_i (i.e. \mathbf{p}_i is reconstructed from the view points \mathbf{c}_j such that $j \in V_i$). The surface estimation has four steps: 3d Delaunay, Ray-Tracing, Manifold Extraction, and Post-Processing.

3.1 From 3d Delaunay Triangulation to Surface Estimation

Let T be the 3d Delaunay triangulation of P . Remind that T is a list of tetrahedra which “partitions” the convex hull of P such that the vertices of all tetrahedra are P . By “partitions”, we means that the union of tetrahedra is the convex hull and the intersection of two tetrahedra t_0 and t_1 is either empty or a t_0 vertex or a t_0 edge or a t_0 triangle (i.e. a t_0 facet). Here are two useful properties of T [2]: (1) the edges of all tetrahedra roughly define a graph neighborhood of P in the different directions and (2) two different triangles of the tetrahedra are either disjoint, or have one vertex in common, or have two vertices (and its joining edge) in common.

Assume that P samples an unknown surface S_0 . We would like to approximate S_0 by a triangle list S whose vertices are in P . The denser sampling P of S_0 , the better approximation of S_0 by S . Since this approximation boils down to define connections between points of P which are neighbors on the surface, a possible approach is to search S as a subset of the facets of all tetrahedra of T . In this case, the triangles of S meet property (2) above. Now assume that (3) all vertices of S are regular. A vertex \mathbf{p} of S is regular if the edges opposite to \mathbf{p} in the triangles of S having \mathbf{p} as vertex form a simple polygon (Fig. 2). S is a manifold surface since it meets (2) and (3).

3.2 3d Delaunay

As suggested in Section 3.1, the first step is the 3d Delaunay triangulation of P . Remind that \mathbf{p}_i has very bad accuracy if it is reconstructed in degenerate configuration: if \mathbf{p}_i and all $\mathbf{c}_j, j \in V_i$ are (nearly) collinear [6]. This case occurs in part of the camera trajectory which is a straight line and if points reconstructed

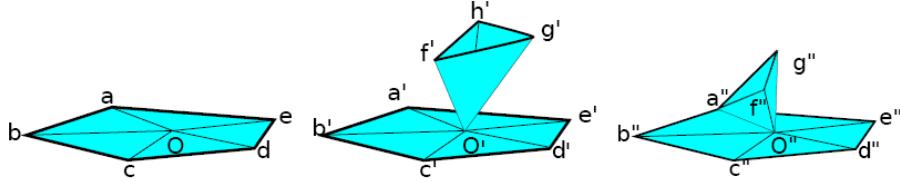


Fig. 2. O is regular since the edges opposite to O define a simple polygon $abcdea$. O' and O'' are not regular since polygons $a'b'c'd'e'a' - f'g'h'f'$ and $a''b''c''d''e''a''f''g''a''$ are not simple (the former is not connected, the latter has multiple vertex a'').

from this part are in the “neighborhood” of the straight line. Thus, P is filtered before Delaunay: we remove \mathbf{p}_i from P if all angles $\mathbf{c}_j \mathbf{p}_i \mathbf{c}_k$ ($j, k \in V_i$) are less than a threshold ϵ . It is also possible to improve the final S by adding “artificial points” in P which have empty visibility lists (more details are given in Section 4).

3.3 Ray-Tracing

Now we use the visibility information to segment T in “outside” and “inside” tetrahedra. Note that a tetrahedron is 100% inside or 100% outside since our target surface is included in the facets of all tetrahedra of the partition T . A tetrahedron which is intersected by ray (line segment) $\mathbf{c}_j \mathbf{p}_i, j \in V_i$ is outside since point \mathbf{p}_i is visible from view point \mathbf{c}_j . In practice, all tetrahedra are initialized inside and we apply ray-tracing for each available ray to force outside all tetrahedra intersected by the ray.

In our catadioptric context where points are reconstructed in almost all directions around view point, the view points are in the convex hull of P . This implies that the region outside the convex hull of P can not be intersected by ray and this region is classified inside. For implementation convenience [3] (obtain a complete graph of tetrahedra), tetrahedra are added to establish connections between the “infinite point” and the facets of the convex hull. These tetrahedra are classified inside although they have no physical volume in 3d.

3.4 Manifold Extraction by Region Growing

At first glance, S could be defined by the list of triangles separating the inside and the outside tetrahedra. Unfortunately, the experiment shows that S is not manifold because it has vertices which are not regular (Section 3.1).

It is suitable to change the outside and inside definitions for the manifold extraction: *outside* becomes *intersected*, *inside* becomes *non-intersected*, and now the *outside* tetrahedra form a sub-list O of *intersected* tetrahedra such that its border S (a list of triangles) is manifold. The tetrahedra which are not *outside* are *inside*. These new definitions apply in the sequel of the paper.

Growing one tetrahedron at once. The manifold extraction is a region growing process: we progressively add to O new tetrahedra, which were inside

and intersected, such that the border S of O is continually manifold. Region growing is a convenient way to guarantee the manifold constraint, but the final O depends on the insertion order of the tetrahedra in O . Indeed, a tetrahedron in O which has vertices in S enforces manifold constraints on these vertices, which are shared by other tetrahedra which are not (or not yet) in O .

To reduce the manifold constraints, we choose the new tetrahedron such that it has at least one facet included in S (i.e. it is in the immediate neighborhood of O). We also choose a priority score for each intersected tetrahedron to define the insertion order: the number of rays which intersect the tetrahedron. The tetrahedra in the neighborhood of O are stored in a priority queue for fast selection of the tetrahedron with the largest priority.

Growing several tetrahedra at once. Note that the region growing above does not change the initial topology of O . Here O is initialized by the tetrahedron with the largest score and it has the ball topology. This is a problem if the true outside space has not the ball topology, e.g. if the camera trajectory contains closed loop(s) around building(s). In the simplest case of one loop, the true outside space has the toroid topology and the computed outside space O can not close the loop (left of Fig. 3).

We correct this kind of problem with the following region growing. Firstly, we find a vertex on S such that all inside tetrahedra which have vertex S are intersected. Then, we force all these tetrahedra to outside. If all vertices of these tetrahedra are regular, the region growing is successful and O is increased. In the other case, we restore these tetrahedra to inside. In practice, we alternate “one tetrahedron at once” and “several tetrahedra at once” region growings until no tetrahedron may be added in O .

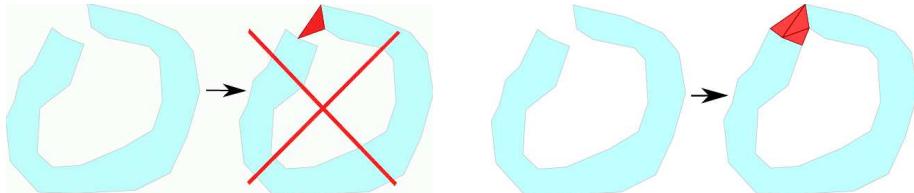


Fig. 3. Left: adding one tetrahedron at once in O (in light blue) can not close the loop due to non regular vertex. Right: adding several tetrahedra at once close the loop.

3.5 Post-Processing

Although the surface S provided by the previous steps is manifold, it has several weaknesses which are easily noticed during visualization. Now we list these weaknesses and explain how to remove (or reduce) them using prior knowledge of the scene.

A “peak” is a vertex p_i on S such that the ring of its adjacent triangles in S defines a solid angle w which is too small (or too large) to be physically

plausible, i.e. $w < w_0$ or $w > 4\pi - w_0$ where w_0 is a threshold. We try to remove peak \mathbf{p}_i from S as follows. We consider the acute side of the peak (inside or outside tetrahedra) and reverse its status (inside becomes outside, and vice versa). The removal is successful if all vertices of the reversed tetrahedra are regular. Otherwise, the reversed tetrahedra are reversed a second time to go back in the previous configuration and we try to remove another peak. In practice, we go through the list of S vertices several times to detect and remove the peaks.

The surface S is smoothed to reduce the reconstruction noise. Let \mathbf{p} be the concatenation vector of all \mathbf{p}_i in S . We apply a mesh smoothing filter $\mathbf{p} \leftarrow \mathbf{p} + \Delta\mathbf{p}$ where $\Delta\mathbf{p}$ is a discrete laplacian defined on the mesh vertices [18].

Up to now, S is closed and contains triangles which correspond to the sky (assuming outdoor image sequence). These triangles should be removed since they do not approximate a real surface. They also complicate the visualization of the 3d model from a bird's-eye view. Firstly, the upward vertical direction \mathbf{v} is robustly estimated assuming that the camera motion is (roughly) on a horizontal plane. Secondly, we consider open rectangles defined by the finite edge $\mathbf{c}_i\mathbf{c}_{i+1}$ and the two infinite edges (half lines) starting from \mathbf{c}_i (or \mathbf{c}_{i+1}) with direction \mathbf{v} . A triangle of S which is intersected by an open rectangle is a sky triangle and is removed from S . Now S has hole in the sky. Lastly, the hole is increased by propagating its border from triangle to triangle such that the angle between triangle normal (oriented from outside to inside) and \mathbf{v} is less than threshold α_0 .

Last, the texture should be defined for each triangle of S . We use a simplified version of [9], which gets "as is" the texture of a well chosen view point \mathbf{c}_j for each triangle t of S . In our case where the image sequence has hundreds of images (not dozens), we pre-select a list of candidate \mathbf{c}_j for t as follows: t is entirely projected in the \mathbf{c}_j image (large t are splitted), t is not occluded by an other triangle of S , \mathbf{c}_j provides one of the k -largest solid angle for triangle t .

4 Experiments

The acquisition set up is described at the end of Section 2.1. Our sequence has 208 images taken along a full turn around a church. The trajectory length is about $(25 \pm 5\text{cm}) \times 208 = 52 \pm 10\text{m}$ (the exact step lengths between consecutive images are unknown). The radii of large and small circles of the catadioptric images are 563 and 116 pixels.

The geometry estimation step (Section 2) reconstructs 76033 3d points from 477744 2D points (inliers). The RMS error of the final bundle adjustment is 0.74 pixels. The estimated view field angles (Fig. 1) are $a_0 = 41.5$ and $a_1 = 141.7$ degrees; the angles provided by the mirror manufacturer are $a_0 = 40$ and $a_1 = 140$ degrees. Fig. 4 shows images of the sequence and the result of this step.

Then the surface is estimated (Section 3) using $\epsilon = 5$ degrees, $w_0 = \pi/2$ steradians, $\alpha_0 = 45$ degrees, and $k = 10$. Fig. 5 explains the advantages of the steps of our method. Row 1 shows that the reconstructed points can not be used "as is" as vertices of the surface. The surface fairing (peak removal and surface smoothing) is necessary. Remind that a 3d point may be inaccurate for several reasons: (1) if

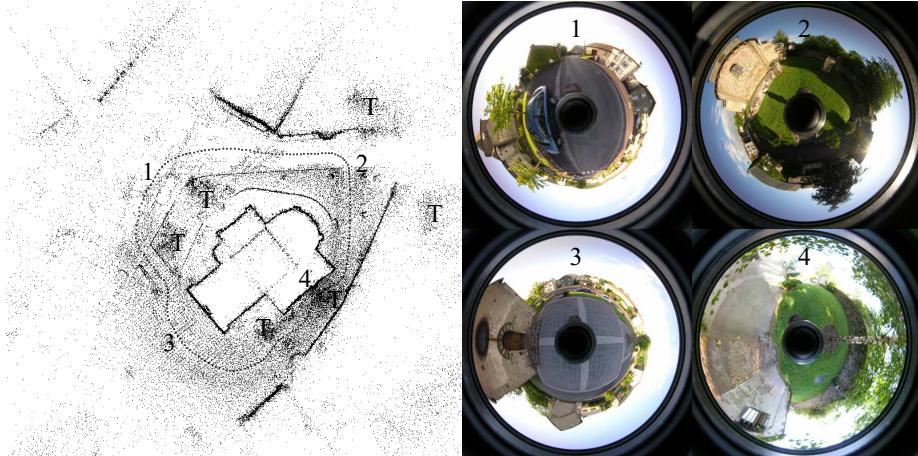


Fig. 4. Top view of the geometry estimation result (left) and four catadioptric images (right) for the church sequence. The top view includes 76033 points, 208 poses around the church, numbers for the locations of the four images, “T” for tree locations around the camera trajectory, and numbers for the image locations.

it has large depth (it is reconstructed from far camera poses), (2) if it has small depth (the central approximation of the true non-central camera model provides bad accuracy in our context) and (3) image noise. Row 2 shows top views of the surface before (left) and after (right) the sky removal. On the left, the surface is closed and we see the sky triangles. On the right, the surface can be visualized at a bird’s eye view. Note that the current version of the sky removal process is very simple; it should be improved to remove large triangles on the top of the model (including those which connect trees and the church). The left of row 3 shows that the surface forms a blind alley if the “several tetrahedra at once” region growing is not used (the pedestrian can not go from location 3 to location 4, see Fig. 4). The right of row 3 shows that the loop is closed around the church if this region growing is used (the pedestrian can go from location 3 to location 4). Row 4 shows that the “several tetrahedra at once” region growing is also very useful to improve the outside space estimated by the “one tetrahedron at once” region growing. However, both region growings are not enough to avoid problems as the one shown on the left of row 5: there is an ark which connects a vertical surface to the ground, although all tetrahedra which define the ark are intersected by rays. We tried to solve this problem by changing the priority score of the tetrahedra, but arks always appear somewhere. Here we greatly reduce the risk of arks thanks to a simple method. In the 3d Delaunay step, “artificial points” are added in P such that the long tetrahedra potentially involved in arks are splitted in smaller tetrahedra. The artificial points are added in a critical region for visualization: the immediate neighborhood of the camera trajectory. Technical

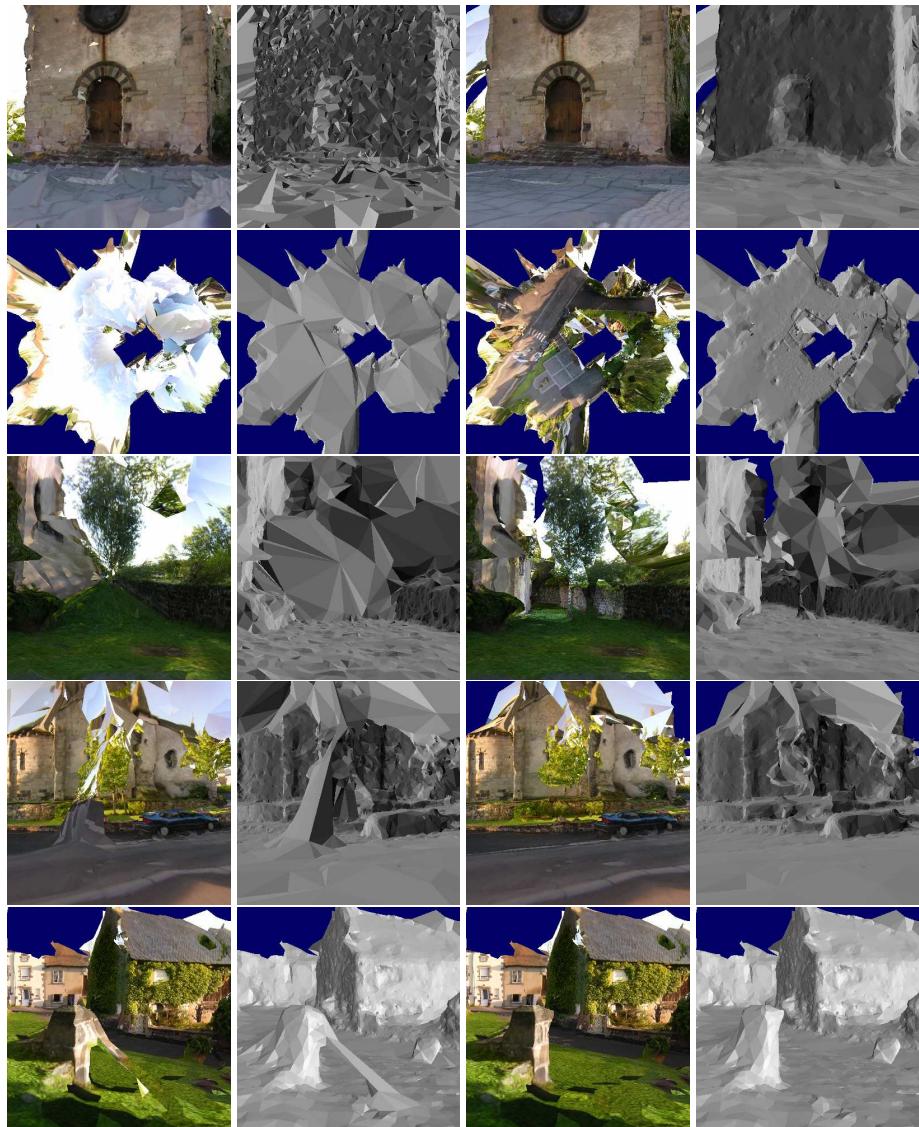


Fig. 5. Row 1: local view of the church 3d model without (left) or with (right) peak removal and surface smoothing (Section 3.5). Row 2: top view without (left) or with (right) the sky removal (Section 3.5). Rows 3 and 4: without (left) or with (right) the “several tetrahedra at once” region growing (Section 3.4). Row 5: without (left) or with (right) the artificial points (Section 3.2). In all cases, gray levels encode the triangle normals.

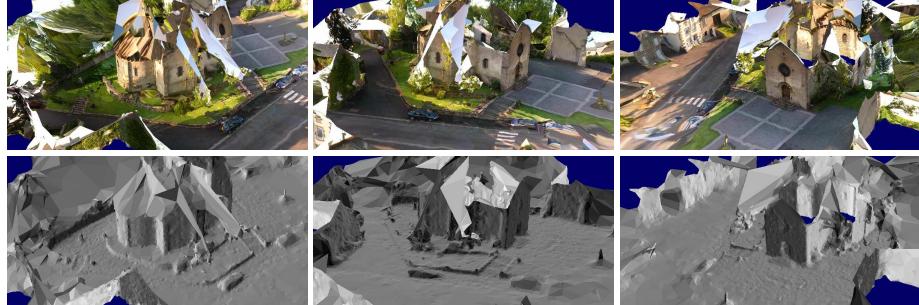


Fig. 6. Bird's-eye views of the church 3d model obtained with the complete method

details are described in the appendix for the paper clarity. The right of row 5 shows that ark is removed thanks to the artificial points. Fig. 6 shows other views of the church obtained with the complete method.

The 3d Delaunay has 59780 vertices and is reconstructed using the incremental 3d Delaunay of CGAL [3]. During the ray tracing step, 191947 tetrahedra are intersected by 398956 rays. The manifold extraction step by region growing provides a surface with 94228 triangles and 162174 outside tetrahedra. The ratio between outside and intersected tetrahedra is 86% (the ideal value is 100%). Lastly, 898 sky triangles are removed. The surface estimation (without texturing) takes about 30 seconds on a Core 2 Duo E8500 at 3.16 GHz. A complete walkthrough around the church is provided in a **mpeg video** available at www.lasmea.univ-bpclermont.fr/Personnel/Maxime.Lhuillier. This video is cyclic and includes the sky since the 3d model is viewed from a pedestrian's-eye view

5 Conclusion

The proposed method has two steps: geometry estimation and surface estimation. We briefly summarize the former and focus on the latter. Our results contrast with the previous ones since we are able to provide manifold surface (up to sky removal) from a reconstructed sparse cloud of points (instead of dense) in non trivial cases. The current system is able to reconstruct the main components of outdoor scene (ground, buildings, dense vegetation) and allows interactive walkthrough. Technical improvements are possible for several steps (surface fairing, sky removal and texture mapping). Future work includes the integration of edges in the 3d model, a real-time version of the surface estimation step, and a specific process for thin structures such as trunks and electric posts.

Appendix

The artificial points have empty visibility lists, their number is equal to 0.5% of the number of the reconstructed points, and they are randomly and uniformly

added in the immediate neighborhood N of the camera trajectory. We define N by the union of half-balls (southern hemispheres such that the north direction is \mathbf{v}) centered on all \mathbf{c}_j with radius $r = 10 \text{ mean}_j \|\mathbf{c}_{j+1} - \mathbf{c}_j\|$.

References

1. <http://www.kaidan.com>
2. Cazals, F., Giesen, J.: Delaunay triangulation based surface reconstruction: ideas and algorithms. INRIA Technical Report 5394 (2004)
3. <http://www.cgal.org>
4. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH (1996)
5. Faugeras, O., Le Bras-Mehlman, E., Boissonnat, J.D.: Representing stereo data with the delaunay triangulation. Artificial Intelligence, 41–47 (1990)
6. Hartley, R., Zisserman, A.: Multiple view geometry in computer vision. Cambridge University Press, Cambridge (2000)
7. Hiep, V.H., Keriven, R., Labatut, P., Pons, J.P.: Towards high-resolution large-scale multi-view stereo
8. Labatut, P., Pons, J.P., Keriven, R.: Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and Graph-Cuts. In: The International Conference on Computer Vision (2007)
9. Lempitsky, V., Ivanov, D.: Seamless Mosaicing of Image-Based Texture Maps. In: The Conference on Computer Vision and Pattern Recognition (2007)
10. Lhuillier, M., Quan, L.: Match propagation for image based modeling and rendering. IEEE Trans. on Pattern Analysis and Machine Intelligence 24(8), 1140–1146 (2002)
11. Lhuillier, M.: Automatic scene structure and camera motion using a catadioptric system. Computer Vision and Image Understanding 109(2), 186–203 (2008)
12. Lhuillier, M.: A generic error model and its application to automatic 3d modeling of scenes using a catadioptric camera. International Journal of Computer Vision 91(2), 175–199 (2011)
13. Lorensen, W.E., Cline, H.E.: Marching cubes: a high resolution 3d surface construction algorithm. Computer Graphics 21, 163–169 (1987)
14. Lovi, D., Birkbeck, N., Cobzas, D., Jagersand, M.: Incremental free-space carving for real-time 3d reconstruction. In: 3DPVT 2010 (2010)
15. Pan, Q., Reitmayr, G., Drummond, T.: ProFORMA: probabilistic feature-based on-line rapid model acquisition. In: The British Machine Vision Conference (2009)
16. <http://www.ptgrey.com>
17. Pollefeys, M., Nister, D., Frahm, J.M., Akbarzadeh, A., Mordohai, P., Clipp, B., Engels, C., Gallup, D., Kim, S.J., Merell, P., Salmi, C., Sinha, S., Talton, B., Wang, L., Yang, Q., Stewenius, H., Yang, R., Welch, G., Towles, H.: Detailed real-time urban 3D reconstruction from video. International Journal of Computer Vision 78(2), 143–167 (2008)
18. Taubin, G.: A signal processing approach to fair surface design. In: SIGGRAPH (1995)