Hierarchical shape-based surface reconstruction for dense multi-view stereo

Patrick Labatut Jean-Philippe Pons Renaud Keriven
IMAGINE
ENPC/CSTB, LabIGM, Université Paris-Est
{labatut,pons,keriven}@imagine.enpc.fr

Abstract

The recent widespread availability of urban imagery has lead to a growing demand for automatic modeling from multiple images. However, modern image-based modeling research has focused either on highly detailed reconstructions of mostly small objects or on human-assisted simplified modeling. This paper presents a novel algorithm which automatically outputs a simplified, segmented model of a scene from a set of calibrated input images, capturing its essential geometric features.

Our approach combines three successive steps. First, a dense point cloud is created from sparse depth maps computed from the input images. Then, shapes are robustly extracted from this set of points. Finally, a compact model of the scene is built from a spatial subdivision induced by these structures: this model is a global minimum of an energy accounting for the visibility of the final surface.

The effectiveness of our method is demonstrated through several results on both synthetic and real data sets, illustrating the various benefits of our algorithm, its robustness and its relevance for architectural scenes.

1. Introduction

Reconstruction of 3D models from urban imagery has long been an active topic of research in computer vision and photogrammetry. Applications such as Google Earth or Microsoft Virtual Earth have allowed a broad audience to visualize large-scale models of cities with superimposed street level or aerial imagery. Still, the models are mostly handmade and automatic generation from images of such content is clearly desirable. Several methods for automatic or guided image-based modeling have been developed and the following sections give an overview of these various attempts.

1.1. Dense multi-view stereo

Dense multi-view stereo has received a lot of attention since the comparison of [26]. While the accuracy of the latest results is now challenging range data, few dense multi-view methods are appropriate for large scenes taken in a general setting and without silhouette information (*e.g.* large-scale outdoor scenes), see [31] for a recent evaluation. These approaches tend to produce overly complex meshes and trade a highly detailed reconstruction for the loss of characteristic geometric features of the scenes. The proposed approach is applicable to such scenes and directly outputs a simple shape-based model.

1.2. Geometry processing

While geometry processing techniques could be applied to simplify the output meshes of dense multi-view stereo pipelines, such combination is not only less efficient than the presented method, it is also less powerful and not adequate. First, mesh simplification (we refer the reader to [23]) is mainly suited to perfectly meshed, almost noise-free surfaces far from the typical output of multi-view reconstruction algorithms. Besides, simplification often requires user intervention for quality inspection. Recently, more involved mesh segmentation methods have appeared and demonstrated impressive results as a preliminary step to guide subsequent remeshing or simplification (see [27] for a survey). These techniques face the same problems with imperfect inputs. Our method implicitly combines segmentation and shape-based simplification during surface reconstruction. The acquisition process is accounted for throughout the pipeline while the mentioned post-processings are more likely to worsen initial reconstruction errors.

1.3. Automatic urban modeling

Dedicated methods such as [34] and [10] have been elaborated for architectural scenes. A few dominant planes are detected in a sparse Structure-from-Motion (SfM) point

cloud. These planes are used as a coarse shell on which parametrized models of architectural elements are then fitted. In contrast with general purpose dense multi-view stereo methods, these approaches depends on strong architectural cues and are limited to reconstruction of scenes where their numerous assumptions are practically verified.

1.4. Human-assisted image modeling

Human-assisted reconstruction was pioneered by [9]: from edges marked in the images by a user and selected simple primitives, the interface aligns the primitives with the edges. This initial effort was a source of inspiration for the development of commercial products such as Autodesk ImageModeler [1] and Google Sketchup [3] that exploits photometric cues to guide a simplified model reconstruction from images. [28] presented an easier approach for architectural scenes with abundant parallel lines, helping the user and constraining optimizations thanks to extracted lines and vanishing points. These methods do not require calibrated images and includes either a manual, an assisted or an automatic SfM step often relying again on strong architectural cues. In our case, if such information is unavailable, it can be recovered using a combination of computer vision techniques similar to [29].

1.5. Toward automatic compact modeling

Few authors have tried to address a similar problem as this paper, most of them dealing only with piecewise-planar scenes. [14] and [5] focus on the robust extraction of multiple planes applied to sparse SfM point clouds, extract a limited number of planes and are not designed to output simplified piecewise-planar dense reconstructions. On the other hand, [17] and [4] both output such reconstructions but have major restrictions. The former proposes a visibility-consistent interpolatory reconstruction from SfM clouds but can not deal with outliers. The latter exploits extracted edges but its mesh reconstruction uses less robust heuristics and only seems applicable to scenes with similar points of view.

In light of the previous analysis, we draw the conclusion that no satisfying, general enough method exists to automatically build compact shape-based models of scenes from images. Applications would include not only image-based compact modeling as demonstrated in this paper, but also reverse-engineering or shape and scene recognition, interpretation and indexing.

We propose a new surface reconstruction algorithm with strong shape priors, applicable to dense multi-view stereo. The presented approach combines three successive steps. First, a dense point cloud of the scene is generated from merged depth maps. Then, multiple shapes (of a predefined set of shapes) are robustly extracted from this point cloud and a hierarchical description of the scene is built. Finally, a partition of space induced by this scene description is computed. This subdivision of space has a number of desirable properties for our problem. By labeling the cells of this subdivision as inside or outside of the scene, a mesh representing the scene can be extracted as a subset of the subdivision facets. To this end, we define an energy on the space of such labelings that only accounts for the visibility of the fitted points. This energy is suitable to minimum s-t cuts optimization allowing a globally optimal surface to be generated from the space subdivision and the fitted points. We show how to extend the final step to output partly shape-based (hybrid) reconstructions combining shape elements in some areas with non-shape based parts. Finally, note that our approach is different from range segmentation methods [18], not applicable here: most techniques are limited to 2.5D data, can not deal with the amount of noise and outliers generated by dense stereo matching, and above all, do not reconstruct a piecewise-primitive surface mesh.

The paper is organized as follows: section 2 provides background on the required material to understand our contributions, section 3 describes our shape-based reconstruction pipeline, section 4 deals with implementation details and finally, in section 5, results on both synthetic and real data are shown and discussed.

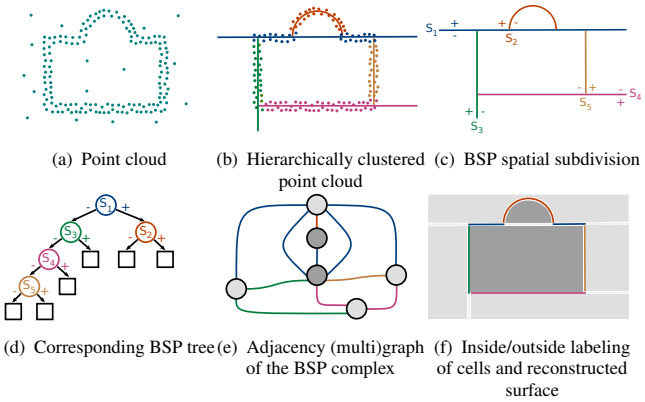
2. Background

2.1. Robust regression

A large body of work has been dedicated to robust regression. Most techniques widely used in computer vision can be seen as optimizing some objective function (number of bin votes for the Hough transform and number of inliers inside a band for RANSAC [12]) with an appropriate sampling of the model parameters (by discretizing the parameters space or by randomly sampling models supported by a minimal set of points). [30] points out the limitations of RANSAC when applied to data with multiple structures. While the Hough transform naturally handles such data, the bin size adjustment is delicate and the method faces inherent difficulties on noisy data. Extending random sampling methods for multiple structures regression is challenging: it requires dealing with other structures as outliers. While we do not pretend to solve the general problem of extracting multiple structures from noisy point cloud with outliers ([8, 32] are some recent attempts), in 3.2, we show however that exploiting additional information from our specific problem (the acquisition process and the geometry of the extracted shapes) improves the robustness of existing approaches beyond their traditional limits.

2.2. Generalized binary space partitioning trees

Originally developed to address the hidden-surface problem [15], a binary space partitioning tree (BSP tree for



(a) Point cloud (b) Hierarchically clustered point cloud (c) BSP spatial subdivision

(d) Corresponding BSP tree (e) Adjacency (multi)graph of the BSP complex (f) Inside/outside labeling of cells and reconstructed surface

Figure 1: ***Generalized BSP trees*** for hierarchical clustering of point clouds and boundary representation of piecewise-primitive surfaces.

short) is a versatile structure widely used both for spatial partitioning and boundary representation with particular applications in rendering, robot motion and path planning. A BSP tree is a binary tree defining a recursive partition of space into pairs of subspaces w.r.t. planes at arbitrary positions and orientations. Instead of planes only, any oriented hypersurface may be used to split the space into a negative and a positive half. Each node in the tree corresponds to a splitting hypersurface and each leaf to an unpartitioned area of space.

In section 3.2, a BSP tree will be used as a data partitioning structure to hierarchically cluster a point cloud into sets of shapes while in section 3.4, it will serve as a boundary representation for surface reconstruction. As shown in Fig. 1, a BSP tree induces a cell complex (a partition of space into cells). Each cell of this complex corresponds to a leaf of the BSP tree but one leaf of the tree may give rise to several cells[1]. Each facet of this complex is contained in one of the splitting surfaces. Two different cells may be linked by more than one facet since they may not be convex. By labeling each cell of this complex as inside or outside, a surface can be directly extracted from the complex (see Fig. 1(f)) and this surface is an assembly of patches from the various splitting surfaces of the BSP tree. The reconstruction of such piecewise-primitive surfaces and the implicit recovery of shape boundaries and vertices motivate this particular choice of spatial subdivision. Other advantages include the extension of detected shapes to less textured areas allowing the reconstruction to capture areas possibly missed by the depth maps generation. Finally, wrongly detected shapes do not significantly affect the complex (and the reconstruction) as they only split facets and cells.

[1]The space between two close parallel planes split by a large sphere corresponds to two leaves but three cells, for instance.

2.3. Surface reconstruction with minimum s-t cuts

Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be a *directed* graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$ with non-negative weights. Two special vertices, the source s and the sink t are distinguished as the terminals of this *network graph*. A s-t cut $(\mathcal{S}, \mathcal{T})$ is a partition of $\mathcal{V}$ into two disjoints sets $\mathcal{S}$ and $\mathcal{T}$ such that $s \in \mathcal{S}$ and $t \in \mathcal{T}$. The cost of an s-t cut is the sum of the weights of the edges from $\mathcal{S}$ to $\mathcal{T}$: $\mathrm{c}(\mathcal{S}, \mathcal{T}) = \sum_{(p,q)\in\mathcal{S}\times\mathcal{T}} w_{pq}$. According to the Ford-Fulkerson theorem [13], finding an s-t cut with minimum cost is the same as computing the maximum flow from the source to the sink. Efficient algorithms with low-polynomial complexity exist to solve this problem. With an adequate graph construction, many segmentation problems in computer vision can be solved by global minimization of the corresponding functional $\mathrm{c}(\mathcal{S}, \mathcal{T})$, provided the energy of the problem may be expressed in this framework [20]. While the optimization domain in computer vision has traditionally been a regular subdivision of some space, minimum s-t cuts on complexes were first introduced in [19] to globally optimize surface functionals. The use of sparse random complexes was proposed for their adaptivity over uniform grids. In 3.4 and 3.5, our optimization domain is a sparse graph derived from the adjacency graph of the cells of a complex, adaptive to point samples, and embedding the recovered shapes. Furthermore, in contrast with the *graph cuts* minimal surfaces of [7], our optimization problem is intrinsically discrete.

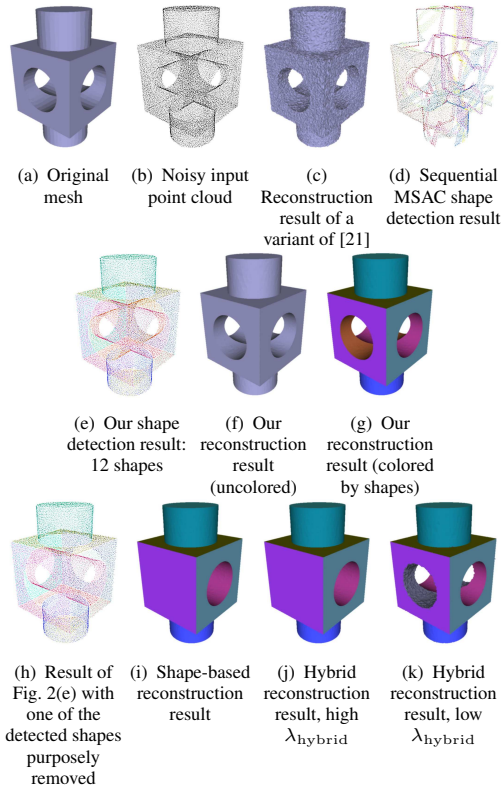
3. Reconstruction algorithm

This section details the different steps involved in our method: the dense point set generation, the extraction of shapes and corresponding hierarchical clustering of the point cloud, and the final piecewise-primitive surface reconstruction. In this paper, the considered classes of shapes are limited to planes, spheres, cones and cylinders but the overall algorithm may be generalized to any classes of oriented surfaces instantiable from a small number of points and for which point distances and local normals can be computed approximately.

3.1. Dense point cloud from depth maps

Initial sparse and downscaled depth maps are computed between pairs of input images: a simple geometric plane sweeping is used with a thresholded (multi-level) normalized cross-correlation matching score. The different points from all the depth maps are clustered according to their positions in the different camera frustums and in the images. The positions of the points are then locally refined to optimize their photo-consistency scores. The final result is a set of points each carrying a tuple of views where they were seen. Obviously, this step still generates a noisy point cloud with a decent amount of outliers. The next two steps of our

pipeline are designed to robustly handle this kind of input.



(a) Original mesh (b) Noisy input point cloud (c) Reconstruction result of a variant of [21] (d) Sequential MSAC shape detection result

(e) Our shape detection result: 12 shapes (f) Our reconstruction result (uncolored) (g) Our reconstruction result (colored by shapes)

(h) Result of Fig. 2(e) with one of the detected shapes purposely removed (i) Shape-based reconstruction result (j) Hybrid reconstruction result, high λ_{hybrid} (k) Hybrid reconstruction result, low λ_{hybrid}

Figure 2: ***block data set and results.***

3.2. Hierarchical detection of shapes

3.2.1 Single shape robust extraction

Our shape detection and fitting is based on the RANSAC framework of [12]: random shapes supported by minimal sets of points are enumerated to optimize an objective function counting the number of points (the shape inliers) inside a band around a shape instance. As recalled in 2.1, plain RANSAC is not suited for robust regression in data with multiple structures: Fig. 2(d) shows the catastrophic result of a sequence of successive shape extractions with a slight variant of RANSAC.

One of our contributions is to demonstrate that a combination of significant alterations to this shape optimization via random sampling (including geometric priors and exploiting the density of the point cloud near the surface) can make the approach robust to challenging point clouds especially from passive stereo. Before any shape detection occurs, a 3D k-D tree of the point cloud is computed to efficiently find the k nearest neighbors (k-NN for short) and estimate an oriented normal for each point by fitting a plane to these k-NN. For a densily sampled surface, the k-NN of a point near the surface are likely to also lie near the surface, leading to a more reliable normal estimation. The k-NN of an outlier point, however, are much more spatially spread and its normal will likely be incoherent with its neighbors'.

First, the random sampling of shapes is modified as follows to draw meaningful shapes: 1. The closer the points the higher the chance they are inliers to the same shape. Of the few points to be randomly selected to create a shape instance, the first one is drawn uniformly in the point cloud, while the next are uniformly drawn but only within a ball of small radius (a fixed multiple of the maximum inlier distance). This geometric ball search query can effectively be answered as a range query in the k-D tree of the points. This localized sampling follows NAPSAC [24] developed for high-dimensional robust estimation where an assumed distribution of inliers and outliers lead to this idea. 2. The point cloud from 3.1 is still noisy and random shapes supported by minimal sets of point may lead to systematic wrong hypotheses (with consequently a wrong optimal shape). An improved estimate of the instantiated random shape is obtained by locally refitting it to all the inlier points within the band restricted to the ball used to search for a minimal set.

The objective function to be minimized over all the enumerated shapes is changed in the following ways: 1. It is based on the MSAC variant [33] of RANSAC and penalizes inliers according to their distances to the instantiated surface. 2. The acquisition process intervenes in the inliers counting procedure (and in the shape sampling): a point (or one of the points to instantiate a shape) is considered an inlier only if its visibility information agrees with the instantiated shape, i.e. if the local normal to this shape does not make a wide angle with any of the lines of sight of the point. 3. The surface orientation is similarly considered in the inliers counting procedure (and in the shape sampling): a point is an inlier only if its normal is close to the local normal of the instantiated shape. 4. Finally, the inlier counting procedure exploits the graph induced by the k-NN relation to count the number of inliers in the largest connected regions from the seed points (that instantiated the surface) and inside the inliers band. Not only such a combination of RANSAC and region growing helps avoid the so-called "hallucination" problem of RANSAC variants (artificially finding structure by relating small clusters), but it is also more efficient (only a subset of the points are visited).

Shapes of the different classes (planes, spheres, cylinder and cones here) are tentatively extracted from the point set, and the best fitting shape is selected (provided there is one).

3.2.2 Hierarchical extraction

Instead of repeatedly applying the single shape extraction by sequentially removing fitted point from the point set (as depicted in Fig. 2(d)), the detection is guided towards interesting shapes to simultaneously build the BSP tree required

for the surface reconstruction step. From a previously built BSP tree, a restricted shape extraction is tried in each active leaf. If no shape can be extracted from a leaf, the leaf is marked as unactive and will not be explored again. After a successful extraction and before splitting the point cloud of the leaf, a few steps are followed. Large enough connected sets of inliers (as in 3.2.1) are found in the band around the detected surface. The surface is refitted to all these new inliers points which are now excluded from the point cloud. Outliers lying in the band are however kept for further shape sampling. Finally, a new node corresponding to the shape is created along with two leaves and points are reassigned to the leaves where they are located[2]. The whole process is iterated until no further shape extraction is possible.

3.3. Approximation of the induced cell complex

Practical exact computation of complexes whose cells are delimited by general second order surfaces is still the subject of active research [16, 11], not to mention the queries required by our surface reconstruction step. To circumvent this major problem, an approximation of the cell complex induced by the BSP tree is computed by using an adaptive multi-domain volume mesh generator [25] which extends the surface meshing algorithm of [6]. This algorithm works by refining a 3D Delaunay triangulation and only requires as input an oracle answering for a point which domain it is associated with (in our case, the leaf of the BSP tree where the point is located). The output is a labeled Delaunay triangulation, which approximates the interfaces between the distinct valued domains: each tetrahedron of this triangulation is labeled according to its domain. The cells of our approximated BSP complex are then found as the connected components of identically labeled tetrahedra. The facets of the complex (and the whole adjacency graph) are found as the connected components of triangle facets between two tetrahedra with the same labels pairs.

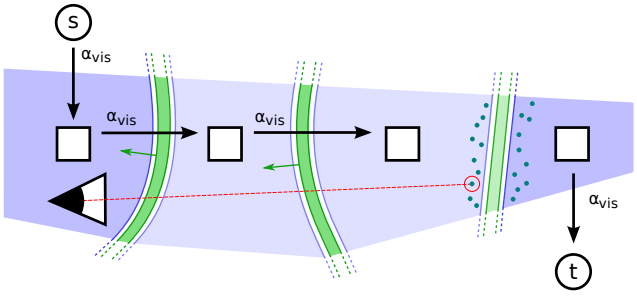


Figure 3: ***Visibility.*** A fitted point (red circled dot) affects the oriented facets (darker green) and cells (darker blue) weights along one of its lines of sight (red dotted line).

[2]A variant consists in identifying shape inliers in the whole point cloud to then split all the intersected leaves and not only the current one.

3.4. Surface reconstruction

Once a BSP tree of surfaces has been extracted from the point cloud and the original points clustered (one cluster per extracted shape abd one for points not fitted), the complex induced by this BSP tree can be computed. As previously announced in 2.2, the reconstructed surface $\mathcal{S}$ is sought as a particular subset of the BSP complex facets, which bounds the interior of the scene: the output surface is a labeled triangulated mesh with or without boundaries whose facets are labeled with their supporting shapes. It is also guaranteed to be watertight and intersection-free.

An energy is defined on the space of all such surfaces (or equivalently the space of all inside/outside labelings of cells). This energy only consists of a term $\mathrm{E}_{\mathrm{vis}'}(\mathcal{S})$ ensuring the visibility-consistency of the surface :

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S})$$

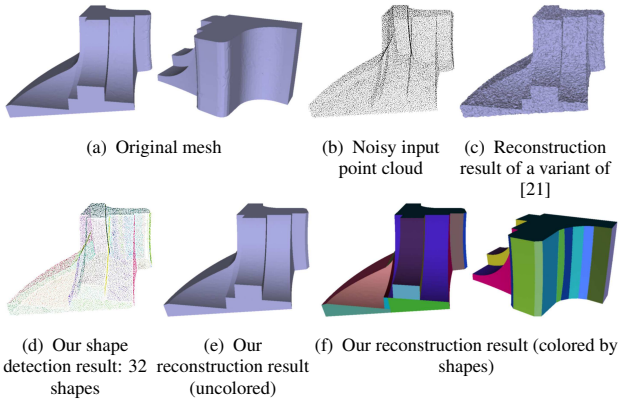
3.4.1 Network graph

The network graph considered for the cut is straightforwardly derived from the BSP complex. Its set of vertices stands for the cells of the complex, augmented with the source s and the sink t terminal vertices. The edges of the graph correspond to the oriented facets of the complex (its 2D faces) which are shared by adjacent cells: an edge from from a vertex v_i to a vertex v_j is the oriented facet from the cell c_i to the cell c_j of the complex. Moreover each non-terminal vertex/cell is linked to the sink and the source vertices. Out of the common trend in computer vision, our network graphs have a spatially varying connectivity. Moreover they may be multigraphs, as two cells could be adjacent through different facets (this only means that these facets are coupled in the optimization).

3.4.2 Surface visibility

The visibility term of our energy is a variant of the one designed in [21] which penalizes mis-alignments and mis-orientations of the surface w.r.t. the lines of sight of the fitted points. While we are also given a point cloud with lines of sight, a subtle difference with [21] exists: here, the points conveying visibility information do not coincide with the vertices of our complex, but are instead located inside cells, near some facet contained in one of the fitted shapes.

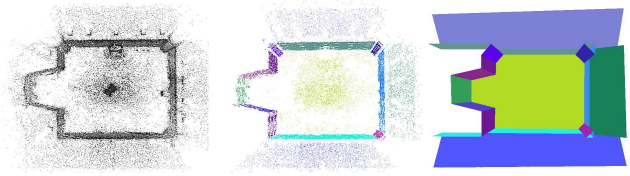
The corresponding visibility construction is adapted as shown in Fig. 3: the oriented facets crossed by a line of sight (darker green) get a weight of α_{vis} (the confidence in the fitted point, derived from its matching score), while the cell (darker blue) where the camera optical center lies is linked to the source s with an α_{vis} weight and the cell behind the shape of the fitted point (darker blue) is linked to the sink t with a weight α_{vis}. These weights for cells being inside



(a) Original mesh (b) Noisy input point cloud (c) Reconstruction result of a variant of [21]

(d) Our shape detection result: 32 shapes (e) Our reconstruction result (uncolored) (f) Our reconstruction result (colored by shapes)

Figure 4: ***fandisk data set and results.***



(a) Point cloud from depth maps (top view) (b) Our shape detection result: 14 shapes (top view) (c) Our shape-based reconstruction result (top view)

Figure 5: ***Results on the castle-P30 data set of [31].***

or outside and for facets being part of the reconstruction are accumulated for all the available lines of sight.

3.5. Hybrid surface reconstruction

The above reconstruction works well for scenes or for objects that can be easily decomposed in surfaces from the set of shapes and when no severe occlusion hinders some parts of the scenes from being sampled by the depth maps. Since the above reconstruction relies on a boundary representation, missing surface patches required to close the object volume can lead to gross error in the reconstruction (see Fig. 2(h) and 2(i)). To capture most of the geometry of the object, we propose to compute a hybrid reconstruction combining shape elements as before with some points of the depth maps to recover a faithful reconstruction of the whole scene. This is achieved as follows: instead of restricting the optimization domain to the approximated BSP complex, the whole triangulation is used, augmented with points that were not fitted to the detected shapes. This is done without altering the meshing of the shapes, by first computing the Delaunay triangulation of the points that were not fitted and then refining this triangulation (without modifying the positions of the vertices) using the multi-domain volume mesh generator (and recovering approximated BSP complex cells and facets as above). This way, the optimization domain is now embeds both the shapes and the points likely to reside in fine or uncaptured details of the scene. The output surface is still a triangulated mesh but whose facets are either unlabeled or labeled with their supporting shapes. The discrete energy used in the optimization is modified as follows:

$$\mathrm{E}(\mathcal{S}) = \mathrm{E}_{\mathrm{vis}'}(\mathcal{S}) + \mathrm{E}_{\mathrm{vis}}(\mathcal{S}) + \lambda_{\mathrm{hybrid}}\,\mathrm{E}_{\mathrm{hybrid}}(\mathcal{S})$$

where $\lambda_{\mathrm{hybrid}}$ is a positive weighting constant.

The term $\mathrm{E}_{\mathrm{vis}'}(\mathcal{S})$ penalizes visibility inconsistencies w.r.t. the fitted points and is the same term as previously but on the triangulated domain instead (all triangular facets crossed by a line-of-sight are penalized, and as before, the ray is extended to cross the corresponding fitted shape). The term $\mathrm{E}_{\mathrm{vis}}(\mathcal{S})$ refers to points not fitted to shapes and is exactly the same as the visibility term of [21]. Finally, the term $\mathrm{E}_{\mathrm{hybrid}}(\mathcal{S})$ is the surface quality term of [22] penalizing only triangular facets that do not belong to facets of the BSP complex (*i.e.* facets not belonging to recovered shapes). As shown in Fig. 2(j) and 2(k), adjusting the weight $\lambda_{\mathrm{hybrid}}$ allows switching from a purely shape-based reconstruction to a hybrid reconstruction with finer details (apart from Fig. 2(j), all the presented hybrid reconstruction results used the same value of $\lambda_{\mathrm{hybrid}}$).

4. Implementation details

Our prototype implementation of the described algorithm extensively uses the Computational Geometry Algorithms Library (CGAL) [2] for the various geometric computations it needs. CGAL offers excellent and robust implementations of all the needed constructions and queries for k-D trees used in 3.2 and Delaunay triangulations and Delaunay-based volume and surface mesh generation used in 3.3, 3.4 and 3.5.

Fitting of second order surfaces is done with standard least-squares of the Euclidean distance to the shape and is implemented with Levenberg-Marquardt optimization.

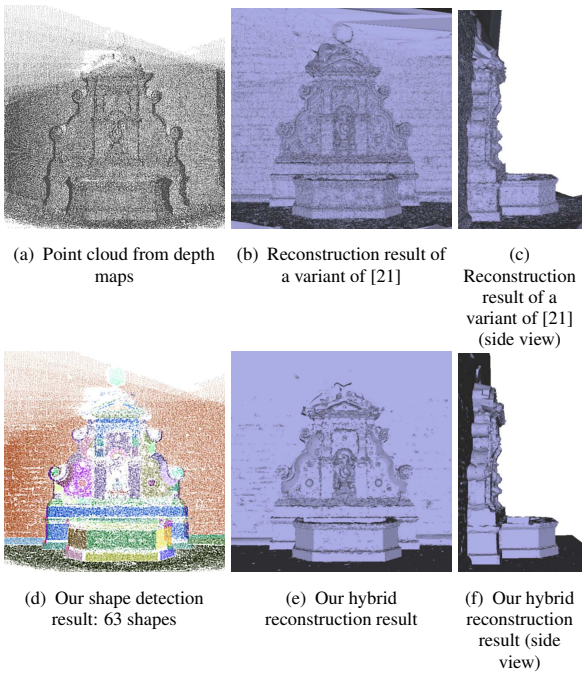
5. Experiments[3]

We compared our approach with a variant of the surface reconstruction step of [21] including only its visibility term complemented with the surface quality term of [22] (the input point cloud is the same as the one computed in 3.1). While [21] does not output high precision reconstructions, it is most similar to our final reconstruction step and is able to cope with the difficult open scenes of [31].

5.1. Synthetic data

Fig. 2 and 4 evaluate the last two steps of our shape-based segmentation and reconstruction pipeline (which constitutes the major contributions of this paper) on synthetic

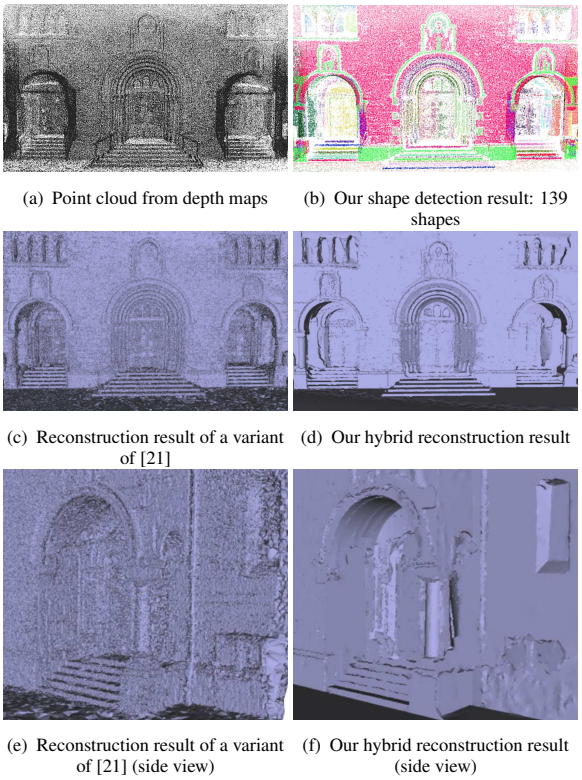
[3] Additional images and quantitative results are contained in the supplemental material.



(a) Point cloud from depth maps (b) Reconstruction result of a variant of [21] (c) Reconstruction result of a variant of [21] (side view)

(d) Our shape detection result: 63 shapes (e) Our hybrid reconstruction result (f) Our hybrid reconstruction result (side view)

Figure 6: ***Results on the fountain-P11 data set of [31].***



(a) Point cloud from depth maps (b) Our shape detection result: 139 shapes

(c) Reconstruction result of a variant of [21] (d) Our hybrid reconstruction result

(e) Reconstruction result of a variant of [21] (side view) (f) Our hybrid reconstruction result (side view)

Figure 7: ***Results on the Herz-Jesu-P25 data set of [31].***

data. In each case, the input point cloud was generated from the vertices of a mesh of the object. The associated visibility information was determined with ray casting and occlusion computation using 64 virtual cameras around the object. Some amount of isotropic Gaussian noise was added to the locations of the points (with a std. dev. representing 0.2% of the maximum dimension of the bounding box). In these synthetic experiments, our purely shape-based reconstruction works extremely well and automatically outputs a faithful representation of the original object. Note how the fandisk back, which is not exactly a cylinder is approximated with several cylindric patches.

5.2. Real data

We tested our algorithm on several scenes from the publicly available benchmark of [31]. These large-scale outdoor open scenes are quite difficult data sets, and few traditional multi-view stereo algorithms can cope with them. The *castle-P30* scene is an ideal candidate for our purely shape-based reconstruction as it features large facades, ground and roof planes. Our result shown in Fig. 5 is a concise description and simplified model of the scene. The *fountain-P11* and *Herz-Jesu-P25* scenes of Fig. 6 and 7 combine easily identifiable shapes with very fine details making them suitable to challenge our hybrid surface reconstruction. The result of [21] in Fig. **??** is noisy but contains most of the details of the fountain. Our reconstruction automatically extends the wall and ground, identifies the planar parts of the fountain base and approximates sculptures with small decorations with smooth second order patches. The results of Fig. 7 on the *Herz-Jesu-P25* scene are even more striking. While [21] outputs a very noisy model, note how the staircases of our reconstruction are perfectly straight (noisy staircases are typical cases of failure of RANSAC methods), the columns and archways are smooth and the facades and the ground (which produces lots of mismatches) have been replaced by planes.

6. Conclusion

We have presented a novel dense multi-view stereo method with strong shape priors that directly outputs a compact segmented model of the scene, contrasting with traditional approaches aiming at overly detailed models. Our approach encompasses clustering, multiple structures detection in noisy point clouds with outliers and shape-based surface reconstruction. We have shown encouraging results on both synthetic and challenging real-world data which clearly demonstrate the benefits of our approach.

Future work involves improving the shape class selection of 3.2.1 to include an MDL or MML-inspired criterion for instance. Extending our reconstruction pipeline to other, more complex, shapes would allow our simplified modeling technique to better capture more general scenes. Fi-

nally, a local refinement could be applied as a lightweight post-processing to our hybrid reconstructions to improve the transitions between fine details and shapes. Applying our shape-based surface reconstruction pipeline to more data sets and especially range scan data is also expected.

References

[1] Autodesk ImageModeler, 2009.

[2] CGAL, Computational Geometry Algorithms Library, 2009.

[3] Google SketchUp, 2009.

[4] C. Baillard and A. Zisserman. Automatic reconstruction of piecewise planar models from multiple views. In *IEEE Conference on Computer Vision and Pattern Recognition*, 1999.

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

[6] J.-D. Boissonnat and S. Oudot. Provably good sampling and meshing of surfaces. *Graphical Models*, 67:405–451, 2005.

[7] Y. Boykov and V. Kolmogorov. Computing geodesics and minimal surfaces via graph cuts. In *IEEE International Conference on Computer Vision*, 2003.

[8] H. Chen, P. Meer, and D. E. Tyler. Robust regression for data with multiple structures. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 1069–1075, 2001.

[9] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *ACM SIGGRAPH*, 1996.

[10] A. 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.

[11] L. Dupont, M. Hemmer, S. Petitjean, and E. Schömer. Complete, exact and efficient implementation for computing the adjacency graph of an arrangment of quadrics. In *15th Annual European Symposium on Algorithms*, 2007.

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

[13] L. R. Ford and D. R. Fulkerson. *Flows in Networks*. Princeton University Press, 1962.

[14] F. Fraundorfer, K. Schindler, and H. Bischof. Piecewise planar scene reconstruction from sparse correspondences. *Image and Vision Computing*, 24(4):395–406, 2006.

[15] H. Fuchs, Z. M. Kedem, and B. F. Naylor. On visible surface generation by a priori tree structures. *ACM Computer Graphics*, page 124–133, 1980.

[16] N. Geismann, M. Hemmer, and E. Schömer. Computing a 3-dimensional cell in an arrangement of quadrics: Exactly and actually! In *17th Annual Symposium on Computational Geometry*, 2001.

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

[18] A. Hoover, G. Jean-Baptiste, X. Jiang, P. J. Flynn, H. Bunke, D. B. Goldgof, K. Bowyer, D. W. Eggert, A. Fitzgibbon, and R. B. Fisher. An experimental comparison of range image segmentation algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(7):673–689, 1996.

[19] D. Kirsanov and S. J. Gortler. A discrete global minimization algorithm for continuous variational problems. Technical Report TR-14-04, Harvard Computer Science, July 2004.

[20] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(2):147–159, 2004.

[21] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, Delaunay triangulation and graph cuts. In *IEEE International Conference on Computer Vision*, 2007.

[22] P. Labatut, J.-P. Pons, and R. Keriven. Robust and efficient surface reconstruction from range data. *Computer Graphics Forum*, 2009. to appear.

[23] D. Luebke, M. Reddy, J. D. Cohen, A. Varshney, B. Watson, and R. Huebner. *Level of Detail for 3D Graphics*. Morgan Kaufmann, 2002.

[24] D. R. Myatt, P. H. S. Torr, S. J. Nasuto, J. M. Bishop, and R. Craddock. NAPSAC: High noise, high dimensional robust estimation - It's in the bag. In *British Machine Vision Conference*, 2002.

[25] J.-P. Pons, F. Ségonne, J.-D. Boissonnat, L. Rineau, M. Yvinec, and R. Keriven. High-quality consistent meshing of multi-label datasets. In *Information Processing in Medical Imaging*, 2007.

[26] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2006.

[27] A. Shamir. A survey on mesh segmentation techniques. *Computer Graphics Forum*, 27(6):1539–1556, 2008.

[28] S. N. Sinha, D. Steedly, R. Szeliski, M. Agrawala, and M. Pollefeys. Interactive 3D architectural modeling from unordered photo collections. In *ACM SIGGRAPH Asia*, 2008.

[29] N. Snavely, S. M. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80(2):189–210, 2008.

[30] C. V. Stewart. Bias in robust estimation caused by discontinuities and multiple structures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19:818–833, 1997.

[31] C. Strecha, C. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition*, 2008. `http://cvlab.epfl.ch/~strecha/multiview/denseMVS.html`.

[32] R. Toldo and A. Fusiello. Robust multiple structures estimation with J-Linkage. In *European Conference on Computer Vision*, 2008.

[33] P. H. S. Torr and A. Zisserman. MLESAC: A new robust estimator with application to estimating image geometry. *Computer Vision and Image Understanding*, 78:138–156, 2000.

[34] T. Werner and A. Zisserman. New techniques for automated architectural reconstruction from photographs. In *European Conference on Computer Vision*, 2002.