Architectural Decomposition for 3D Landmark Building Understanding

Nikolay Kobyshev[1] Hayko Riemenschneider[1] András Bódis-Szomorú[1] Luc Van Gool[1,2]

[1] CVL, ETH Zurich, Switzerland [2] VISICS, KU Leuven, 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 three types of structural cues, namely dominant mirror symmetries, rotational symmetries, and polylines capturing free-form shapes of the landmark not explained by symmetry. Combining these cues enables modeling the variability present in complex 3D models, and robustly decomposing them into architectural structural elements. Our architectural decomposition facilitates significant 3D model compression and shape-specific modeling.

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) [52], internet-scale point cloud reconstruction from imagery [1], high-accuracy detailed surface reconstructions via dense Multi-View Stereo (MVS) [15, 19], and achieved success in procedural modeling of facades [37]. LiDAR is an alternative dominant technology to obtain point clouds of urban scenes [48].

In this work [1], we tackle the abstraction and understanding of 3D point clouds delivered by such state-of-the-art technologies. Planar priors [51, 46], or a Manhattan-world assumption [48] proved to be satisfactory for many man-made structures. However, for a large mass of buildings, especially landmark architecture or general objects, a simple planar abstraction will not suffice. We propose a method for abstracting and decomposing 3D reconstructions by exploiting self-similarities within the model. Such a decomposition is a first step towards understanding and compactly modeling the architectural elements of a landmark.

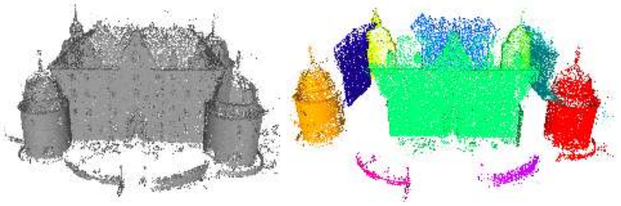

Figure 1: Our method segments a point cloud of a complex landmark building into coherent architectural structural elements (ASE), such as walls, towers and free-form parts based on symmetry cues only.

Our method is based on weak architectural priors that naturally hold for a majority of buildings, namely mirror symmetries, rotational symmetries and vertical wall. 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, 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 wall, tower, arch etc., as illustrated in Figure 1. We refer to these components as *architectural structural elements* (ASEs) in the paper. We formulate the decomposition problem as an energy-driven, multi-label point cloud segmentation, which we solve efficiently via graph cuts. Our contributions:

- 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 rotational symmetries, 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.

[1] This work was supported by the European Research Council (ERC) under the project VarCity (#273940) at www.varcity.eu.

Our work paves the road to 3D model compression [53], shape-specific models [10, 3] and guided navigation [50].

2. Related Work

Creating solid models 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 [4] there may be other constraints. For example, for simple Manhattan-style skyscrapers the modeling can be as simple as a rectangular box [48]. For Haussmannian architecture, strong priors or regular floors may be sufficient to model the buildings [47]. For more general architecture, more relaxed structural principles such as symmetries have to be used [33, 41, 32]. Further even, in the case of real cities with regular planar buildings and complex shapes like statues, a hybrid model [26] or a topology joining approach [31] may be applied. Finally, semantic segmentation approaches [16, 40, 22, 45] may be useful for urban scenes.

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

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 assignment), or an attraction force to the fitted shape primitive is used (soft assignment). 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 [13]. Schnabel et al. [44] detect simple primitives such as planes, spheres, cylinders, etc. and further extend shape primitives across the remaining surfaces for completion [43, 54].

For the soft assignments 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. [17] have shown this for piecewise planar priors and the works of Dame and Bao [10, 3] showed this for arbitrary shape priors learned from 3D training data.

Bodis et al. [5, 6] follow a different approach and directly minimize the surface to extract locally planar superpixel surfaces to avoid complexity of shape primitives.

Lafarge et al. [25] 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 [49] proposed an efficient Monte Carlo sampler for detecting parametric objects in large scenes exploiting parallel processing and reversible jump between different primitive types.

Lafarge et al. [26] 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.

Overall, these methods provide a better understanding through shape priors, yet cannot handle large, complex shapes such as architectural elements in entire landmark buildings. To the best of our knowledge, [53] is the only related work for abstraction of MVS-obtained buildings as it tries to decompose buildings into 2D sweeping profiles. However, in our experience, iteratively finding profiles has problems separating the actual architectural elements.

Symmetry Detection Symmetries have been explored in computer vision for a long time and review reports are available [30, 36]. However, among the first to apply it for 3D buildings were [34, 35, 39].

Mitra et al. [34] introduce a voting for symmetries for reflection, rotation and translation. In their follow-up work, [35], 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. [9] take it one step further and let the symmetries influence the Structure-from-Motion optimization, whereas Koeser at al. [24] exploit mirror symmetries in dense reconstructions from a single view.

Zheng et al. [55] rearrange parts of objects within large yet clean shape collections. In their recent work, Liu et al. [28] define replaceable substructures and use the shape graph for remodeling.

Contrary to these approaches, we use noisy MVS point clouds and tackle the decomposition and understanding of real-world 3D landmark reconstructions. Hence, we aim for the segmentation of such data into architectural structural elements (ASE). In particular, we propose a method for segmenting complex landmark structures into parts using a range of structural cues such as mirror symmetries, rotational symmetries 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 of the building. Fig. 2 gives an overview of our approach.

In a preprocessing step, we align our input point cloud with the gravity vector 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. Exploiting the natural vertical prior for walls, we project the points and their normals to the ground. We denote an oriented 2D point by $p_i = (\mathbf{x}_i, \mathbf{n}_i)$ and its height above the ground by h_i. Further, we perform detection of structural cues on the 2D point cloud.

First, we analyze the point cloud to detect its dominant mirror symmetries (to find opposing walls) and rotational symmetries (to detect surfaces of revolution, e.g. towers) (Fig. 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 (Fig. 2c). 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 (Fig. 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 3.3. The 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 other types of symmetries and to 3D symmetries. Our main scheme for collecting symmetry evidence is Hough-space voting [2, 20]. Inspired by [34], we generate votes for each pair of points for a symmetry in Hough-space.

3.1.1 Point Matching

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

$$|h_i - h_j| < t_h, \quad (1)$$

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

$$t_h = 0.1 \cdot (\max_i h_i - \min_i h_i). \quad (2)$$

We note that this simple criterion could be replaced by a more sophisticated matching of local 3D shape descriptors, e.g. spin images [29], FPFH [42] or 3D SURF[21].

3.1.2 Detecting Mirror Symmetries

For mirror symmetries, every matching pair of oriented points (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 Fig. 3a. This symmetry line is parametrized by a pair (D_{ij}, ϕ_{ij}) (i.e. a point) in the Hough-space $\mathcal{H}^{\text{mir}}(D, \phi)$, 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 Hough-space votes for the example of Fig. 2a.

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

$$\mathcal{H}^{\text{mir}}(D_1, \phi) + \mathcal{H}^{\text{mir}}(D_2, \{\phi + \pi/2\} \text{ mod } 2\pi) \quad (3)$$

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

This is solved exhaustively, where for each discrete value of ϕ, the maximal $\mathcal{H}^{\text{mir}}(D_1)$ and $\mathcal{H}^{\text{mir}}(D_2)$ 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.

3.1.3 Detecting Rotational Symmetries

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 Fig. 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

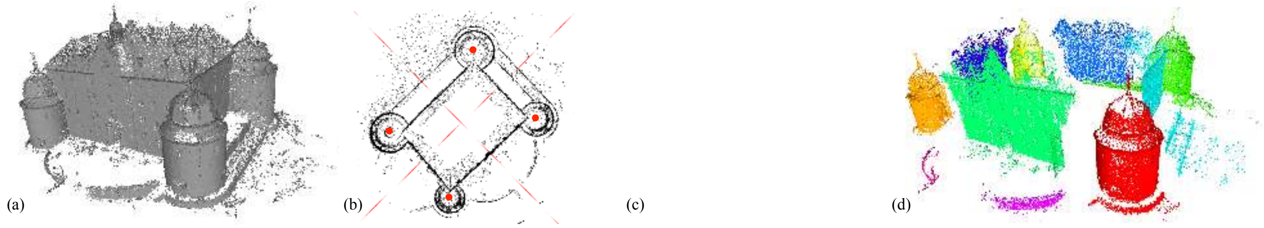

Figure 2: An overview of our method: (a) semi-dense point cloud as input, (b) identification of mirror symmetries and rotational symmetries (axes in red), (c) extraction of free-form polylines (red lines), (d) our segmentation result.

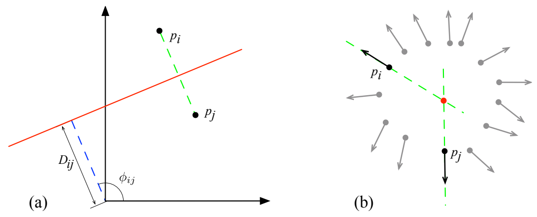

Figure 3: Our voting schemes for mirror (a) and rotational symmetries (b). (a) the points p_i and p_j vote for the direction of the red perpendicular bisector of the segment connecting the points, parametrized by its distance D_{ij} to the origin and its depicted angle ϕ_{ij}. (b) The points p_i and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

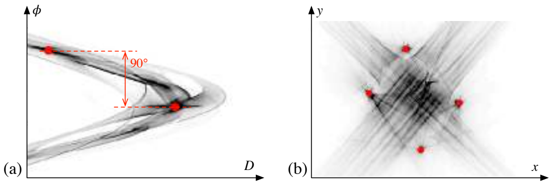

Figure 4: Hough-spaces for voting for (a) symmetry lines of mirror symmetries, (b) pivot points of rotational symmetries for the point cloud shown in Fig. 2. Red dots show the extracted peaks (see the text for details).

throughout the datasets in the experimental section, and recall that the detected symmetries enter as hypotheses into the final optimization in Section 3.3. An example of the corresponding Hough-space is shown in Fig. 4b.

3.2. Detecting Free-form Shapes

To be able 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 [13] (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 3 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 [11]. In a final cleaning step, short polylines are eliminated. The output is a set of polylines roughly approximating the shape of dense clusters in the 2D point cloud. An example is shown in Fig. 2c.

3.3. Structural Element Assignment

The major and 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, rotational symmetry, or free-form polyline discussed in Sections 3.1 and 3.2.

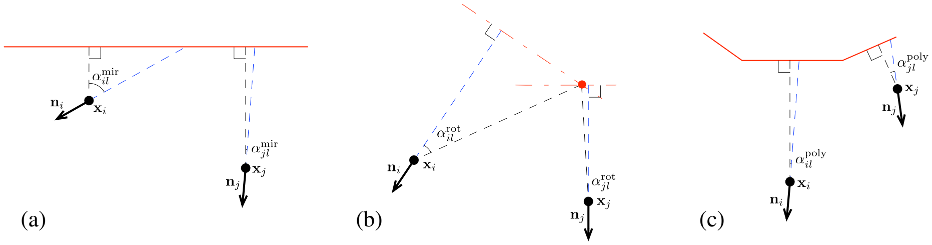

Figure 5: Definition of angle α_{il}^t between a point and a structural cue: (a) mirror symmetry (t = mir), (b) rotational symmetry (t = rot), (c) free-form polyline (t = poly).

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:

$$E(l_1, l_2, \ldots, l_N) = \sum_{i=1\ldots N} \Theta(p_i, l_i) + \gamma_1 \cdot \sum_i \sum_j \Psi(p_i, p_j, l_i, l_j) + \gamma_2 \cdot \sum_{\forall l \in \mathcal{L}} \Gamma(l), \quad (4)$$

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 structural cue generates structural elements.

Our unary cost for each of the structural cues is

$$\Theta(p_i, l_i) = 1 - W^t(l_i) \cdot K^t(\mathbf{x}_i, l_i) \cdot C^t(\mathbf{n}_i, l_i). \quad (5)$$

where index $t \in \{\text{mir}, \text{rot}, \text{poly}\}$ 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.

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

Our pairwise term in (4) is the weighted Potts penalty

$$\Psi(p_i, p_j, l_i, l_j) = \begin{cases} 0, & l_i = l_j \\ e^{-\lambda_E d_E(p_i, p_j)}, & l_i \neq l_j. \end{cases} \quad (6)$$

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 [53].

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:

$$\Gamma(l) = \begin{cases} 1, & \exists i : l_i = l \\ 0, & \text{otherwise}. \end{cases} \quad (7)$$

We solve the multi-label optimization efficiently via the α-expansion algorithm [8, 7, 23]. The parameters γ_1 and γ_2 balance the relative importance of the three major energy terms. Their choice is discussed in Section 4.

Next, we explain how each type of cue $t = \{\text{mir}, \text{rot}, \text{poly}\}$ is incorporated into the unary term (5) by defining each of its subterms $C^t(\mathbf{n}, l)$, $K^t(\mathbf{x}, l)$ and $W^t(l)$.

3.3.1 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 subterms of (5) for mirror symmetries (t = 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

$$C^{\text{mir}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{mir}}), \quad (8)$$

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 5a). Separation of points on the two sides of the symmetry axis is encoded into the definition of $K^{mir}(\mathbf{x}_i, l)$. For the structural element l^+ generated by the positive side of the axis,

$$K^{\text{mir}}(p_i, l^+) = \begin{cases} 1, & x_i \cos\phi + y_i \sin\phi + D_i \geq 0 \\ 0, & \text{otherwise} \end{cases} \tag{9}$$

where $\mathbf{x}_i = (x_i, y_i)$ is the position of point p_i and (D, ϕ) is the Hough-parametrization of the symmetry axis. For the structural element l^- on the negative side, we use $\leq$ in (9)

The weight $W^{\text{mir}}(l)$ in (5) is set to 1 for all structural elements generated by mirror symmetries (high importance).

3.3.2 Energy Terms for Rotational Symmetries

Each rotational symmetry generates a single structural element identified by a label $l \in \mathcal{L}$. Herein, we define the subterms of (5) for rotational symmetries (t = 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

$$C^{\text{rot}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{rot}}), \tag{10}$$

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 5b).

Unlike dominant mirror symmetries, which tend to be more global for a dataset, we aim to extract local rotational symmetries. The area of influence is determined by the points which vote for the rotational symmetry. Hence, we define the kernel as

$$K^{\text{rot}}(\mathbf{x}_i, l) = \exp\left\{-\frac{||\mathbf{x}_i - \mathbf{c}_l||^2}{2{\sigma_l}^2}\right\}, \tag{11}$$

where $\mathbf{x}_i$ is the 2D position of *any* point p_i in the dataset, $\mathbf{c}_l$ is the pivot/center of rotation and σ_l is the standard deviation of the distances between $\mathbf{c}_l$ and all *support* points.

As for mirror symmetries, the importance weight $W^{\text{rot}}(l)$ is set to 1 for all rotational symmetries (high importance).

3.3.3 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 subterms of (5) for t = 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

$$C^{\text{poly}}(\mathbf{n}_i, l) = 1 - \sin(\alpha_{il}^{\text{poly}}), \tag{12}$$

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 5c.

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

$$K^{\text{poly}}(\mathbf{x}_i, l) = 1 - \frac{1}{1 + \exp\{\lambda_p(\tau_p - ||\mathbf{x}_i - \hat{\mathbf{x}}_{il}||)\}}, \tag{13}$$

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 inflexion 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 (5) 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 and define polyline *usefulness*

$$U_l = \sum_k f_{lk} \cdot \min(\sum_{l'=1\ldots L} \Theta(m_{lk}, l'), 1), \tag{14}$$

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

$$W^{\text{poly}}(l) = \frac{1}{1 + \exp\{\lambda_w(t_w - U_l)\}}, \tag{15}$$

where we set an abrupt change $\lambda_w = 20$ similarly to (13), and t_w is set to 0.6 in all experiments.

4. Experiments

In this section we evaluate the properties of our 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. The method is tested on 6 noisy datasets. They have different architectural elements like towers, arches, ellipsoids, planar and free-form walls, as summarized in Table 1.

We evaluate our results quantitatively by comparing them with our ground truth segmentations. The segmentation accuracy is measured by the 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 [12]. We report the mean accuracy over all labels.

Dataset	# pts	3D Source	# ASE			Method	Segmentation accuracy [%]										
			mir	rot	poly		Per label, sorted by type of architectural structural element (ASE)										Mean
Orebro [38]	1.7M	PMVS [14]	2	4	2	Clustering	6.0	0	26.8	0	1.3	7.3	8.5	2.9	7.1	0.9	6.1
						Plane fitting	89.8	81.7	72.3	58.7	0	0	25.0	7.3	14.5	10.5	36.0
						Method [44]	75.2	78.0	81.0	77.9	68.5	63.6	70.9	53.0	39.7	59.4	66.7
						Our method	97.5	97.4	97.6	96.3	96.9	97.9	98.6	98.5	100.0	100.0	**98.1**
Arch [38]	300K	PMVS [14]	2	0	0	Clustering	35.5	13.1	3.6	5.5							14.4
						Plane fitting	76.4	67.3	67.5	44.2							63.8
						Method [44]	58.1	62.8	63.1	15.1							49.8
						Our method	94.6	86.6	94.4	51.3							**81.7**
Colosseum [53]	200K	SfM [53]	0	0	3	Clustering	20.8	2.9	33.9								19.2
						Plane fitting	17.9	7.7	29								18.2
						Method [44]	49.3	52.8	35.0								45.7
						Our method	99.9	78.6	98.8								**92.4**
Sant'Angelo [18]	60K	SfM [53]	0	0	3	Clustering	3.1	1.7	8.2	2.7	0.5	6.5	0.8	24.4			6.0
						Plane fitting	39.5	33.3	8.6	22.3	77.7	0	7.7	18.1			25.9
						Method [44]	91.1	58.6	33.7	68.2	83.2	97.3	97.5	99.6			78.7
						Our method	59.4	55.5	100	99.8	97.5	98.3	96.1	89.1			**87.0**
Trinity Chapel (failure case)	200K	SfM [53]	0	0	3	Clustering	27.6	4.9	0	1.6	5.2						7.86
						Plane fitting	24.7	55.9	24.3	26.0	9.5						28.1
						Method [44]	28.2	29.2	44.2	24.3	61.8						37.5
						Our method	63.0	86.7	0	75.5	0						**45.1**

Table 1: Datasets and comparison of our method with the baselines. Reported is segmentation accuracy (Jaccard index [%])

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 as 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.

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 to 5. Changing the value of γ_2 affects the number of detected labels yet is robust. Only very low values result in an over-segmentation.

The computational bottleneck is Hough voting that is determined by its quadratic complexity. For voting for symmetries, we downsample the datasets to maximally 350K points, computation time for which is 600K seconds.

4.2. Comparison to Baselines

In this section we evaluate our method against baseline approaches. Due to absence of directly comparable work, we compare to alternative ways of unsupervised decomposition of a point cloud.

The first baseline groups the points based on their geometric proximity. For that, we use Power Iteration Clustering [27] which propagates information about point similarity and then explicitly use k-means clustering to group the points. The input is a sparse distance matrix that contains the distances of every point to their 100 closest neighbors. Although k is generally unknown, we set it to the number of ground truth structural elements. 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. [44] that decomposes a point cloud into a set of primitives like planes, tori, spheres etc.

The results are provided in Fig. 6 and in Tab. 1. The clustering baseline does not arrive at a reasonable decomposition. The segmentation it provides separates the point clouds either over the edges where the point density is lower (curved areas of Orebro and Arch) or completely arbitrarily (Colosseum), resulting in low mean segmentation accuracy.

Plane fitting results in significantly better results. For suitable landmarks, such as Arch, it is able to capture the ASEs very well. 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 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 [44] is very efficient at fitting primitives into point clouds. However, it suffers from several drawbacks. First, in the absence of clear primitives it fails to fit a sensible model (see Colosseum, where each side is shared between several models). Our method is capable of dealing with these elements due to the free-form polylines cue. Second, the primitives are too simple for buildings. For example, in Orebro different parts of towers that have different width get different cylinders assigned to them.

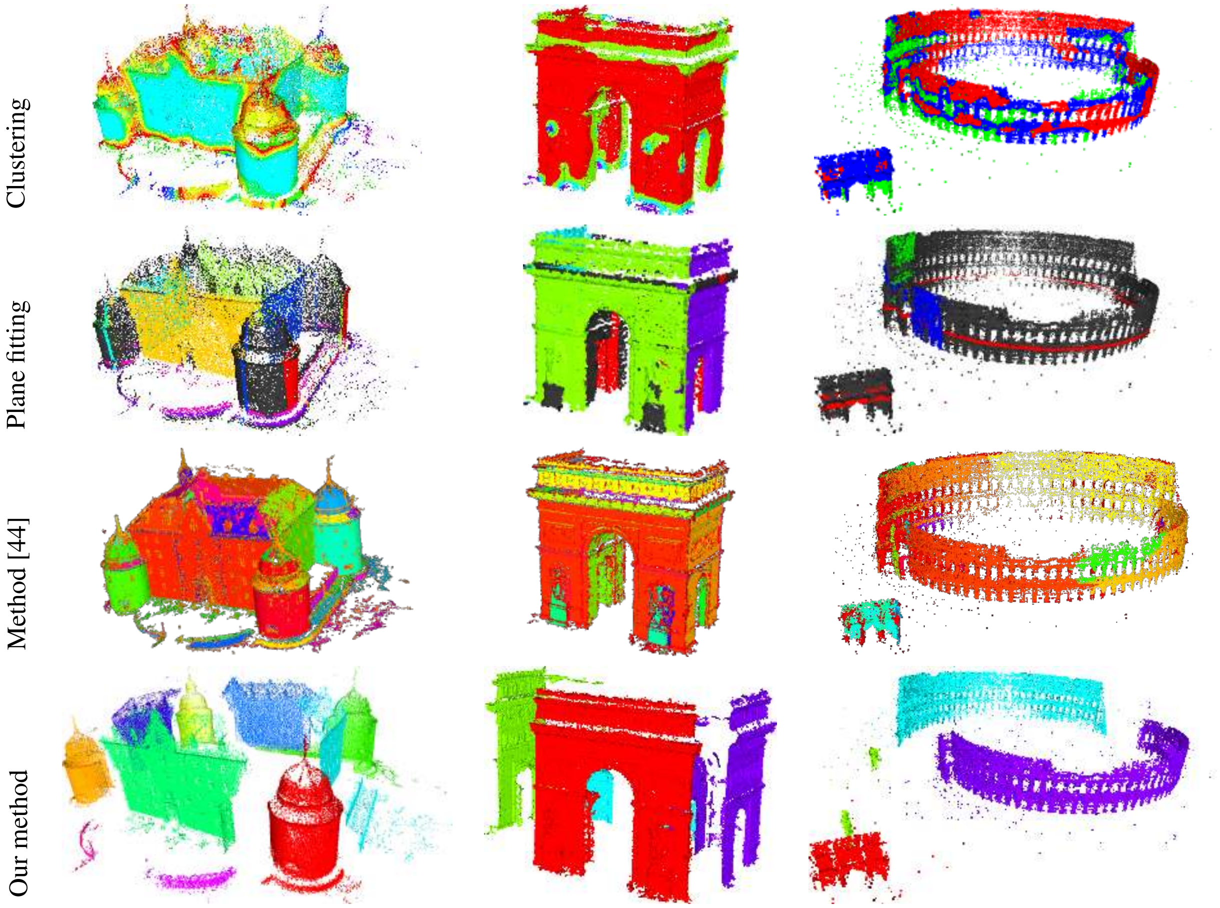

Figure 6: Our method on Orebro, Arch and Colosseum datasets. The baselines fail to capture all structural elements as they either contain no higher level features (clustering), enforce hard priors (plane fitting) or fit redundantly many primitives (method [44]). Our method successfully decomposes the complex scenes due to the global optimization.

Our method successfully and consistently outperforms the baselines as it is optimized for multiple specialized structural features rather than hard constraints or local clustering. In the quantitative evaluation shown in Tab. 1. The method 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. 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. The parts of the meadow are assigned to the closest ASEs (cyan).

Overall our method successfully decomposes all models into their ASEs. For models without any mirror or rotational symmetries it will revert to only using free-form polylines. Failure cases occur when the symmetries or free-form lines are not correctly detected. Missing data effects any of the methods as no direct 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 basic planar abstraction should suffice, we look at more complex architectural landmarks.

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

In future work we plan to iterate symmetry extraction and structural element assignment, infer long-distance graph connections to complete empty areas of the 3D reconstruction, and enlarge the list of detected symmetries.

References

[1] S. Agarwal, N. Snavely, I. Simon, S. Seitz, and R. Szeliski. Building rome in a day. In *ICCV*, 2009.

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

[3] Y. Bao, M. Chandraker, Y. Lin, and S. Savarese. Dense Object Reconstruction with Semantic Priors. In *CVPR*, 2013.

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

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

[6] A. Bódis-Szomorú, H. Riemenschneider, and L. Van Gool. Superpixel Meshes for Fast Edge-Preserving Surface Reconstruction. In *CVPR*, 2015.

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

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

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

[10] A. Dame, V. Prisacariu, C. Ren, and I. Reid. Dense Reconstruction Using 3D Object Shape Priors. In *CVPR*, 2013.

[11] D. Douglas and 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):112–122, 1973.

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

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

[14] Y. Furukawa, B. Curless, S. Seitz, and R. Szeliski. Towards Internet-scale Multi-view Stereos. In *CVPR*, 2010.

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

[16] S. Gould, J. Rodgers, D. Cohen, and G. E. ad D. Koller. Multi-class segmentation with relative location prior. *IJCV*, 80(3):300–316, 2008.

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

[18] Q. Hao, R. Cai, Z. Li, L. Zhang, Y. Pang, and F. Wu. 3D Visual Phrases for Landmark Recognition. In *CVPR*, 2012.

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

[20] P. Hough. Machine analysis of bubble chamber pictures. In *International Conference on High Energy Accelerators and Instrumentation*, 1959.

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

[22] N. Kobyshev, H. Riemenschneider, and L. Van Gool. Matching Features Correctly through Semantic Understanding. In *3DV*, 2014.

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

[24] K. Kser, C. Zach, and M. Pollefeys. Dense 3d reconstruction of symmetric scenes from a single image. In *DAGM*, 2011.

[25] F. Lafarge, R. Keriven, and M. Bredif. Combining meshes and geometric primitives for accurate and semantic modeling. In *BMVC*, 2009.

[26] F. Lafarge, R. Keriven, M. Bredif, and H. Vu. A hybrid multiview stereo algorithm for modeling urban scenes. *PAMI*, 35(1):5–17, 2013.

[27] F. Lin and W. Cohen. Power iteration clustering. In *ICML*, 2010.

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

[29] J. Liu, S. Ali, and M. Shah. Recognizing human actions using multiple features. In *CVPR*, 2008.

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

[31] A. Mansfield, N. Kobyshev, H. Riemenschneider, W. Chang, and L. Van Gool. Frankenhorse: Automatic Completion of Articulating Objects from Image-based Reconstruction. In *BMVC*, 2014.

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

[33] A. Martinovic, M. Mathias, J. Weissenberg, and L. Van Gool. A Three-Layered Approach to Facade Parsing. In *ECCV*, 2012.

[34] N. J. Mitra, L. J. Guibas, and M. Pauly. Partial and approximate symmetry detection for 3D geometry. In *SIGGRAPH*, 2006.

[35] N. J. Mitra, L. J. Guibas, and M. Pauly. Symmetrization. In *SIGGRAPH*, 2007.

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

[37] P. Müller, G. Zeng, P. Wonka, and L. Van Gool. Image-based procedural modeling of facades. In *SIGGRAPH*, 2007.

[38] C. Olsson and O. Enqvist. Stable structure from motion for unordered image collections. In *SCIA*, 2011.

[39] M. Pauly, N. J. Mitra, J. Wallner, H. Pottmann, and L. J. Guibas. Discovering Structural Regularity in 3D Geometry. *SIGGRAPH*, 2008.

[40] H. Riemenschneider, A. Bodis-Szomoru, J. Weissenberg, and L. Van Gool. Learning Where To Classify In Multi-View Semantic Segmentation. In *ECCV*, 2014.

[41] H. Riemenschneider, U. Krispel, W. Thaller, M. Donoser, S. Havemann, D. Fellner, and H. Bischof. Irregular lattices for complex shape grammar facade parsing. In *CVPR*, 2012.

[42] R. B. Rusu, N. Blodow, and M. Beetz. Fast Point Feature Histograms (FPFH) for 3D Registration. In *ICRA*, 2009.

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

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

[45] S. Sengupta, J. Valentin, J. Warrell, A. Shahrokni, and P. Torr. Mesh Based Semantic Modelling for Indoor and Outdoor Scenes. In *CVPR*, 2013.

[46] S. Sinha, D. Steedly., and R. Szeliski. Piecewise Planar Stereo for Image-based Rendering. In *ICCV*, 2009.

[47] O. Teboul, L. Simon, P. Koutsourakis, and N. Paragios. Segmentation of building facades using procedural shape prior. In *CVPR*, 2010.

[48] C. Vanegas, D. Aliaga, and B. Benes. Building reconstruction using manhattan-world grammars. In *CVPR*, 2010.

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

[50] J. Weissenberg, M. Gygli, H. Riemenschneider, and L. Van Gool. Navigation using Special Buildings as Signposts. In *MapInteract*, 2014.

[51] T. Werner and A. Zisserman. New techniques for automated architecture reconstruction from photographs. In *ECCV*, 2002.

[52] C. Wu. Towards linear-time incremental structure from motion. In *3DPVT*, 2013.

[53] C. Wu and S. Agarwal. Schematic surface reconstruction. In *CVPR*, 2012.

[54] L. Zebedin, J. Bauer, K. F. Karner, and H. Bischof. Fusion of Feature- and Area-Based Information for Urban Buildings Modeling from Aerial Imagery. In *ECCV*, 2008.

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