Connect-and-Slice: an hybrid approach for reconstructing 3D objects Supplementary Material

Hao Fang   Florent Lafarge
Université Côte d'Azur, Inria
Firstname.Lastname@inria.fr

This supplementary material presents a set of additionnal experiments on the impact of the connectivity analysis and the linear constraints, robustness to noise, performance and comparisons with approximation pipelines and a building reconstruction method.

1. On the connectivity analysis step

The connectivity analysis step (presented in Section 4 of the paper) aims to search for and reconstruct structurally-valid surface components, called structural facets. This step allows us to quickly process a part of the input primitives and solve obvious primitive assembling situations. Figure 2 illustrates the relation between strongly connected primitives, creases, border polygons and structural validity.

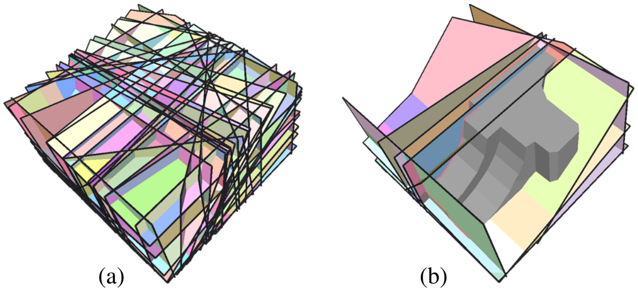

Figure 1. Primitive slicing with and without structural facets. Intersecting the 45 primitives of the defect-free version of the *Fandisk* model (see Figure 4) produces a complex partition composed of nearly $2K$ facets (a). By embedding the 36 structural facets (grey mesh), the complexity of the partition with the 9 remaining primitives drops to less than a hundred facets (b).

As illustrated in Figure 1, the presence of structural facets reduces the complexity of the partitioning data-structure. Note that more advanced connectivity relationships inspired from collision detection problems could be used to better match primitives, but this would be more time-consuming than a direct distance between 3D rectangles.

This connectivity analysis step is however less efficient in presence of defect-laden data with typically a lower number of extracted structural facets. Figure 4 shows how the number of structural facets is reduced when noise is introduced in a defect-free point cloud.

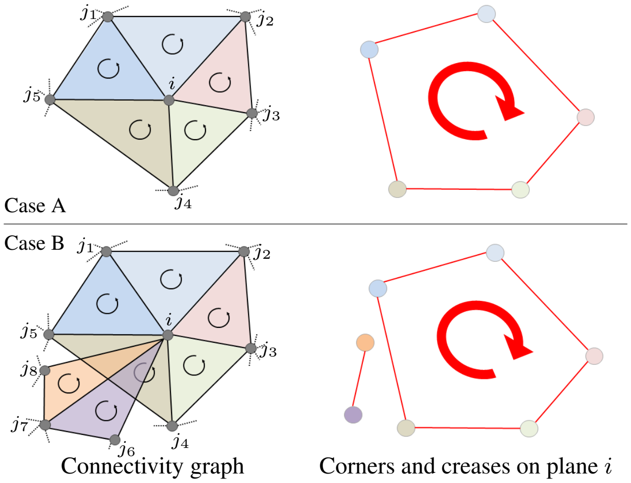

Figure 2. In Case A (top), primitive i is strongly connected to 5 primitives $j_1, .., j_5$. In the connectivity graph (top left), we detect 3-cycles (black curved arrows) that correspond to (corner) points at the intersection of 3 planes in the 3D space (see colored dots in the top right polygon). We then detect creases (red edges) by searching the pairs of corners which have exactly 2 primitives in common. A close sequence of creases (see red curved arrow) is a border polygon. The structural validity condition is respected here because all the creases generated from primitive i belong to the border polygon. In Case B (bottom), 3 more primitives j_6, j_7 and j_8 strongly connected to primitive i are added. The structural validity condition is not respected anymore because the left isolated crease does not belong to the border polygon.

2. Robustness to noise

As mentionned in the paper in Section 7, our algorithm is relatively robust to noise as long as primitives can be decently detected. Figure 3 shows how the output surface and its geometric error to data evolve when we progressively increase noise in a defect-free point cloud.

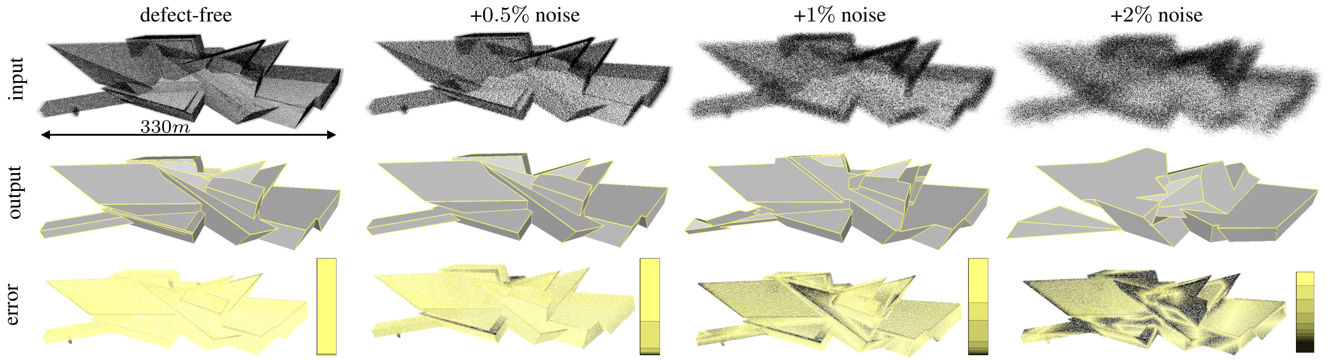

Figure 3. Robustness to noise on dataset *Museum*. Our output surface meshes (middle) are weakly affected by noise as long as primitive detection can capture the main planar components of the object. When adding 2% of noise (expressed w.r.t. the 3D bounding box diagonal), primitives are no longer correctly detected. Yellow-to-black colored points (bottom) represent the Hausdorff distance from the defect-free point cloud to output surface (yellow $= 0m$, black $\geq 8m$).

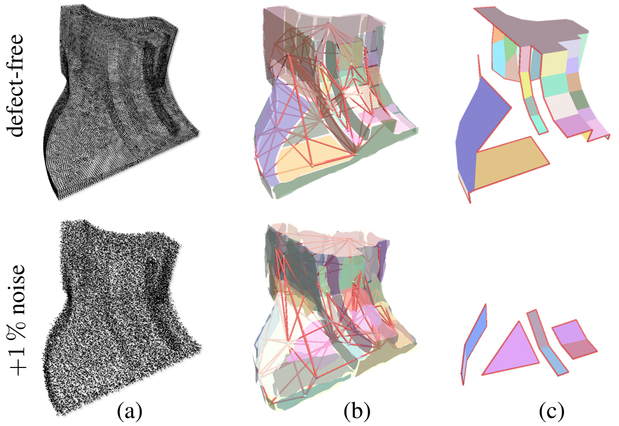

Figure 4. Connectivity analysis on *Fandisk*. Primitives and associated connectivity graph (b) are typically accurate when input data (a) is clean (top). Our quick connectivity analysis allows us to process 36 of the 45 initial primitives on the top example, leading to the reconstruction of 36 structural facets (c). When data is defect-laden, for instance highly noisy (bottom), the connectivity analysis is less efficient: connectivity graph contains many ambiguities that restricts the number of structural facets. Nevertheless the 7 structural facets recovered on the bottom example are all relevant. Anchor edges are colored in red in (c).

3. Impact of the linear constraints

The linear constraints introduced in the energy formulation (Section 6 of the paper) allow to impose some geometric guarantees on the output surface. These guarantees, which are intersection-free, 2d-manifold and watertight, can be optionnally desactivated. Figure 5 shows the impact of these constraints on the output solution. The activation of the 2d-manifold and watertight constraint is required in most cases, unless the end-user is satisfied with a rough polygon soup. The activation of the intersection-free constraint is required only when input data contained defects as noise and outliers.

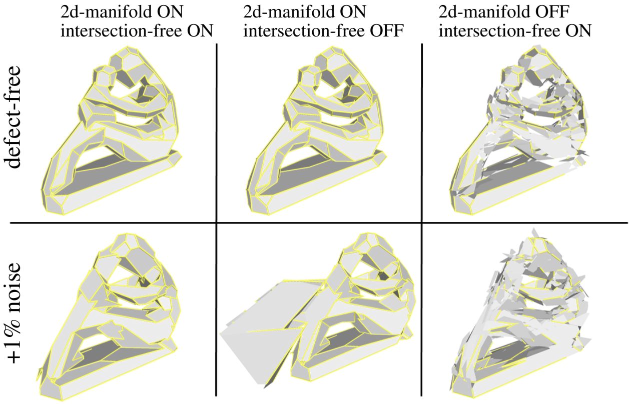

Figure 5. Impact of constraints. Without activating the 2d-manifold and watertight constraint, the output surface exhibits poorly connected facets as well as holes and edges adjacent to more than two facets (right). Deactivating the intersection-free constraint has no impact on the quality of the output mesh when input data is defect-free (top-middle) but tends to make the output surface too complex with groups of facets that self-intersect (bottom-middle).

4. Performance

Table 1 gives performances in terms of running time and memory consumption for reconstructed models shown in the paper. This table also provides information on the point cloud size, the number of detected primitives, the number of detected sructural facets, the number of candidate facets in the partition, and the number of facets in the output surface for these models.

	Mechanical	*Fandisk*	*Church*	*Euler*	*Indoor*	*Couch*	*Face*	*House*	*Museum*	*Hand-20*	*Hand-100*	*Hand-300*	*Hand-1200*
input size	382K	27K	18K	4M	186K	460K	144K	16K	129K	369K	369K	369K	369K
#primitives	60	45	80	45	50	50	70	51	75	20	100	300	1200
#structural facets	35	36	5	1	10	3	3	5	10	0	8	23	84
#facets in $\mathcal{F}$	2.2K	213	1.9K	11.2K	3.1K	2.6K	6.6K	3.1K	5.5K	429	7.3K	39K	71K
output complexity	298	89	291	1.1K	401	381	876	376	637	79	875	3747	7759
connectivity analysis (sec)	1.8	1.2	0.9	9	1	1.2	1.4	0.3	1.6	1.5	2.7	11.1	21.9
space partitioning (sec)	2.5	1	1.8	3	2	1.7	3	2.2	0.9	0.6	4.2	21	81.8
surface extraction (sec)	2.4	2	15	684	108	18	23	62	5	10	45	286	529
memory peak (Mb)	63	36	186	738	201	194	269	201	167	45	141	546	741

Table 1. Performance on some reconstructed models in terms of running time and memory consumption. The output complexity is expressed in number of active facets returned by the surface extraction solver.

5. Comparison with approximation pipelines

An alternative strategy to produce concise polyhedral meshes consists in simplifying dense triangular meshes. The later are preliminary extracted from input points by smooth surface reconstruction algorithms such as Poisson [4]. Once extracted, edges of the dense mesh can be progressively contracted until reaching a target number of facets [3, 6]. Eventually the edge contraction can be driven by planar primitives preliminary detected in the dense mesh [8]. Such contraction mechanisms however return triangular meshes whose adjacent facets are unlikely to be coplanar, and thus to form a meaningful planar part of the object. The methods proposed in [2, 1] adopt another approach by assembling planar primitives with a connectivity graph robustly built from the dense mesh. Shape abstraction [10, 7] is also a way to simplify regular objects at different levels of details. To be efficient, these various surface approximation algorithms requires geometrically and topologically accurate dense meshes as input. Unfortunately, this requirement cannot be guaranteed from real-world data corrupted by noise, outliers and occlusions.

We compared our algorithm with the surface approximation methods QEM [3], SAMD [8] and VSA [2]. Because these methods operate from meshes, we first reconstructed a dense triangle mesh from input points using the screened Poisson algorithm [5]. As shown in Figure 6, polyhedral meshes produced by our algorithm are geometrically more accurate than those returned by these methods at a similar mesh complexity, i.e. with the same number of output facets. The accuracy gain is particularly high at low mesh complexity where approximation methods cannot capture correctly the structure of the object anymore and tend to shrink the output surface. By operating directly from input points, our algorithm does not depend on an intermediate dense mesh reconstruction step in which geometric and topological errors frequently occur. Moreover, the large polygonal facets returned by our algorithm approximate the object more efficiently than the triangle facets returned by edge contraction (QEM and SAMD) or constrained Delaunay triangulation built from a primitive connectivity graph (VSA).

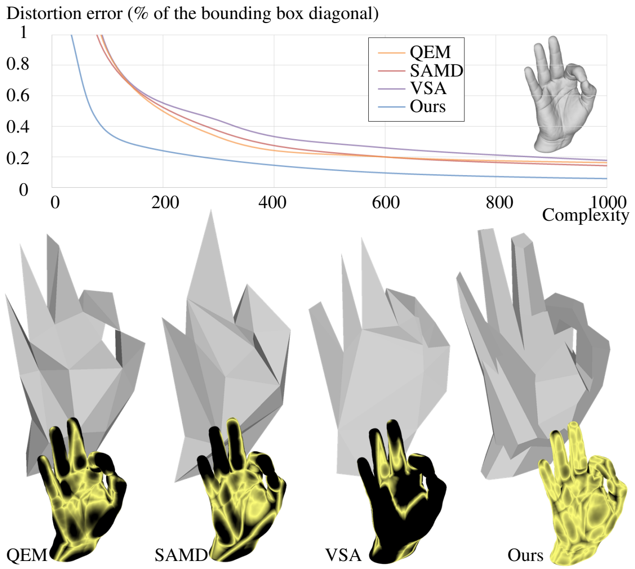

Figure 6. Comparisons with surface approximation methods on *Hand*. Our algorithm presents a better complexity-distortion tradeoff than QEM [3], SAMD [8] and VSA [2] (see graph). Complexity is given by the number of output facets whereas distortion error is measured as the RMS Hausdorff distance from input points to output surface. Our algorithm performs best at low complexity. For instance, our 60-facet output mesh still captures well the different fingers. At the same complexity, QEM, SAMD and VSA cannot approximate the hand accurately, as shown by grey meshes and colored input points ranging from yellow (Hausdorff distance =0) to black (Hausdorff distance $\geq 1\%$ of the bounding box diagonal).

6. Comparison with a specialized building reconstruction method

While generic, our algorithm produces competitive results for the building reconstructing task in comparison to the specialized method of Verdie et al. [9]. Figure 7 shows comparative results obtained on two models: one simple, *Cottage*, and one more complex, *Church*. Our reconstructions on *Cottage* and *Church* have a RMS error of 0.13 and 0.29 meter respectively, and are performed in 8 and 18 seconds. In contrast, the LOD2 reconstructions of [9], which suffer from approximated discrete space partitions, have higher errors (0.14m and 0.4m) and processing times (52s and 198s).

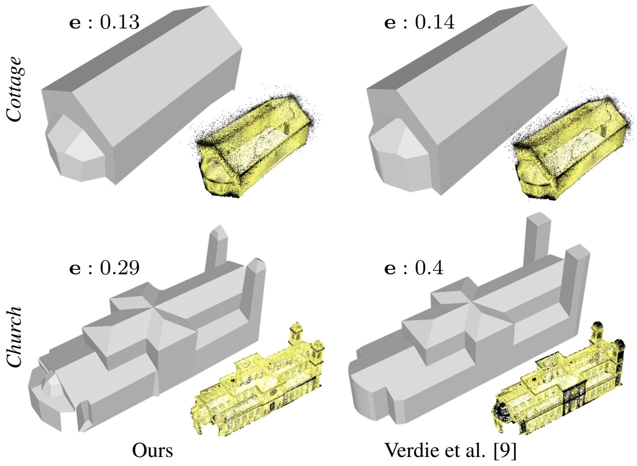

Figure 7. Comparisons with a specialized building reconstruction method. While visually similar, our output meshes have a lower geometric error than the meshes produced by specialized building reconstruction method [9], especially on the most complex model (bottom). Yellow-to-black colored points represent the Hausdorff distance from the input points to the output surface (yellow $= 0m$, black $\geq 0.5m$ for *Cottage*, and yellow $= 0m$, black $\geq 1m$ for *Church*). $\mathbf{e}$ refers to the RMS Hausdorff distance over all the input points.

References

[1] D. Chen, P. Sitthi-amorn, J. Lan, and W. Matusik. Computing and fabricating multiplanar models. In *Computer graphics forum*, volume 32, 2013.

[2] D. Cohen-Steiner, P. Alliez, and M. Desbrun. Variational shape approximation. In *Siggraph*, 2004.

[3] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, 1997.

[4] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, 2006.

[5] M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *Trans. on Graphics*, 32(3), 2013.

[6] P. Lindstrom. Out-of-core simplification of large polygonal models. In *SIGGRAPH*, 2000.

[7] R. Mehra, Q. Zhou, J. Long, A. Sheffer, A. Gooch, and N. Mitra. Abstraction of man-made shapes. *Trans. on Graphics*, 28(5), 2009.

[8] D. Salinas, F. Lafarge, and P. Alliez. Structure-Aware Mesh Decimation. *Computer Graphics Forum*, 34(6), 2015.

[9] Y. Verdie, F. Lafarge, and P. Alliez. LOD Generation for Urban Scenes. *Trans. on Graphics*, 34(3), 2015.

[10] M. Yumer, K. Ersin, and D. Levent. Co-abstraction of shape collections. *Trans. on Graphics*, 31(6), 2012.