Efficient Architectural Structural Element Decomposition

Nikolay Kobyshev[a,*], Hayko Riemenschneider[a], András Bódis-Szomorú[a], Luc Van Gool[a,b]

[a] *Computer Vision Laboratory, ETH Zurich, Sternwartstrasse 7, CH-8092 Zurich, Switzerland*
[b] *Vision for Industry Communications and Services (VISICS), KU Leuven, Kasteelpark Arenberg 10, B-3001 Heverlee, Belgium*

Abstract

Decomposing 3D building models into architectural elements is an essential step in understanding their 3D structure. Although we focus on landmark buildings, our approach generalizes to arbitrary 3D objects. We formulate the decomposition as a multi-label optimization that identifies individual elements of a landmark. This allows our system to cope with noisy, incomplete, outlier-contaminated 3D point clouds. We detect four types of structural cues, namely dominant mirror symmetries, rotational symmetries, shape primitives, and polylines capturing free-form shapes of the landmark not explained by symmetry. Our novel method combine these cues enables modeling the variability present in complex 3D models, and robustly decomposing them into architectural structural elements. Our proposed architectural decomposition facilitates significant 3D model compression and shape-specific modeling.

Keywords: 3D city model, architecture, structure, element, landmark, decomposition, optimization

1. Introduction

Modeling our environment is a common strive in photogrammetry, computer vision and graphics. 3D modeling from imagery has been going through a great evolution over the past decades, maturing methods like incremental Structure-from-Motion (SfM) [1, 2, 3], Internet-scale point cloud reconstruction from imagery [4], high-accuracy detailed surface reconstructions via dense Multi-View Stereo (MVS) [5, 6], and ach ieved success in procedural modeling of facades [7, 8]. LiDAR is an alternative dominant technology to obtain point clouds of urban scenes [9].

In this work, we tackle the abstraction and understanding of 3D point clouds delivered by such state-of-the-art technologies. Planar priors [10, 11, 12, 13], or a Manhattan-world assumption [14] proved to be enough for many man-made structures. However, for a large mass of buildings, especially landmark architecture or general objects, a simple shape primitive abstraction will not suffice. Instead, we propose to decompose a 3D reconstruction by exploiting symmetries within the model. Such a decomposition is a first step towards understanding and compactly modeling the architectural elements of a landmark.

Our method is based on weak architectural priors that naturally hold for a majority of buildings, namely mirror symmetries, rotational symmetries and wall verticality. The method starts with a semi-dense 3D point cloud that may be contaminated by noise and gross outliers, and may be highly inhomogeneous. Structure-from-Motion (SfM) point clouds often suffer from such contamination. We show how to robustly detect *structural cues*, more precisely, axis directions of dominant mirror symmetries, the pivot of rotational symmetries, shape primitives, and free-form parts that are not explained by the symmetries. These cues provide a strong guidance for extracting dominant and semantically meaningful components of a model, such as a wall, a tower, an arch, etc., as illustrated in Figure 1. We refer to these components as *architectural structural elements* (ASE) throughout this work. We formulate the decomposition problem as an energy-driven, multi-label point cloud segmentation. Our contributions:

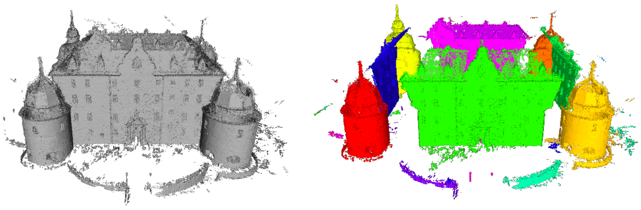


Figure 1: Our method segments a point cloud of a landmark building into coherent architectural structural elements, such as walls, towers based on structural cues. The noisy points that do not belong to any component get a random assignment.

- a model that combines symmetries and free-form polylines for decomposing a point cloud into ASEs,
- methods for detecting structural cues (dominant mirror symmetries and bodies of rotation, as well as residual free-form parts) in point clouds,
- a global energy formulation and optimization approach for partitioning a point cloud into meaningful structural components based on structural cues.

Our proposed abstraction paves the road to 3D model compression [15] or to shape-specific models [16, 17].

[*] Corresponding author

2. Related Work

Reconstructing surfaces from point clouds is a dominant problem in computer vision as well computer graphics. The range of research varies from volumetric segmentation to the detection of symmetries and repetitions, enforcing shape priors and shape primitives. Depending on the architecture or manufacturing [18] there may be other constraints. For example, for simple Manhattan-style skyscrapers the modeling can be as simple as a rectangular box [14]. For Haussmannian architecture, strong priors or regular floors may be sufficient to model the buildings [19]. For more general architecture, more relaxed structural principles such as symmetries have to be used [20, 8, 21]. Further even, in the case of real cities with regular planar buildings and complex shapes like statues, a hybrid model can be applied [22].

For landmark architecture there are few rules that hold across multiple landmarks, hence a more per-exemplar approach is needed. This is the direction we propose in this work, where we tackle the decomposition and understanding of architectural structures for landmarks.

2.1. Primitive Detection

In the line of shape priors for arbitrary surface reconstruction, there are two general cases. Either the raw data is replaced by a fitted shape primitive (hard prior), or in the other case an attraction force to the fitted shape primitive is used (soft prior). Both hard and soft priors are used in various forms (e.g. primitive fitting, shape grammars, etc.) to produce robust and clean results.

Methods for hard priors use robust fitting of models like planes, cylinders [23]. Schnabel et al. [24] detect simple primitives such as planes, spheres, cylinders, etc., and further extend shape primitives across the remaining surfaces for completion [25].

For the soft priors the prior is only included within the optimization which smooths out the final surface by guiding the shape as suggested by the prior. Haene et al. [26] have shown this for piecewise planar priors and the works of Dame and Bao [16, 17] showed this for arbitrary shape priors learned from 3D training data.

Monszpart et al. [27] have recently shown that indoors and outdoors man-made scenes can be efficiently represented as regular arrangements of planes. Moreover, just piecewise-planar representation can produce very high quality models of indoor environments, as shown in [28]. Plane fitting can be useful to enhance the representation of urban data, as shown in [29]. Oesau et al. [30] showed that it is possible to interleave plane detection and regularization of the scene with the detections. Lafarge et al. [31] detected multiple types of shape primitives in a recognition-style method, as they first cluster the input data into planar, concave, convex and non-developable surface types. This in turn defines the type of primitive to detect and removes much of the complexity of detecting all feasible shape primitives.

Verdie and Lafarge [32] proposed an efficient Monte Carlo sampler for detecting parametric objects in large scenes exploiting parallel processing and reversible jump between different primitive types.

In another work by Lafarge et al. [33] it is demonstrated that integration of primitive models can be useful for multi-view reconstruction.

Lafarge et al. [22] also propose a hybrid solution between shape primitives and arbitrary mesh topology. The authors initially show how to estimate the fitting of multiple shape primitives efficiently. The hybrid solution then allows for compact models while still preserving the details for arbitrary structures.

Primitive detection can be also done in 2D, on image sequences. For example, Pham et al. [34] fit orthogonal models to multi-view images.

Overall, these methods provide a better understanding through shape priors and primitives. Yet all methods have implicitly two drawbacks. First, they still require models for each specific types of primitive. Second, the related works cannot handle non-standard shape primitive which are large, complex shapes such as architectural elements (e.g. an entire tower) in landmark buildings.

To the best of our knowledge, [15] is the only related work for abstraction of buildings as it tries to decompose buildings into 2D sweeping profiles. However, in our experience, it iteratively finds the profiles and has problems decomposing into these architectural elements. Hence, [15] would benefit from a more holistic decomposition.

2.2. Symmetry Detection

Symmetries have been explored in computer vision for a long time and review reports are available [35, 36]. However, among the first to apply it for 3D buildings were [37, 38, 39].

Mitra et al. [37] introduce a voting for symmetries for reflection, rotation and translation. In their follow-up work, [38], they use the voting space to create a symmetrization effect to enhance symmetries while maintaining the shape of the model. Pauly et al. [39] discover structural regularity by detecting repeated structures in 3D objects, which have been generated by the means of computer graphics. They simultaneously evaluate the repetition pattern and detect the repeating geometric elements.

Cohen et al. [40] take it one step further and let the symmetries influence the Structure-from-Motion optimization, whereas Koeser at al. [41] exploit mirror symmetries in dense reconstructions from a single view.

Zheng et al. [42] rearrange parts of objects within large yet clean shape collections. In their recent work, Liu et al. [43] define replaceable substructures and use the shape graph for remodeling.

Contrary to these approaches, we start from point clouds and tackle the decomposition and understanding of real-world 3D landmark reconstructions. In summary, (1) we are the first to show a working method on noisy data, as other research has only worked with symmetries on clean 3D models, (2) others take pre-separated components for structural analysis and we aim to automate this separation, (3) our strengths over ill-suited point-wise analysis or recognition is that we can extract high-level groupings based on symmetries and coherent global similarities rather than making local decisions, and (4) we propose a new notion that extends from fixed primitives sets to arbitrary shapes, hence we provide a more general and scalable approach.

In particular, our point clouds are built from image-based reconstructions using Structure-from-Motion (SfM) tools [2, 3] and contains many noisy points and inaccuracies over clean CAD-style models. Further, we propose a method for segmenting complex landmark structures into parts using a range of structural cues such as mirror symmetries, bodies of rotation, plane primitives, and free-form polylines. More cues can easily be included in our unified optimization scheme.

3. ASE Decomposition

The overall goal of our method is to decompose a 3D point cloud representing a landmark into semantically meaningful architectural structural elements (ASE) of the building. Figure 2 gives an overview of our approach.

The input to our algorithm is a 3D point cloud of a building. In a preprocessing step, we align our input point cloud with the gravity vector (we extract it from a raw point cloud using the method suggested in [15]) and scale it to real-world (metric) scale, which can be easily automatized knowing the (coarse) GPS positions of the cameras. Next, we extract normals by Principal Component Analysis (PCA) over the 3D k-NN neighborhood of each point. We denote an oriented 3D point by $p_i = (\mathbf{X}_i, \mathbf{N}_i)$. Exploiting the natural vertical prior for walls, for some of the cues we project the points and their normals to the ground. We denote the coordinates and normals of the projected points as $p_i = (\mathbf{x}_i, \mathbf{n}_i)$. The corresponding point height is denoted as h_i.

We only use the 2D projections for detecting the mirror and rotational symmetries and extracting free-form shapes. By doing that, we assume that the rotational and mirror symmetries have vertical lines and planes of rotation. This is true for the vast majority of buildings (with an exception of, for example, falling towers).

Next, we perform detection of structural cues. First, we analyze the point cloud to detect its dominant mirror symmetries (to find opposing walls) and rotational symmetries (to detect bodies of revolution, e.g. towers) (Figure 2b). Second, we extract a rough floor plan of the point cloud to capture free-form structures that are not well explained by the symmetries (Figure 2c). Additionally, we extract planes and use them as low-level structural cues as well. Finally, we formulate a global energy minimization to robustly assign every 3D point to the structural elements that are generated either by a symmetry or a free-form shape (Figure 2d).

It is important to note that the symmetry detection in Section 3.1 and the free-form polyline extraction in Section 3.2 only provide symmetries and polylines that enter as *hypotheses* into a final optimization, discussion of which is detailed in Section 3.4. This optimization can suppress unlikely structural cues.

3.1. Symmetry Analysis

Symmetries are prominent properties of many landmarks. It is common for buildings to have self-reflection (e.g. opposite walls are often symmetric) or rotational symmetry (e.g. for towers or domes).

In this work, we demonstrate the detection of mirror and rotational symmetries. We extract them in the aforementioned 2D ground projection. Our symmetry extraction scheme is general and can be extended to other types of symmetries and to 3D symmetries. The main goal is the optimization over various structural cues, however additionally we demonstrate our methods in two variants for collecting symmetry evidence, namely Hough-space voting [44, 45] and RANSAC [23]. Inspired by [37], we generate votes for each pair of points for a symmetry in Hough-space, and show the extension of the approach with RANSAC that is less demanding to computational resources and memory. The RANSAC approach is greatly inspired by the Hough-space voting, and we present them both for clearer comparison and analysis of each approach.

3.1.1. Point Matching

For Hough voting, in order to prevent filling in the voting space with votes for unlikely 2D symmetries, only a selected subset of all possible point pairs is allowed to vote for symmetries. For simplicity, our criterion for a point pair $\{p_i, p_j\}$ (a *matching pair* from now on) to generate a vote is

$$|h_i - h_j| < t_h, \tag{1}$$

where h_i is the height of p_i over the ground as introduced earlier, and t_h is a height difference threshold defined as

$$t_h = 0.1 \cdot (\max_i h_i - \min_i h_i). \tag{2}$$

We note that this simple criterion could be replaced by a more sophisticated matching of local 3D shape descriptors, e.g. spin images [46], FPFH [47] or 3D SURF[48].

3.1.2. Detecting Mirror Symmetries

Hough Voting Approach. For mirror symmetries, every matching pair (p_i, p_j) votes for a hypothesized symmetry line, which is the perpendicular bisector of the 2D segment connecting p_i and p_j, as shown in Figure 3a. This symmetry line is parametrized by a pair (D_{ij}, ϕ_{ij}) (i.e. a point) in the Hough-space $\mathcal{H}^{\mathrm{mir}}(D, \phi)$, where D_{ij} is the distance of the line from the origin and ϕ_{ij} is the characteristic angle of the line shown in the figure. Figure 4a visualizes these

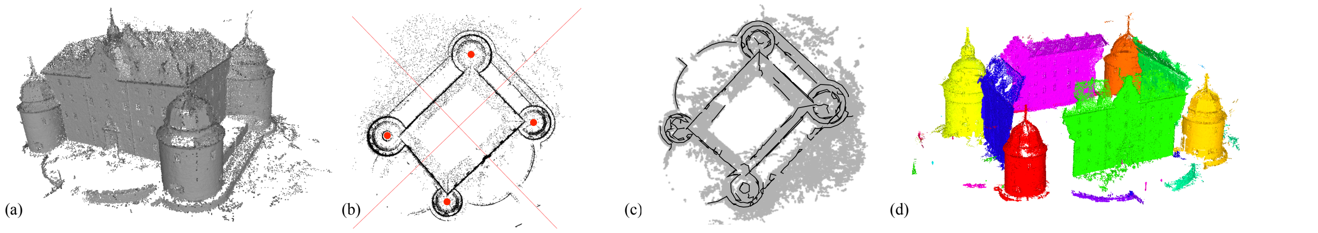


Figure 2: An overview of our method: (a) semi-dense point cloud as input, (b) identification of mirror symmetries and centers of bodies of revolution (axes in red), (c) extraction of free-form polylines, (d) our segmentation result.

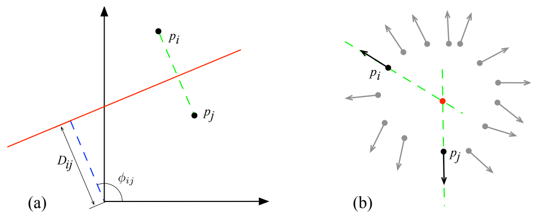


Figure 3: Our voting schemes for mirror (a) and rotational symmetries (b). (a) the points p_i and p_j vote for the direction of the red perpendicular bisector of the segment connecting the points, parametrized by its distance D_{ij} to the origin and its depicted angle ϕ_{ij}. (b) The points p_i and p_j vote for a pivot point that lies at the intersection of the green lines passing through the points parallel to the normals.

Hough-space votes for the example of Figure 2a. In this parametrization, we move the origin to be below and to the left from the lowest left-most point of the cloud. This is done to reduce the voting space: no symmetry line will have the characteristic angle ϕ_{ij} in $(\pi; \frac{3}{2}\pi)$.

Next, we extract dominant peaks in Hough-space, which correspond to likely axes of mirror symmetry. Here, we restrict ourselves to the two dominant perpendicular symmetries, which allows for a robust and parameter-free peak detection. More precisely, we seek the global maximum of

$$\mathcal{H}^{\mathrm{mir}}(D_1, \phi) + \mathcal{H}^{\mathrm{mir}}(D_2, \{\phi + \pi/2\} \bmod 2\pi) \qquad (3)$$

as a function of (D_1, D_2, ϕ) to obtain the two peaks, i.e., two symmetry axes (D_1^*, ϕ^*) and $(D_2^*, \{\phi^* + \pi/2\} \mathrm{mod}\ 2\pi)$.

This is solved exhaustively, where for each discrete value of ϕ, the maximal $\mathcal{H}^{\mathrm{mir}}(D_1)$ and $\mathcal{H}^{\mathrm{mir}}(D_2)$ are found (1D searches) and summed. However, this comes at virtually no cost as a coarse discretization (in the order of 360 × 200) of our 2D Hough-space proved to work well with our datasets.

RANSAC Approach. To form a mirror symmetry hypothesis, only two points are needed. If we have randomly selected a pair of points (p_i, p_j), they form a hypothesis for a symmetry line that is a perpendicular bisector of the 2D line between $\mathbf{x}_i$ and $\mathbf{x}_j$. Then, a part of the point cloud lying on one side of the symmetry line (the side can be chosen arbitrarily, we use the part with less points to save computational resources) is flipped over the symmetry line. We perform mirror reflection of the point cloud,

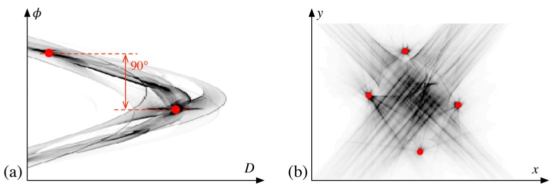


Figure 4: Hough-spaces for voting for (a) symmetry lines of mirror symmetries, (b) pivot points of rotational symmetries for the point cloud shown in Figure 2. Red dots show the extracted peaks (see the text for details).

i.e., we add the vector from the point to its closest point on the symmetry line to its coordinates twice. As a result of this operation, we have two groups of points lying on the same side of the symmetry line. We count a point as an inlier if it has a point from another group within the proximity threshold $t_m = 0.05$ for a metric-scaled point cloud. An example of the described procedure is shown in Figure 5. We repeat the random hypothesis generation iterations until RANSAC confidence is higher than 99%. We define confidence ξ as

$$\xi = 1 - (1 - w^2)^i, \qquad (4)$$

where i is the number of iterations and w is the ratio of number of inliers to the overall number of points. In the equation, w^2 is a best guess of the probability that both randomly selected points are inliers, and the value of ξ therefore indicates the probability that at least one hypothesis out of i ones is made with two inliers.

To define a good value for the parameter t_m, we run a solution stability test shown in Figure 6. In the picture one can see results of the RANSAC procedure run multiple times on the same dataset. It can be seen that the more small the threshold of point proximity t_m goes, the more similar are the solutions over multiple iterations.

As before, we aim to find the dominant perpendicular symmetries of the dataset. Therefore, we count the points as inliers only if they have a normal that is perpendicular to the symmetry line (the absolute value of the dot product between the normal and the symmetry line direction is lower than a threshold $t_d = 0.1$) and run the procedure twice, restricting the result of the second run from being too close to the first detected symmetry. As a result, we end up with the two symmetry lines that correspond to the

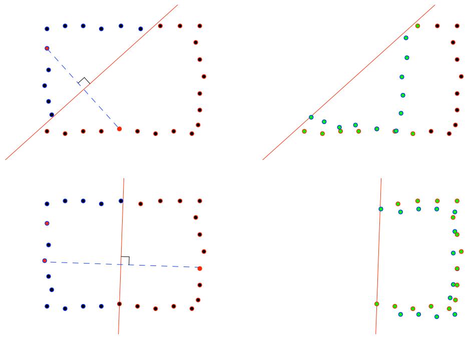


Figure 5: Mirror symmetry detection with RANSAC. Left: two randomly selected points (red inside) form a hypothesis (red line). Points from one side (with a blue outline) are mirrored, the result is shown on the right. Inliers (i.e. the points that have a point from another symmetry side nearby) are colored green. In the first guess (top row) the guess is not lucky and there are not many inliers. In the second guess (bottom row) every point is an inlier.

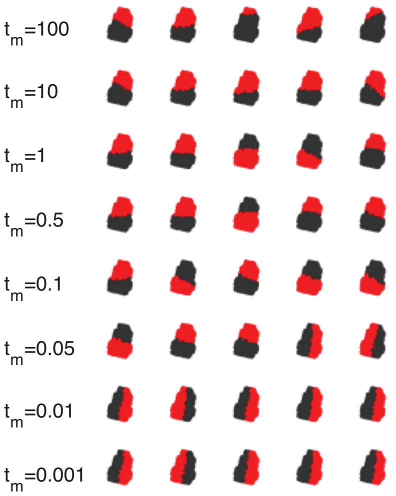


Figure 6: The RANSAC solution stability test of parameter t_m [inlier distance in meters] on the Fraumuenster dataset. The columns show different solutions of the same mirror symmetry problem by RANSAC. For different rows, different values of t_m are chosen. It can be seen that the smaller t_m goes, the more stable the solution becomes.

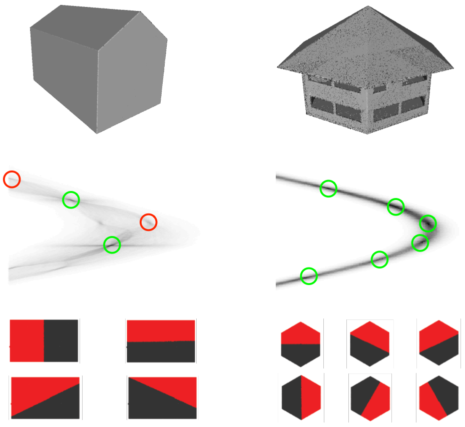


Figure 7: Mirror symmetry detection performance on different cases. Top: initial datasets (a house with orthogonal walls and a hexagonal building), middle: Hough-space voting, bottom: RANSAC detections (from a top view). See text for the detailed discussion.

dominant perpendicular symmetries of the point cloud. It should be noted that orthogonality of the detected symmetries is not enforced (as in the case of Hough transform). However, the search terminates after two symmetries are found.

Although, as will be shown further, our optimization is independent on the orthogonality of the detected mirror symmetries, we restrict ourselves to the two dominant directions. There are two reasons for choice of such a model (in the discussion we are referring to Figure 7):

Detection stability. Detecting maxima in Hough-space is a challenging task. Usually, non-maximum suppression (NMS) is used for that. In Figure 7, the house model in the left column has two dominant symmetries (that are highlighted with green circles on the Hough-space plot). However, there are two diagonal flip symmetries that also contribute a high score to the Hough-space (highlighted with red). If our detections are based only on NMS, the method becomes extremely sensitive to the NMS parameters that have to be tuned for every model. For Hough-spaces, another advantage the restrictive search brings is the solution to the space discontinuity problem. In case of general Hough voting symmetry lines that pass close to origin can have very different angles. This might be a problem for classical Hough space maxima detection techniques, such as NMS (as points that are far away in the Hough space might be very close in the original space). However, when using the restrictive search the problem disappears as the search is performed row-wise.

Model simplicity. In our experiment, RANSAC was always able to return two dominant mirror symmetries after two iterations. However, if the number of mirror symmetries is unknown, we need to determine the number of iterations after which to stop. In case of the house (left side of the picture), four iterations is too much (as it only has two mirror symmetries), whereas for a hexagonal building six iterations is needed to detect all symmetries. The task of analyzing multiple detections and determining the number of correct ones we leave for future work.

3.1.3. Detecting Rotational Symmetries

Hough Voting Approach. In the case of rotational symmetries, each matching point pair (p_i, p_j) votes for a hypothesized rotational pivot point (x_{ij}, y_{ij}) in a 2D Hough-space $\mathcal{H}^{\text{rot}}(x, y)$ of pivot points. The hypothesized pivot resides at the intersection of the two lines, each passing through the corresponding point parallel to its normal direction, as shown in Figure 3b.

Since for rotational symmetries we do not have such a natural simplifying constraint for peak detection as in the case of mirror symmetries, the standard scheme is employed for peak extraction. We use non-maximum suppression with window size w and perform repeated peak extraction until a confidence threshold c is reached as $\mathcal{H}^{rot}(x, y) > c$. We note however, that parameters w and c are kept fixed throughout the datasets in the experimental section, and recall that the detected symmetries enter as hypotheses into the final optimization in Section 3.4. An example of the corresponding Hough-space is shown in Figure 4b.

RANSAC Approach. We define body of revolution (BOR) as a set of 2D circles parallel to the ground plane in 3D space that share the same rotational pivot point and are located on different heights. A natural extension to this is a requirement for the radii of the circles to change smoothly with the height change, but we let the final global optimization take care of this.

To detect bodies of revolution, we make hypotheses about their rotational pivot points on the 2D ground plane by simply extracting 2D circles. Then (in the final optimization), we use context-aware point-wise potentials to decide whether a point is likely to belong to BOR with a center given by a particular 2D circle.

Given a pair of points (p_i, p_j), we generate a hypothesis for a circle with a center $\mathbf{c}$ at the point of intersection of the lines that go through $(\mathbf{x}_i, \mathbf{x}_j)$ and are parallel to $(\mathbf{n}_i, \mathbf{n}_j)$. The radius r of the circle is the mean value of distances $||\mathbf{x}_i - \mathbf{c}||$ and $||\mathbf{x}_j - \mathbf{c}||$ of points p_i and p_y to the circle centre $\mathbf{c}$. If the difference in these distances is too high (larger than half a meter), we discard the hypothesis. We count the point p_k as an inlier to the current hypothesis if

$$r - r_t \leq ||\mathbf{x}_k - \mathbf{c}|| \leq r + r_t, \tag{5}$$

i.e. it lies on the same circle within a radius threshold t_r.

To prevent the algorithm to find incomplete circles (i.e., the circles that do not have support from different sides), we bin the inliers into eight angular bins over the circle's circumference. The binning process is illustrated in Figure 8.

For every bin, we compute density value P_b that is a ratio of a number of points in the bin b to the total number of points in the circle. To evaluate how equally are the points distributed, we compute the entropy

$$\bar{H}(P) = -\sum_{b=1}^{8} P_b \log_8 P_b \tag{6}$$

that is normalized between zero and one. In the RANSAC procedure, we compare the hypotheses not by the number of inliers, but by the entropy value. The hypothesis with the highest entropy is considered to be the best.

As in the case of mirror symmetries, we run the RANSAC procedure until the confidence in the best hypothesis reaches the value of 99% (again, we use Eq. 4 to determine the confidence). However, to avoid fitting circles to the point clouds that don't have any, we discard the procedure if it reaches the maximal number of iterations.

3.2. Detecting Free-form Shapes

To describe parts of the input model that are not explained by symmetries (free-form parts), we additionally summarize the 2D projected point cloud (including symmetric parts) as a low number of polylines. In other words, our method is designed to extract a set of 2D polylines that approximate the entire floor plan of the object.

As a preprocessing, points lying on near-vertical surfaces (walls, typically) are extracted, as these are more representative for creating a floor plan. We remove points lying on slanted surfaces (e.g. roofs) and horizontal surfaces based on the angle between their 3D normals and the vertical direction, and we aim to summarize the remaining subset of ground-projected points with a low number of 2D polylines.

First, we decompose the 2D point cloud into disjoint local partitions and robustly extract a single dominant 2D line segment from each such partition on-the-fly. The algorithm performs a single run through all points for efficiency, selects the next unpartitioned point as seed, and grows a partition in the 2D k-nearest neighborhood up to a distance threshold. The latter controls the size of the aperture, i.e. the extent of the partitions. The dominant line segment is extracted from each partition using RANSAC [23] (with soft-scoring) and least-squares segment fitting to the inliers. This simple approach tends to preserve locally linear structures while summarizing a 2D point cloud into three orders of magnitude less segments.

Second, the line segments are to be snapped and linked into polylines. To do so, we mark two end-points of segments for snapping if they are nearest neighbors and are within a distance threshold. From these pairwise adjacencies, we identify connected groups of vertices and collapse them into a single vertex at their centroid. Next, the full set of segments is linked into polylines between end-points and junctions in a tracing procedure based on vertex valence, and the resulting polylines are subject to a Douglas-Peucker polygon simplification [49]. In a final cleaning step, short polylines are eliminated. The output is a set of polylines roughly approximating the shape of

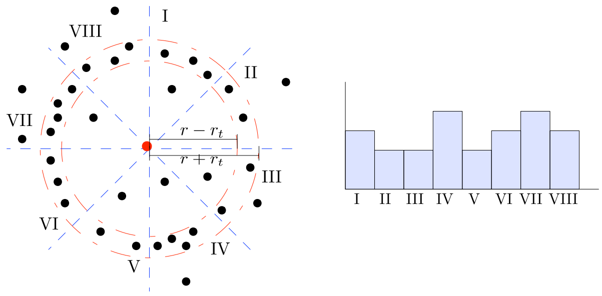


Figure 8: Splitting the circle hypothesis into angular bins. The red lines define a circle hypothesis. The circle is split into eight segments, and points in each are counted to form a density distribution illustrated on the histogram to the right.

dense clusters in the 2D point cloud. An example of detected polylines is shown in Figure 2c and Figure 11.

3.3. Detecting Planes

Planes are a very low-level cue, but are ubiquitous in buildings. Hence, we detect the planes in 3D with RANSAC. We always detect as many planes as needed to explain 90% of points and let the global optimization remove the redundant ones w.r.t. the other structural cues.

3.4. Structural Element Assignment

The final part of our method is the assignment of an architectural structural element to every oriented point $p_i = (\mathbf{x}_i, \mathbf{n}_i)$ in the point cloud, where $\mathbf{x}_i, \mathbf{n}_i \in \mathbb{R}^2$, $i = 1 \ldots, N$ denote the 2D location and normal of point p_i on the ground, respectively. As will be discussed in detail, each structural element is generated by one of the *structural cues*: mirror symmetry, center of body of revolution, free-form polyline or plane described in the sections above.

We formulate the assignment as a multi-label energy minimization, where each point obtains a label $l_i \in \mathcal{L}$ of a structural element, and the optimal labeling $(l_1^*, l_2^*, \ldots, l_N^*) \in \mathcal{L}^N$ is sought for the following energy:

$$\begin{aligned} E(l_1, l_2, \ldots, l_N) = &\sum_{i=1\ldots N} \Theta(p_i, l_i) \\ &+ \gamma_1 \cdot \sum_i \sum_j \Psi(p_i, p_j, l_i, l_j) + \gamma_2 \cdot \sum_{\forall l \in \mathcal{L}} \Gamma(l), \end{aligned} \quad (7)$$

where $\Theta(p_i, l_i)$ is the unary cost of assigning a point p_i to a structural element l_i, $\Psi(p_i, p_j, l_i, l_j)$ is a pairwise term enforcing smoothness of the labeling solution, and $\Gamma(l)$ encodes our prior for a particular structural element l. In the following discussion, we define these terms in detail and explain how each cue generates structural elements.

Our unary cost for each of the structural cues is

$$\Theta(p_i, l_i) = 1 - W^t(l_i) \cdot K^t(\mathbf{x}_i, l_i) \cdot C^t(\mathbf{n}_i, l_i) \cdot H^t(p_i, l). \quad (8)$$

where index $t \in \{\text{mir}, \text{rot}, \text{poly}, \text{plane}\}$ refers to the type of structural cue, i.e., mirror symmetry, rotational symmetry and polyline, respectively, and the terms are as follows:

- $C^t(\mathbf{n}, l) \in [0, 1]$ is a measure of how consistent the normal $\mathbf{n}$ of a point p is with a structural element l,
- $K^t(\mathbf{x}, l) \in [0, 1]$ is a kernel, namely a function of the position $\mathbf{x}$ of a point p w.r.t. a structural element l,
- $W^t(l) \in [0, 1]$ is a weight encoding how important each structural element $l \in \mathcal{L}$ is for the understanding of the building,
- $H^t(p, l) \in [0, 1]$ is a weight encoding how well do other points support the choice of a given point for a structural element l.

We postpone the exact definition of these terms to the next subsections, as they differ per structural cue (index t).

Our pairwise term in Eq. 7 is the weighted Potts penalty

$$\Psi(p_i, p_j, l_i, l_j) = \begin{cases} 0, & l_i = l_j \\ e^{-\lambda_E d_E(p_i, p_j)}, & l_i \neq l_j. \end{cases} \quad (9)$$

This term penalizes any pair of adjacent points (p_i, p_j) having different labels, where adjacency is defined in 3D via k-NN search, $k = 100$. The penalty vanishes with the 3D Euclidean distance $d_E(p_i, p_j)$ between adjacent points, and λ_E controls the speed of decrease. In our experiments, we set λ_E as the smallest ball neighborhood radius such that 75% of all input points have at least 100 neighbors, as also suggested in [15].

Finally, our label cost $\Gamma(l)$ penalizes once for each label $l \in \mathcal{L}$ occurring in the solution, i.e., it reduces the number of structural elements occurring in the solution:

$$\Gamma(l) = \begin{cases} 1, & \exists i : l_i = l \\ 0, & \text{otherwise.} \end{cases} \quad (10)$$

We solve the multi-label optimization efficiently via the α-expansion algorithm [50, 51, 52]. In [53] it is shown that α-expansion is capable of solving energy minimization problems with label cost, such as the one in Eq. 7. The parameters γ_1 and γ_2 balance the relative importance of

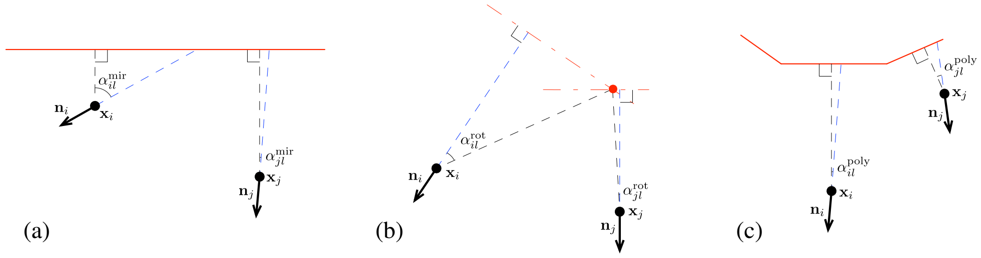


Figure 9: Definition of angle α_{il}^t between a point and a structural cue: (a) mirror symmetry ($t = \text{mir}$), (b) rotational symmetry ($t = \text{rot}$), (c) free-form polyline ($t = \text{poly}$).

the three major energy terms, their choice is discussed in the experiments section.

Next, we explain how each type of structural cue $t = \{\text{mir}, \text{rot}, \text{poly}, \text{plane}\}$ is incorporated into the unary term (Eq. 8) by defining each of its sub-terms $C^t(\mathbf{n}, l)$, $K^t(\mathbf{x}, l)$, $W^t(l)$ and $H^t(p, l)$.

3.4.1. Energy Terms for Mirror Symmetries

Every detected mirror symmetry generates two architectural structural elements (represented by two labels), one for each side of the symmetry axis, designed to separate points into symmetric halves, e.g., opposing walls. As discussed in Section 3.1.2, we extract the two dominant orthogonal mirror symmetries for simplicity and robustness (see Figure 2b). It should be noted, however, that our optimization scheme allows for more than two mirror symmetries.

We now define the sub-terms of Eq. 8 for mirror symmetries ($t = \text{mir}$). The consistency between the normal $\mathbf{n}_i \in \mathbb{R}^2$ of a point p_i and a structural element of label l generated by a mirror symmetry is defined by

$$C^{\text{mir}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{mir}}), \tag{11}$$

where $\alpha_{il}^{\text{mir}} \in [0, \frac{\pi}{2}]$ is the angle between the point's normal $\mathbf{n}_i$ and the normal of the symmetry axis (see Figure 9a). Kernel $K^{\text{mir}}(\mathbf{x}_i, l)$ defines how well do the coordinates of the point support the mirror symmetry. For the structural element l generated by one of the sides of the axis,

$$K^{\text{mir}}(p_i, l) = e^{-\lambda_E d_E(p_i, p_i^*)}, \tag{12}$$

where p_i^* is the closes point to p_i after it is flipped over the mirror symmetry line.

The weight $W^{\text{mir}}(l)$ in Eq. 8 is set to 1 for all structural elements generated by mirror symmetries (high importance). For this cue we also set $H^t(p, l)$ to 1 because mirror symmetries is a global notion that is not dependent on a local point support.

3.4.2. Energy Terms for Rotational Symmetries

Each rotational symmetry generates a single structural element defined by a body of revolution and identified by a label $l \in \mathcal{L}$. Herein, we define the sub-terms of Eq. 8 for rotational symmetries ($t = \text{rot}$).

The consistency between the normal $\mathbf{n}_i \in \mathbb{R}^2$ of a point p_i and the structural element l generated by a rotational symmetry is defined by

$$C^{\text{rot}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{rot}}), \tag{13}$$

where α_{il}^{rot} is the angle between the point's normal $\mathbf{n}_i$ and the line passing through the point and the pivot point of the rotational symmetry (red dot in Figure 9b).

Unlike dominant mirror symmetries, which tend to be more global for a dataset, we aim to extract local rotational symmetries. Therefore, we limit the maximal radii of detected bodies of revolution using a simple kernel:

$$K^{\text{rot}}(\mathbf{x}_i, l) = \begin{cases} 1, & d_E(\mathbf{x}_i, \mathbf{c}) < t_{\text{br}} \\ 0, & \text{otherwise}, \end{cases} \tag{14}$$

where $\mathbf{c}$ is the center of considered body of revolution and t_{br} is the threshold on the maximal circle size (we limit it to half of the maximal point cloud width).

As we would like the detected bodies of revolution to be supported from all sides, we also use the point support weight $H^t(p, l)$. Given a point p_i, and a body of revolution hypothesis l we assume it is assigned to a body of revolution with a center in $\mathbf{c}$. Then, we compute how equally will the points be distributed in a circle with a center in $\mathbf{c}$ and a radius of $d_E(\mathbf{x}_i, \mathbf{c})$. To do so, we use same definition of entropy as in Eq. 6. We split the circle into eight angular bins and count the number of inliers within the same threshold r_t. We assign the value of entropy $\bar{H}(P^*(p_i, l))$ to $H^{\text{rot}}(p_i, l)$, where $P^*(p_i, l)$ is a set of inliers to the current hypothesis of a circle with center in $\mathbf{c}$ going through a point p_i.

As for mirror symmetries, the importance weight $W^{\text{rot}}(l)$ is set to one for all rotational symmetries (high importance).

3.4.3. Energy Terms for Free-form Polylines

Each of the 2D polylines representing the floor plan of the building (Section 3.2) generates a single structural element (hence, a label $l \in \mathcal{L}$). They serve to capture free-form structural elements that are not well explained by symmetry cues. Here, we define the sub-terms of Eq. 8 for $t = \text{poly}$.

The consistency between the normal $\mathbf{n}_i \in \mathbb{R}^2$ of a point p_i and a polyline identified by label l is

$$C^{\text{poly}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{poly}}), \tag{15}$$

where $\alpha_{il}^{\text{poly}}$ is the angle between the point's normal $\mathbf{n}_i$ and the line passing through p_i and the closest point $\hat{p}_{il}$ of the polyline, as shown in Figure 9c.

Similarly to rotational symmetries, polylines have a local influence, encoded in the sigmoid kernel

$$K^{\text{poly}}(\mathbf{x}_i, l) = 1 - \frac{1}{1 + \exp\{\lambda_p(\tau_p - ||\mathbf{x}_i - \hat{\mathbf{x}}_{il}||)\}}, \tag{16}$$

where $\mathbf{x}_i, \hat{\mathbf{x}}_{il} \in \mathbb{R}^2$ are the 2D positions of p_i and of the closest point $\hat{p}_{il}$ on the polyline, respectively. The inflection point of the sigmoid is fixed at $t_p = 1$ meter, and the steepness of the descent to $\lambda_p = 20$ in all our experiments.

A strong symmetry cue should dominate over polylines in its area of influence. Therefore, each polyline needs to be weighted depending on how well it can fit to any symmetry cue. In this vein, we extract the oriented mid-point m_{lk} from each segment e_{lk} of every polyline l, and evaluate $\Theta(m_{lk}, l')$ defined in Eq. 8 for each structural element l' generated by detected mirror (t = mir) and rotational (t = rot) symmetries. $\Theta(m_{lk}, l')$ measures how well the mid-point m_{lk} fits to a symmetry cue. We introduce the polyline *usefulness* measure

$$U_l = \sum_k f_{lk} \cdot \min(\sum_{l'=1\ldots L} \Theta(m_{lk}, l'), 1), \tag{17}$$

where f_{lk} is the normalized length of segment e_{lk} of polyline l, such that $\sum_k f_{lk} = 1$. A high U_l indicates that the polygon contains a large number of segments unexplained by symmetry cues, hence, it is worth considering the polyline to explain the neighboring points as free-form parts. The weight of the polyline is defined as the sigmoid

$$W^{\text{poly}}(l) = \frac{1}{1 + \exp\{\lambda_w(t_w - U_l)\}}, \tag{18}$$

where we set an abrupt change $\lambda_w = 20$ similarly to Eq. 16, and $t_w = 0.6$ in all experiments.

3.4.4. Energy Terms for Planes

For each plane, we define the weight terms as follows. The consistency between the point p_i with a normal $\mathbf{n}_i \in \mathbb{R}^2$ and the structural element l that is defined by plane is

$$C^{\text{rot}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{plane}}), \tag{19}$$

where α_{il}^{rot} is the angle between the point's normal $\mathbf{n}_i$ and normal of the plane.

The distance between the point and the plane is captured by the kernel $K^{\text{plane}}(\mathbf{x}_i, l)$:

$$K^{\text{plane}}(\mathbf{x}_i, l) = e^{-\lambda_E d_E(p_i, p_i^*)}, \tag{20}$$

where p_i^* is the closest point on the plane to p_i.

We also have a preference to planes that are equally supported by points. To evaluate that, we study the 2D point distribution of the plane inliers. First, we compute a 2D convex hull for the point inliers on the plane. Then, we split the space inside the hull into n_s 1 by 1 meter squares, and compute the fraction of points for every square P_n. Finally, we assign the weight

$$\bar{H}(p_i, l) = -\sum_{b=1}^{P_n} P_n \log_{P_n} P_n \tag{21}$$

that equals to 1 if the points are completely uniformly distributed and fades to 0 as the distribution becomes less equal. We assign the same weight $H(p_i, l)$ to every point p_i. As the planes are robust to detect, we also set their importance to $W^{\text{plane}}(l) = 1$.

4. Experiments

In this section we evaluate the properties of our *architectural structural elements* (ASE) decomposition. All used 3D models are reconstructed from unordered image collections. Despite tremendous progress in the research, these models are still incomplete and very noisy compared to clean computer generated collections used in prior work. This method is tested on multiple datasets that have different architectural elements like towers, arches, ellipsoids, planar and free-form walls, as summarized in Table 1. The majority of the datasets are a result of running SfM pipeline, however, some (namely, Fraumuenster and Grossmuenster) are sampled from a city-scale aerial mesh (and are very coarse). The results are also shown in Figure 15 and 16.

We evaluate our results quantitatively by comparing them with our ground truth segmentations. The segmentation accuracy is measured by the intersection over union (also Jaccard index) for every ground truth label and the corresponding label. Due to possible over-segmentation, a mapping is needed between the predicted labeling (where the number of labels can vary) to the fixed labels of the ground truth segmentation. We select the most frequent label per ground truth label, mark it used and as corresponding to the ground truth label. This is a similar procedure as used in the PASCAL VOC challenge [54]. We report the mean accuracy over all labels.

4.1. Method Parameters

In this section we investigate the stability of our method against the internal parameters. Since all the models are metrically scaled, the geometric parameters are fixed to meter scale as described in the text above. The main remaining parameters are the multi-label optimization weights γ_1 (smoothing cost) and γ_2 (label cost). We empirically evaluate the mean accuracy over the ground truth labels as well the number of labels that appear in the final decomposition. Using a grid search over a range of feasible parameters we fix these for all other experiments.

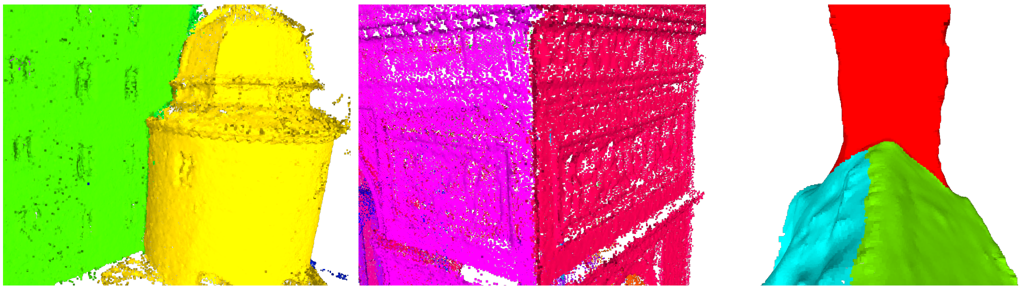


Figure 10: Close-up pictures of the seams between adjacent ASEs demonstrate the accuracy of our method. Left to right: Orebro, junction between the tower and the wall; Arch: two walls; Fraumuenster: rooftop and the tower.

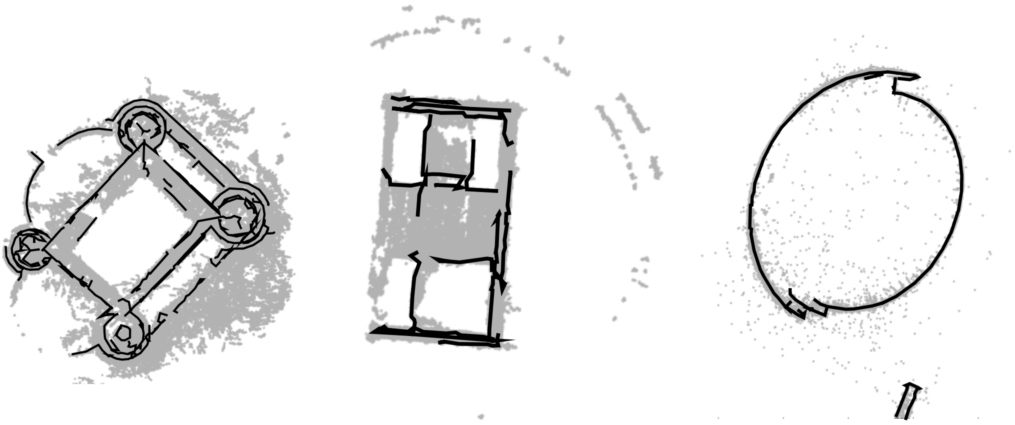


Figure 11: Examples of detected free-form shapes on the datasets. Left to right: Orebro, Arch, Colosseum.

Dataset	# Pts	3D Source	Method	Segmentation accuracy [%]											
				Per label, colored by type of architectural structural element (ASE)											Mean
Orebro [55]	1.7M	PMVS [56]	Curvature split	86.7	86.7	83.3	87.2	72.3	76.6	74.7	66.5	78.4	71.2	15.2	72.64
			Plane fitting	79.9	82.8	89.0	79.0	14.6	18.7	13.8	27.3	32.4	8.1	8.1	41.25
			Method [24]	75.2	78.0	81.0	77.9	68.5	63.6	70.9	53.0	39.7	59.4	34.6	63.80
			Our method	97.0	96.4	98.4	96.7	86.7	89.4	98.3	94.9	100.0	98.8	93.3	**95.45**
Arch [55]	300K	PMVS [56]	Curvature split	75.4	61.3	62.7	6.4	17.5	17.2	19.7	4.9	21.3			31.81
			Plane fitting	18.5	91.6	86.9	3.9	79.0	55.2	60.3	0.1	35.1			47.84
			Method [24]	47.1	62.8	63.1	8.9	47.0	54.5	52.6	16.5	57.5			45.60
			Our method	92.3	92.2	84.8	8.4	88.2	83.4	62.5	40.8	92.2			**71.65**
Colosseum [15]	200K	SfM [3]	Curvature split	88.4	89.1	38.9									72.14
			Plane fitting	17.9	7.7	29									18.20
			Method [24]	49.3	52.8	35.0									45.70
			Our method	99.9	78.6	98.8									**92.43**
Sant'Angelo [57]	60K	SfM [3]	Curvature split	37.9	25.4	66.2	92.1	62.3	82.3	42.7	32.8				54.10
			Plane fitting	39.5	33.3	8.6	22.3	77.7	0	7.7	18.1				25.90
			Method [24]	91.1	58.6	33.7	68.2	83.2	97.3	97.5	99.6				78.65
			Our method	59.4	55.5	100	99.8	97.5	98.3	96.1	89.1				**86.96**
Grossmuenster	200K	MVS [58]	Curvature split	2.8	2.6	2.6	0.4	2.2	2.4	0.8					1.97
			Plane fitting	54.5	42.6	28.6	0	20.2	19.6	11.8					25.35
			Method [24]	43.3	32.3	63.8	0	24.4	18.9	65.4					35.45
			Our method	81.7	54.5	0	0	79.3	87.2	6.2					**44.13**
Fraumuenster	120K	MVS [58]	Curvature split	1.8	1.2	2.5	0.9								1.59
			Plane fitting	68.0	66.0	22.4	82.7								59.82
			Method [24]	64.0	66.2	20.0	76.03								56.56
			Our method	86.3	88.4	84.7	77.6								**84.25**
Trinity Chapel (failure case)	200K	SfM [3]	Curvature split	24.7	28.4	32.9	16.7	32.8							27.10
			Plane fitting	24.7	55.9	24.3	26.0	9.5							28.08
			Method [24]	28.2	29.2	44.2	24.3	61.8							37.54
			Our method	56.6	39.5	67.2	21.8	56.1							**48.24**

Table 1: Overview of the datasets and comparison of our method with the baselines. Results are reported as segmentation accuracy (Jaccard index multiplied by 100). Colors correspond to the components identified by our method: red for mirror symmetries, green for bodies of revolution, yellow for polylines, blue for planes.

The final values are $\gamma_1 = 1$ (smoothing cost) and $\gamma_2 = 100$ (label cost). Increasing $\gamma_1 = 1$ smooths the labels. It is robust in the range of $1 \ldots 5$. Changing the value of γ_2 affects the number of detected labels and is very robust. Only very low values result in extra labels and hence in over-segmentation. Results of the grid search for these parameters is shown in Figure 12.

Hough voting and RANSAC give qualitatively similar results, and the main difference is the time needed for computation. This is shown in Figure 13. In this plot, we compare the time needed for finding for a mirror symmetry by RANSAC and Hough voting. Hough voting is implemented efficiently on C++, while RANSAC is computed with Matlab. However, due to the quadratic complexity of Hough voting, it is an order of magnitude slower on a point cloud with 350K points. In the future experiments, we use RANSAC for detecting ASEs as the point clouds we are operating with are relatively large (up to 1.7M points).

Regarding the time of computation, the main bottleneck of the algorithm is the final 3D optimization. In Figure 14, we show the time taken to do a full optimization on Orebro dataset as a function of the number of points in the dataset. The computational time grows linearly with the number of points.

4.2. Comparison to Baselines

In this section we evaluate our method against baseline approaches. Because of a lack of available, directly comparable work, we compare to alternative ways of unsupervised decomposition of a point cloud. Namely, we investigate curvature-based segmentation and hard-prior approaches for grouping parts of the 3D point cloud into coherent groups.

The first baseline is inspired by [31]. We detect regularized regions of different curvatures, and run connected of components search to identify the components.

As a second baseline, we iteratively fit planes with an inlier threshold of 0.1 meters until 90% of the point cloud is assigned to a plane. For every ground truth structural element, we choose one plane that is best at covering it.

As a third baseline, we use a method of Schnabel et al. [24] that decomposes a point cloud into a set of primitives like planes, tori, spheres, etc.

The results are provided in Figure 15 and 16 as well as in Table 1. As noise in the ground truth is not segmented, to compare the decomposition to the ground truth, we search the connected components of the resulting ASE decomposition and match the closest connected component to the corresponding ground truth element. This we also show on top view visualizations in Figure 15, 16.

The curvature split baseline achieves reasonable results on the meshes that are not too smooth. On Fraumuenster and Grossmuenster datasets (that are extracted from an aerial mesh), the transition between different components is so smooth the curvature does not capture it. On other datasets, for example, Orebro, the method is capable of delivering a reasonable decomposition. It should be noted that sharp edges on parts of the components, for example, a transition from roof to wall in towers, can lead to splitting of the component into multiple ones.

Plane fitting produces reasonable results when the objects to decompose are flat. For suitable landmarks, such as Arch, it is able to capture the ASEs very well (although ignoring the internal structure of the object). Orebro's walls are identified reasonably well, resulting in high scores for their segmentation. The method fails with towers and arbitrarily shaped constructions, resulting in a relatively low mean accuracy for this dataset. Finally, in the absence of planar objects, this method does not provide any reasonable solution (see Colosseum).

The method from [24] is very efficient at fitting primitives into point clouds, as it can be seen on, for example, Arch and Fraumuenster datasets. However, it suffers from several drawbacks. First, in the absence of clear primitive it fails to fit a sensible model (for example, in case of Colosseum, every side is shared between several models). Our method is capable of dealing with these elements because of the free-form polylines cue. Second, the primitives are too simple for buildings. This is seen or towers of Orebro and Sant'Angelo. Different parts of towers that have different width get different primitives (namely, cylinders) assigned to them.

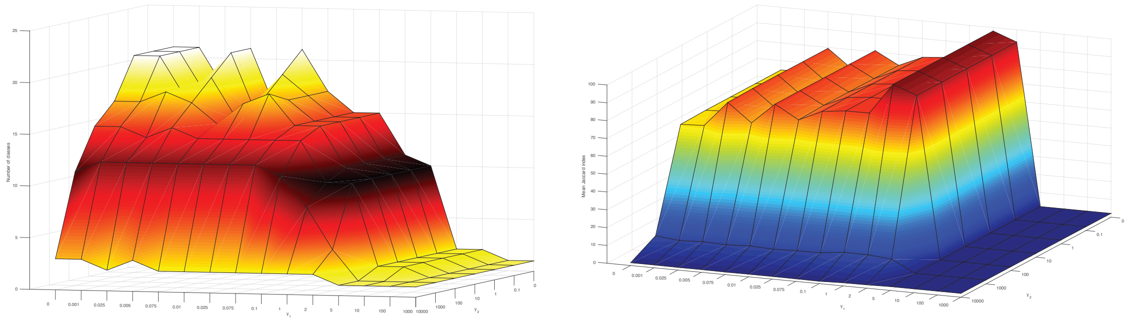
Our method successfully and consistently outperforms the baselines as it is optimized for multiple specialized structural features rather than hard constraints or split based on geometrical pecularities. It can be also seen in the quantitative evaluation shown in Table 1.

It prioritizes the structural cues based on the dataset. For Colosseum, that does not have a strong support for rotational or mirror symmetries, it explains the whole model based on the polylines. Conversely, Arch is completely explained by the mirror symmetries, resulting for good scores for all components but the back wall, where the point support is very weak. For Orebro, the method is able to capture all four wall and side towers, as well as the two half moon-shaped structures at the entrance. On every wall the mirror symmetries and plane cues have to compete as both have high unaries. As a result, two walls are assigned to planes, and two – to the mirror symmetries.

In case of Grossmuenster, all ASEs are successfully determined. However, because of too high influence of the pairwise term in Eq. 7, the area assigned to a body of revolution of a small tower is extended to parts of the building itself.

Finally, the Trinity Chapel is a very noisy point cloud with several SfM errors and very irregular point structure. All considered methods fail to decompose it. For our method, the main challenges related to this datasets were irregular point normals and very irregular point density that made smoothing extremely challenging.

The quality of the segmentation is additionally illustrated in Figure 10, where we show close-up pictures of the seams between different components.



(a) Number of detected labels. The darker the color, the closer to the ground truth number of labels.

(b) Segmentation accuracy [%]. The higher the value, the more correct w.r.t. ground truth.

Figure 12: Results of the grid search over the weights of the energy optimization for Orebro dataset.

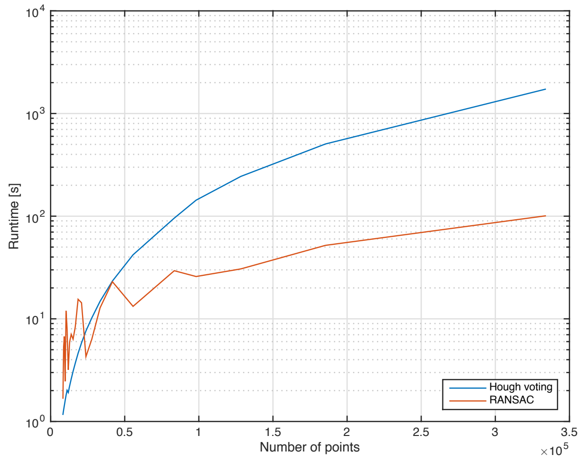


Figure 13: Running time for detecting a mirror symmetry for Hough space voting and RANSAC. RANSAC significantly outperforms Hough space voting as the number of points grows. Please note that the time axis is in logarithmic scale.

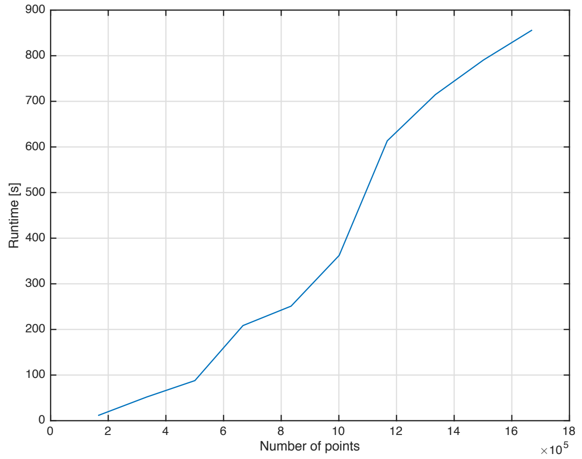


Figure 14: Running time of the 3D optimization (measured on a single core). The subset of points (from Orebro dataset) are generated by parts of the same dataset, keeping the point density the same).

In Figure 11 we demonstrate our method for detecting free-form shapes on several datasets. While detecting floor plans in 2D point cloud projections can be simplified by a number of assumptions (for example, walls can be assumed to be planar and be therefore detected with a line detector), our method is targeted towards general shapes like curved objects (as straight walls can be easily detected with other cues, namely mirror symmetries and planes). It can be seen that the more precise the data is (for example, on Colosseum dataset), the more likely is the method to return one polyline per component. In case of noisy thick walls, like in the Arch dataset, a single wall can be represented with multiple polylines. Adding data awareness to our current method can result in better performance of very noisy sets.

Overall our method successfully decomposes all models into their ASEs. Failure cases occur when the symmetries or free-form lines are not correctly detected. Usually this is the case for missing data, as this affects any of the methods as no underlying structural cues exist.

5. Conclusions and Future Work

This work takes a step towards understanding the architectural and structural elements of landmarks. Although for simple buildings a simple planar abstraction should suffice, we look at architectural landmarks which contain more complex shapes and structure than usual buildings.

Our method for decomposing 3D reconstructions exploits multiple structural cues like mirror and rotational symmetries, free-form polylines and planes. As we formulate it as a multi-label optimization, our method works on noisy 3D point clouds from image-based reconstruction. Experimental evaluation confirms that the results are state-of-the-art for the decomposition of complex landmark buildings.

In future work we plan to iterate symmetry extraction and structural element assignment and infer long-distance graph connections to complete empty areas of the 3D reconstruction to overcome some aspects of the missing data,

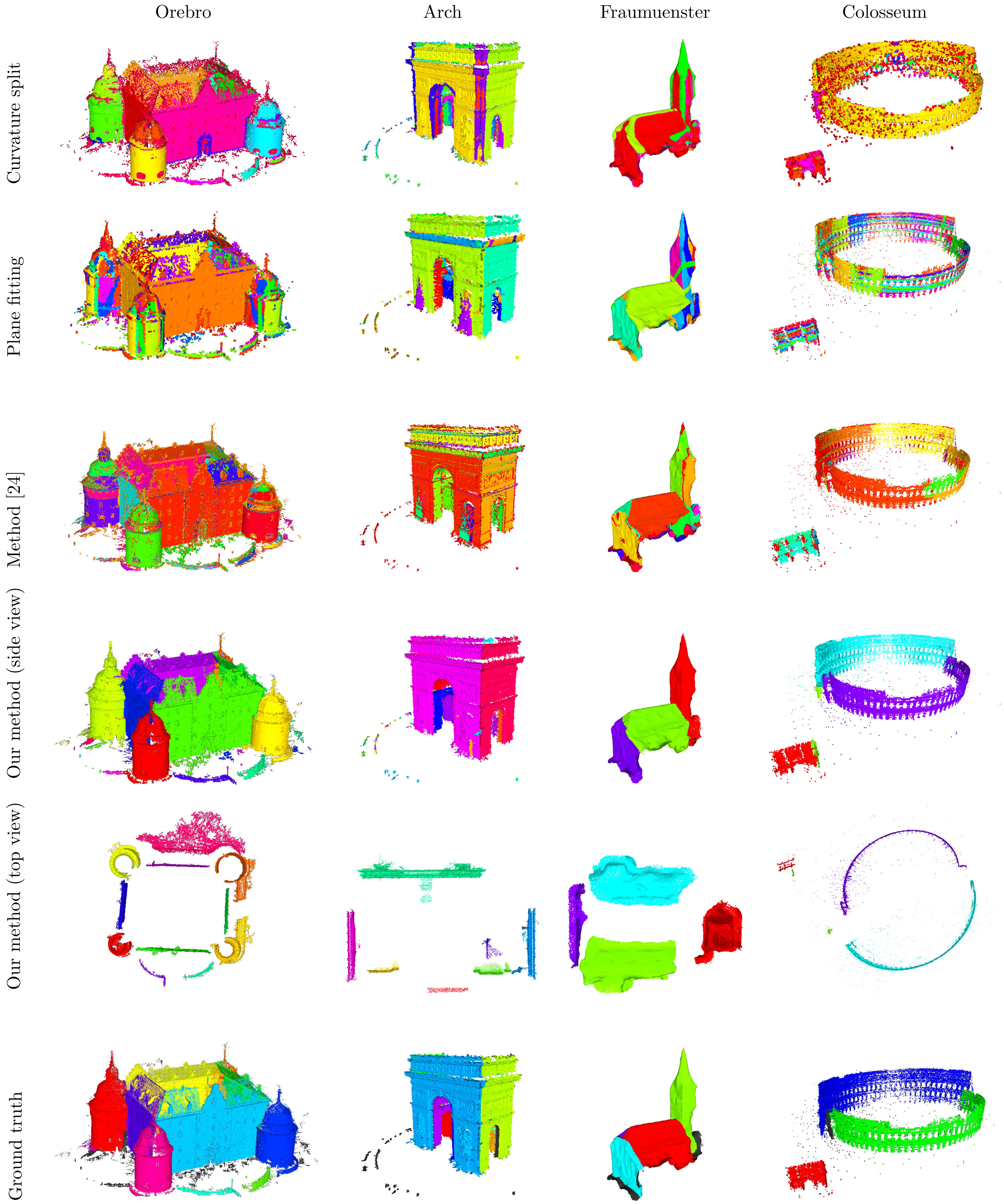


Figure 15: Examples of our method in comparison to standard baseline methods. The baselines fail to capture all structural elements of the landmarks as they either contain no higher level features (curvature) or enforce hard priors (plane fitting). Our method successfully decomposes the complex scenes due to the global optimization of our structural features.

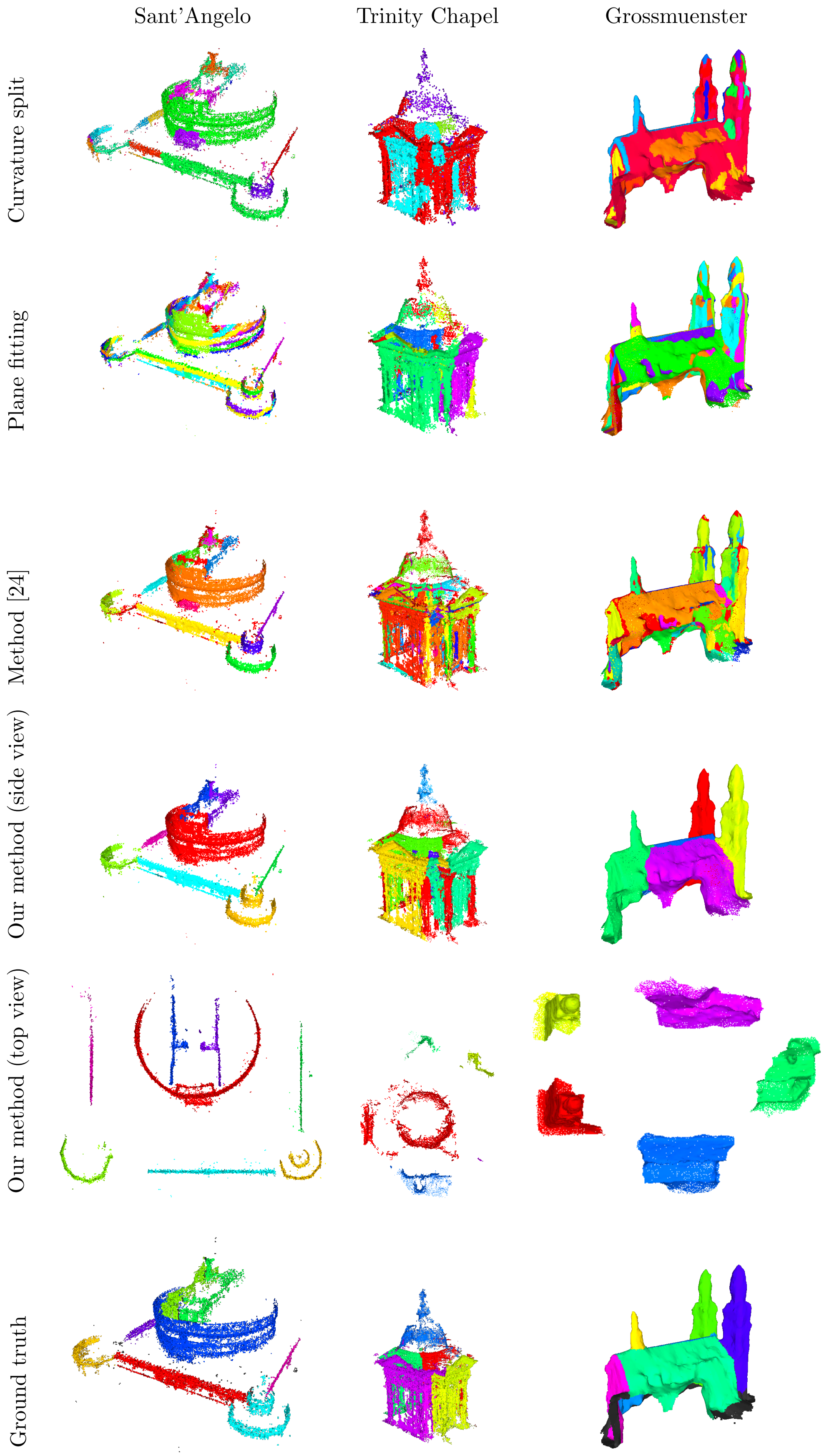


Figure 16: Examples of our method in comparison to standard baseline methods, continued.

as well as to enlarge the list of detected ASEs as well as researching the extraction of general 3D shapes [59, 60]. Another contribution we leave for future work is integrating noise detection into optimization. That will facilitate simultaneous ASE decomposition and generally image-based structure-from-motion dataset denoising.

The understanding of 3D architectural models such as buildings and landmarks, will pave the way for more automation of creating talk maps [61] and creating more structurally correct textured 3D models [62] for better visualization.

Acknowledgments. This work was supported by the European Research Council (ERC) project VarCity (#273940) and the Institute of Photogrammetry and Remote Sensing, ETH Zurich by providing the aerial imagery.

References

[1] A. Locher, M. Perdoch, H. Riemenschneider, L. Van Gool, Mobile phone and cloud – a dream team for 3d reconstruction, in: WACV, 2016.

[2] D. Tingdahl, L. J. Van Gool, An enhanced on-line service for 3D model construction from photographs, International Journal of Heritage in the Digital Era 1 (2) (2012) 277–294.

[3] C. Wu, Towards linear-time incremental structure from motion, in: 3DPVT, 2013.

[4] S. Agarwal, N. Snavely, I. Simon, S. Seitz, R. Szeliski, Building rome in a day, in: ICCV, 2009.

[5] Y. Furukawa, J. Ponce, Accurate, Dense, and Robust Multi-View Stereopsis, PAMI 32 (8) (2010) 1362–1376.

[6] V. Hiep, P. Labatut, J. Pons, R. Keriven, High Accuracy and Visibility-Consistent Dense Multi-view Stereo, PAMI 34 (5) (2012) 889–901.

[7] P. Müller, G. Zeng, P. Wonka, L. Van Gool, Image-based procedural modeling of facades, in: SIGGRAPH, 2007.

[8] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, H. Bischof, Irregular lattices for complex shape grammar facade parsing, in: CVPR, 2012.

[9] A. Toshev, P. Mordohai, B. Taskar, Detecting and parsing architecture at city scale from range data, in: CVPR, 2010.

[10] T. Werner, A. Zisserman, New techniques for automated architecture reconstruction from photographs, in: ECCV, 2002.

[11] S. Sinha, D. Steedly., R. Szeliski, Piecewise Planar Stereo for Image-based Rendering, in: ICCV, 2009.

[12] A. Bódis-Szomorú, H. Riemenschneider, L. Van Gool, Fast, Approximate Piecewise-Planar Modeling Based on Sparse Structure-from-Motion and Superpixels, in: CVPR, 2014.

[13] A. Bódis-Szomorú, H. Riemenschneider, L. Van Gool, Superpixel meshes for fast edge-preserving surface reconstruction, in: CVPR, 2015.

[14] C. Vanegas, D. Aliaga, B. Benes, Building reconstruction using manhattan-world grammars, in: CVPR, 2010.

[15] C. Wu, S. Agarwal, Schematic surface reconstruction, in: CVPR, 2012.

[16] A. Dame, V. Prisacariu, C. Ren, I. Reid, Dense Reconstruction Using 3D Object Shape Priors, in: CVPR, 2013.

[17] Y. Bao, M. Chandraker, Y. Lin, S. Savarese, Dense Object Reconstruction with Semantic Priors, in: CVPR, 2013.

[18] M. Berger, A. Tagliasacchi, L. Seversky, P. Alliez, J. Levine, A. Sharf, C. Silva, State of the art in surface reconstruction from point clouds, in: EUROGRAPHICS, 2014.

[19] O. Teboul, L. Simon, P. Koutsourakis, N. Paragios, Segmentation of building facades using procedural shape prior, in: CVPR, 2010.

[20] A. Martinovic, M. Mathias, J. Weissenberg, L. Van Gool, A Three-Layered Approach to Facade Parsing, in: ECCV, 2012.

[21] A. Martinovic, J. Knopp, H. Riemenschneider, L. Van Gool, 3d all the way: Semantic segmentation of urban scenes from start to end in 3d, in: CVPR, 2015.

[22] F. Lafarge, R. Keriven, M. Bredif, H. Vu, A hybrid multi-view stereo algorithm for modeling urban scenes, PAMI 35 (1) (2013) 5–17.

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

[24] R. Schnabel, R. Wahl, R. Klein, Efficient ransac for point-cloud shape detection, Computer Graphics Forum 26 (2) (2007) 214–226.

[25] R. Schnabel, P. Degener, R. Klein, Completion and reconstruction with primitive shapes, Computer Graphics Forum 28 (2) (2009) 503–512.

[26] C. Haene, C. Zach, B. Zeisl, M. Pollefeys, A Patch Prior for Dense 3D Reconstruction in Man-Made Environments, in: 3DPVT, 2012.

[27] A. Monszpart, G. J. Brostow, RAPTER : Rebuilding Man-made Scenes with Regular Arrangements of Planes, in: Siggraph, 2015.

[28] A. Boulch, M. De La Gorce, R. Marlet, Piecewise-planar 3D reconstruction with edge and corner regularization, Computer Graphics Forum 33 (5) (2014) 55–64.

[29] A. Chauve, P. Labatut, J. Pons, Robust piecewise-planar 3d reconstruction and completion from large-scale unstructured point data, in: CVPR, 2010.

[30] S. Oesau, F. Lafarge, P. Alliez, Planar Shape Detection and Regularization in Tandem, Computer Graphics Forum (2015) 1–14.

[31] F. Lafarge, R. Keriven, M. Bredif, Combining meshes and geometric primitives for accurate and semantic modeling, in: BMVC, 2009.

[32] Y. Verdie, F. Lafarge, Detecting parametric objects in large scenes by monte carlo sampling, IJCV 106 (1) (2014) 55–75.

[33] F. Lafarge, R. Keriven, M. Bredif, H. Vu, Hybrid multi-view reconstruction by jump-diffusion, in: CVPR, 2010.

[34] T. T. Pham, T. J. Chin, K. Schindler, D. Suter, Interacting geometric priors for robust multimodel fitting, IEEE Transactions on Image Processing 23 (10) (2014) 1.

[35] Y. Liu, H. Hel-Or, C. Kaplan, L. Van Gool, Computational Symmetry in Computer Vision and Computer Graphics, Foundations and Trends in Computer Graphics and Vision 5 (1) (2009) 1–195.

[36] N. J. Mitra, M. Pauly, M. Wand, D. Ceylan, Symmetry in 3D geometry: Extraction and applications, Computer Graphics Forum 32 (6) (2013) 1–23.

[37] N. J. Mitra, L. J. Guibas, M. Pauly, Partial and approximate symmetry detection for 3D geometry, in: SIGGRAPH, 2006.

[38] N. J. Mitra, L. J. Guibas, M. Pauly, Symmetrization, in: SIGGRAPH, 2007.

[39] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, L. J. Guibas, Discovering Structural Regularity in 3D Geometry, SIGGRAPH.

[40] A. Cohen, C. Zach, S. N. Sinha, M. Pollefeys, Discovering and exploiting 3D symmetries in structure from motion, in: CVPR, 2012.

[41] K. Kser, C. Zach, M. Pollefeys, Dense 3d reconstruction of symmetric scenes from a single image, in: DAGM, 2011.

[42] Y. Zheng, D. Cohen-Or, M. Averkiou, N. J. Mitra, Recurring part arrangements in shape collections, Computer Graphics Forum 33 (2) (2014) 115–124.

[43] H. Liu, U. Vimont, M. Wand, M.-P. Cani, S. Hahmann, D. Rohmer, N. J. Mitra, Replaceable Substructures for Efficient Part-Based Modeling, in: EUROGRAPHICS, 2015.

[44] D. Ballard, Generalizing the hough transform to detect arbitrary shapes, PR 33 (2) (1981) 111–122.

[45] P. Hough, Machine analysis of bubble chamber pictures, in: International Conference on High Energy Accelerators and Instrumentation, 1959.

[46] A. E. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3d scenes, PAMI 21 (5) (1999) 433–449.

[47] R. B. Rusu, N. Blodow, M. Beetz, Fast Point Feature Histograms (FPFH) for 3D Registration, in: ICRA, 2009.

[48] J. Knopp, M. Prasad, G. Willems, R. Timofte, L. Van Gool, Hough transform and 3d surf for robust three dimensional classification, in: ECCV, 2010.

[49] D. Douglas, T. Peucker, Algorithms for the reduction of the number of points required to represent a digitized line or its caricature, The Canadian Cartographer 10 (2) (1973) 112–122.

[50] Y. Boykov, O. Veksler, R. Zabih, Fast approximate energy minimization via graph cuts, PAMI 23 (11) (2001) 1222–1239.

[51] Y. Boykov, V. Kolmogorov, An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision, PAMI 26 (9) (2004) 124–1137.

[52] V. Kolmogorov, R.Zabih, What energy functions can be minimized via graph cuts?, PAMI 26 (2) (2004) 147–159.

[53] A. Delong, A. Osokin, H. N. Isack, Y. Boykov, Fast Approximate Energy Minimization with Label Costs, International Journal of Computer Vision 96 (1) (2011) 1–27.

[54] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results.

[55] C. Olsson, O. Enqvist, Stable structure from motion for unordered image collections, in: SCIA, 2011.

[56] Y. Furukawa, B. Curless, S. Seitz, R. Szeliski, Towards Internet-scale Multi-view Stereos, in: CVPR, 2010.

[57] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, F. Wu, 3D Visual Phrases for Landmark Recognition, in: CVPR, 2012.

[58] M. Jancosek, T. Pajdla, Multi-View Reconstruction Preserving Weakly-Supported Surfaces, in: CVPR, 2011.

[59] E. Kalogerakis, A. Hertzmann, K. Singh, Learning 3d mesh segmentation and labeling, ACM Transactions on Graphics 29 (4) (2010) 102.

[60] A. Mansfield, N. Kobyshev, H. Riemenschneider, W. Chang, L. Van Gool, Frankenhorse: Automatic completion of articulating objects from image-based reconstruction, in: BMVC, 2014.

[61] J. Weissenberg, M. Gygli, H. Riemenschneider, L. Van Gool, Navigation using special buildings as signposts, in: MapInteract, 2014.

[62] D. Dai, H. Riemenschneider, G. Schmitt, L. Van Gool, Example-based facade texture synthesis, in: ICCV, 2013.