RAPTER: Rebuilding Man-made Scenes with Regular Arrangements of Planes

Aron Monszpart[1] Nicolas Mellado[1,2] Gabriel J. Brostow[1] Niloy J. Mitra[1]

[1]University College London, [2]Université de Toulouse; UPS; IRIT

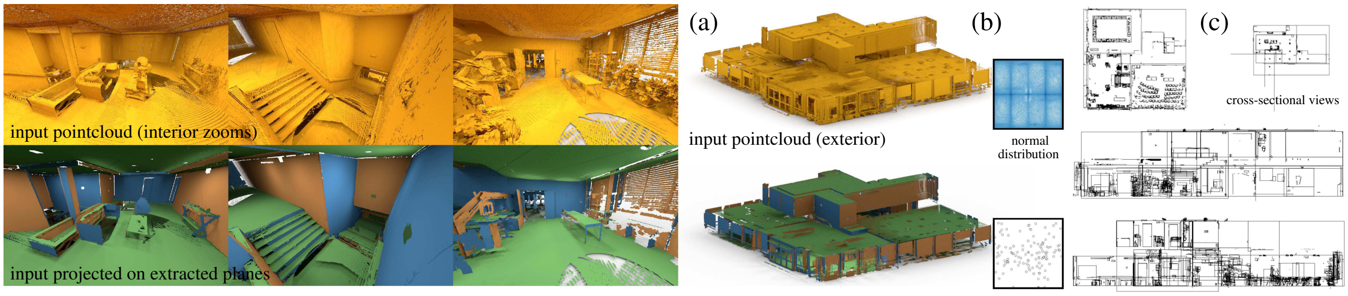
input pointcloud (interior zooms)
input projected on extracted planes
(a)
input pointcloud (exterior)
(b)
normal distribution
(c)
cross-sectional views

Figure 1: *We present a novel approach to extract Regular Arrangements of Planes (RAP) from an unstructured and noisy raw scan (shown in gold). (a) In this example, our algorithm reconstructs a building arrangement from a raw pointcloud, pre-assembled from multiple laser scans. (b) The distribution of the initial normals is very noisy, which makes any greedy arrangement of planes error-prone. Instead, we propose a global algorithm to simultaneously select both the planes along with their sparse inter-relations. (c) Cross-sectional views reveal the discovered regularity of the extracted arrangements at multiple scales, e.g., walls, stairways, chairs, etc. Parallel planes have same color.*

Abstract

With the proliferation of acquisition devices, gathering massive volumes of 3D data is now easy. Processing such large masses of pointclouds, however, remains a challenge. This is particularly a problem for raw scans with missing data, noise, and varying sampling density. In this work, we present a simple, scalable, yet powerful data reconstruction algorithm. We focus on reconstruction of man-made scenes as regular arrangements of planes (RAP), thereby selecting both local plane-based approximations along with their global inter-plane relations. We propose a novel selection formulation to directly balance between data fitting and the simplicity of the resulting arrangement of extracted planes. The main technical contribution is a formulation that allows less-dominant orientations to still retain their internal regularity, and not become overwhelmed and regularized by the dominant scene orientations. We evaluate our approach on a variety of complex 2D and 3D pointclouds, and demonstrate the advantages over existing alternative methods.

CR Categories: I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Surface fitting

Keywords: reconstruction, pointcloud, RANSAC, scene understanding, regular arrangement

Links: DL PDF WEB VIDEO DATA CODE

1 Introduction

Pointclouds are now easy to acquire. They can be recordings of both indoor and outdoor environments, and can easily contain millions of samples. Such data volumes often retain interesting information about the captured scenes, and hence are keenly investigated in the context of scene understanding. These analyses generate valuable scene priors for various computer graphics applications. For example, they can reveal typical object arrangements in scenes [Kim et al. 2012; Mattausch et al. 2014; Chen et al. 2014], provide non-local priors for scene completion [Zheng et al. 2010], deliver workspace affordance metrics [Sharf et al. 2013; Yan et al. 2014], or aid autonomous navigation by providing scene maps [Anand et al. 2011]. A grand goal is to abstract such massive data volumes [Nießner et al. 2013; Nießner et al. 2014] to eventually produce semantic understanding of the scenes [Boulch et al. 2014].

In the context of data analysis, one interpretation of abstraction [Yumer and Kara 2012; Lafarge and Alliez 2013; Oesau et al. 2014] is to reveal interesting high-level global scene characteristics, rather than focus on local details, for example obtained via a full surface reconstruction. In the context of man-made scenes, we observe that scene characteristics are often encoded in the form of inter-part relations, inside and across objects. As many objects primarily comprise of planar faces, man-made scenes can be well abstracted as collections of planes, more importantly, planes along with their inter-relations. In this paper, we focus on the problem of reconstructing raw pointclouds as regular arrangements of planes. The resultant abstractions provide compact and simplified representations, and expose high-level structures hidden in the raw data.

In the case of noisy, incomplete, outlier-ridden data, the challenge is to balance between compliance with data, and reliance on relations to regularize the extracted planes. However, one source of difficulty is that configurations of inter-plane relations are often scene-specific due to sensitivity to the primitives' local and global environments, and hence are not known *a priori*. They can easily be missed, or even worse, wrongly identified. The same set of 3D points can be scans of two parallel walls, or a wall and an ajar door depending on the context. Specifically, it can be particularly difficult to retain non-dominant plane orientations as they can easily be masked and falsely 'regularized'. E.g., Figure 1 shows a scan of an entire building consisting of few million points and having details ranging from exterior walls, interior features such as staircases, rooms with scattered chairs, *etc.,* thus exhibiting a range of different relations in various parts of the scene. The challenge is to detect these multiple interesting scene features in a unified setting.

The most popular pointcloud reconstruction approach is to progressively extract best fitting primitives using RANSAC, or its variants [Schnabel et al. 2007]. The method is attractive given its simplicity, scalability, and probabilistic guarantees. However, such a local and incremental analysis easily misses global scene-level structures (see Figure 8). Various refinements have been proposed including allowing users to explicitly annotate [Arikan et al. 2013] and greedily identify inter-primitive relations [Li et al. 2011a]. The challenge is to bootstrap the algorithms, which all heavily depend on the initial set of primitives. As an extreme case, the initial inter-primitive relations in Figure 1 being extremely noisy, it is very easy for a greedy approach to make erroneous early commitments, resulting in significant overall degradation (see Figure 7).

In this paper, we propose a global approach to *simultaneously* select a set of planes along with their relations. We make the key observation that global relations between planes persist over long distances in man-made scenes, counteracting some of the harm caused by noisy data. Hence, instead of extracting individual primitives, we extract sets of primitives, and in the process favor specific arrangements of primitives over less regular ones. We seek out regular arrangements with regularity being measured by aggregating agreements between primitive pairs in the extracted set. We enable the user to specify for a given task a certain family of sought relations, and our output allows the user to infer further relations. For example, in Figure 1, we extract a *regular arrangement of planes* (RAP) with mutually parallel/orthogonal primitive pairs (walls, chairs, staircase, etc.), even at the expense of the resultant data fitting error being marginally higher. By working directly in the space of relations, the algorithm provides non-local coupling and allows reliable planes extracted in less noisy regions to influence and regularize the corrupted regions of the scans.

The algorithm starts by producing an initial set of candidate primitives using local analysis. Then, we generate a larger set of candidate primitives based on potential inter-primitive relations. This step effectively hypothesizes possible relations and allows reliable primitives to create potentially good candidates in less reliable parts of the data. Finally, in a key selection step, we extract a regular arrangement of planes by balancing between explaining the pointcloud and producing a simple and compact arrangement of planes. The main insight is to defer the final selection to the end, and extract the simplest globally consistent arrangement of planes that best explains the raw input. Thus, expanding the candidate set eventually simplifies the problem. As an important technical contribution, we formulate the RAP extraction problem as a mixed-integer program that allows multiple orientation relations to coexist, even when significantly unbalanced in corresponding number of witnesses. For example, in Figure 1, the chairs are still abstracted as sets of extracted planes, and not unduly regularized by big exterior building walls, *i.e.,* the chairs are not 'snapped' to align with the walls.

We evaluated our algorithm on a range of 2D and 3D datasets and found the method to be robust under noise, sampling variations, and missing data. The extracted primitive arrangements directly provide an abstraction of the input and significantly out-performed specialized structure analysis alternatives. In summary, our main contributions are formulating the problem of pointcloud abstraction as an instance of coupled selection among candidate plane primitives; and proposing a simple, robust, scalable algorithm to extract such a globally coupled RAP directly from raw pointclouds.

2 Related Work

With the growth of acquisition devices, size and volumes of recorded pointclouds continue to evolve rapidly ([Nießner et al. 2013; Nießner et al. 2014]). In order to distill such vast amounts of raw data to usable knowledge, researchers have looked beyond surface reconstruction towards data abstraction and analysis. In this section, we focus on the works immediately related to our problem.

Scene understanding. With the growth of easy to use acquisition devices (e.g., Kinect®) scene understanding involving object segmentation and labeling of indoor scenes has received much attention in recent years. Various solutions have been proposed, both in supervised and unsupervised settings [Anand et al. 2011; Koppula et al. 2011; Silberman and Fergus 2011; Shao* and Monszpart* et al. 2014]. Kim et al. [2013] introduced Voxel-CRF to jointly refine 3D scene reconstructions from RGBD images by accurately segmenting our scene elements from the 3D reconstruction.

Man-made environments have dominant regularity and repeated features. Such symmetries and regularities manifest as redundancy in data, which can be exploited to denoise and consolidate measurements. The idea has been applied to man-made objects [Shen et al. 2012], office environments [Kim et al. 2012; Mattausch et al. 2014], and also for outdoor buildings [Zheng et al. 2010]. Boulch et al. [2014] proposed an interesting discrete optimization by expressing edges and corners as high-order selection potentials on voxel grids, to better abstract man-made environments like buildings. The approach, however, assumes dominant voxel directions similar to Manhattan frames, and is not easy to extend to scenes with multiple clusters of relations (*e.g.,* Fig. 8). The main challenge is to balance between data fit and scene regularity, while still allowing small characteristic directions to retain their identity. More recently, Chen et al. [2014] exploited scene context to recognize objects in an attempt to rapidly consolidate largescale interior RGBD images.

Urban reconstruction. Many man-made scenes explicitly conform to axes. Based on this observation, Gallup et al. [2007] presented a multi-view plane-sweep-based stereo algorithm to recover planar surfaces, potentially slanted, using a GPU-assisted real-time approach. They rely on finding a single ground plane, projecting points from upright objects onto that plane, and then finding the orientation that minimizes the entropy of those points onto an L-frame. Buildings and rooms typically come with such canonical reference frames. This observation is often exploited in the 'Manhattan assumption', wherein planes are fitted, but are restricted to particular frame directions. This approach was successfully demonstrated to improve dense, plane-based reconstructions on multiview stereo data [Furukawa et al. 2009].

Lafarge et al. [2013] use abstraction along with original point samples to obtain superior surface reconstruction, and more recently, [Oesau et al. 2014] propose a graph-cut based formulation for abstracting primitives in indoor scenes. Ramalingam et al. [2013] proposed an efficient method for extracting an arrangement of 3D lines from a single facade image using vanishing points and orthogonal structures via an interesting LP-based optimization. In another recent attempt, Straub et al. [2014] propose a probabilistic framework customized to describe scenes as a superposition of Manhattan frames (i.e., orthogonal frames) and use hybrid Gibbs sampling with gradient-based optimization to abstract urban scenes. To better cope with clutter that would induce large numbers of frames, they only consider frames supported by more than 15% of all normals, effectively narrowing the method to work best for scenes with six or fewer dominant frames. We aim specifically for such cluttered scenes, where all the points should be explained, and many relationships between primitives can be switched on or off, depending on the situation (e.g., modeling pentagons etc.).

Abstraction. In the context of 3D meshes, Mehra et al. [2009] proposed to abstract man-made objects by arrangement of feature curves. The work was later generalized to coanalyze collections of shapes to produce mutually consistent abstractions across shape

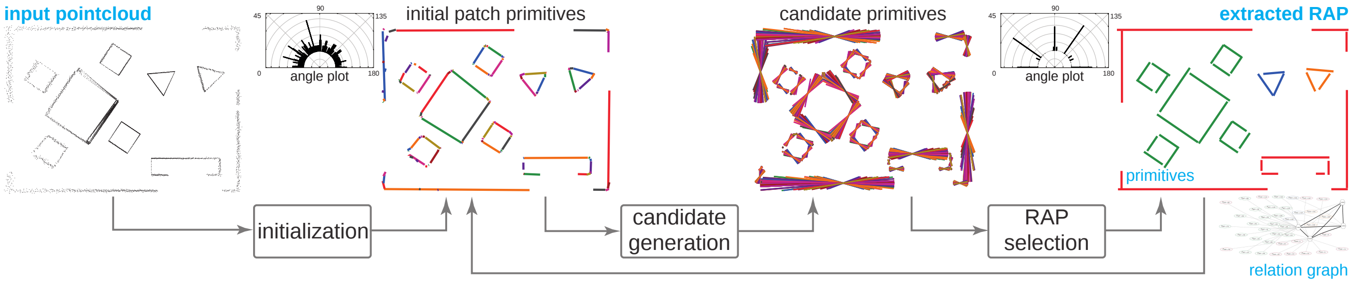

Figure 2: *Algorithm overview. We present* RAPTER *to abstract a raw pointcloud by extracting a globally-coupled regular set of planar primitives (line segments in 2D) along with the inter-primitive relations. The method works in three main stages: (i) initialization, (ii) candidate generation; and (iii) regular arrangements of planes (RAP) selection. As an outer loop, we repeat this following a coarse-to-fine strategy with primitives at bigger scales regularizing extraction in regions of low confidence at smaller scales. Insets show the distribution of angles among normals of pairs of oriented points using the initial and final line segments and point-to-line assignments. Our method discovers the regularity among the lines, and enables weakly supported orientations retain their independence. Input angles:* $\Theta = \{0, \pi/3, \pi/2, 2\pi/3\}$.

collections [Yumer and Kara 2012]. In the context of urban modeling, strategies involve integrating information from multiple acquisition modes (*e.g.,* LiDAR scans, see Zheng et al. [2010]), combining images and scans [Li et al. 2011b], or allowing user interaction to indicate symmetry structures from 3D data (*e.g.,* SmartBoxes [Nan et al. 2010]). Lin et al. [2013] proposed a hierarchy-tree based system to perform semantic decomposition and large-scale, progressive reconstruction of urban LiDAR scans. In a very interesting effort, Sharf et al. [2013] proposed mobility-trees to capture high-level functional information in man-made scenes. The extracted information encoded scene-level object information along with their part-level motion attributes. Very recently, a novel proactive scanning algorithm [Yan et al. 2014] was proposed to allow interaction with the scanned objects in order to acquire and combine recordings of visible and less easily visible object parts.

Local fitting. Many shape analysis algorithms attempt to cope with noisy input data by robustly fitting simple and approximate primitives to raw pointclouds, e.g., planes, cylinders, etc.. The most popular approach involves progressively fitting primitives using RANSAC [Schnabel et al. 2007] and its many variants (for example [Chum and Matas 2005; Ni et al. 2009]). While these methods are often robust to noise, recovering inter-primitive relations is not the goal of such local statistical methods (see Figure 8).

Constrained data fitting. As an alternative, the GlobFit framework [Li et al. 2011a] detects a set of relations from an initial set of primitives (obtained by RANSAC), and then performs data fitting in a constrained optimization setup. However, in such a two stage approach, if the initial set of extracted primitives is corrupted, erroneous relations are easily introduced, and worse 'conformed' to in the fitting stage. In another attempt, Lafarge et al. [2013] introduced point-set structuring for first consolidating raw inputs and resampling extracted planar components. They then propose a Delaunay triangulation based hybrid reconstruction to produce high-quality reconstructions. The method cleverly combines canonical parts from extracted (planar) primitives and free-form parts of the inferred shapes. Such methods, however, decouple primitive extraction from relation detection, which is what we are trying to avoid.

In more interactive contexts, Sinha et al. [2008] allowed users to interactively model buildings by annotating over images. Arikan et al. [2013] allowed the user to interactively specify relations and connections among the set of initial primitives extracted by RANSAC, with the system simultaneously fitting to the input point cloud and ensuring planarity of the polygons.

In a more general data analysis context, the PEARL setup [Isack and Boykov 2012] formulated model-fitting as a labeling problem using a global energy formulation. However, the global coupling involved a smoothness prior across neighboring primitives and hence, did not consider broader inter-primitive relations, which are dominant in man-made scenes (see Figure 8b). [Pham et al. 2014] extended this framework to pairwise non-spatial relations. To preserve diversity in already medium-scale scenes remains difficult due to the exponential explosion of conformity suggesting relations. Instead, RAPTER focuses on simultaneously extracting the primitives and non-local relations in a coupled, global and robust formulation.

3 Overview

Our goal is to convert a raw pointcloud from a scanned scene into *regular arrangements of planes* (RAP), where regularity refers to a prescribed list of favored inter-plane relations. As output, we produce arrangements of planes, with associations to their respective raw points. See Figure 2 for an overview.

We observe that man-made environments primarily consist of planar parts that are mutually related. Typical inter-plane relations include parallelism, coplanarity, orthogonality, symmetry, etc. Hence, we give preference to such regular arrangements, to reconstruct the input pointclouds (e.g., LiDAR and Kinect® scans, etc.). We measure goodness of fit as a balance between two factors: (i) *data cost:* the data-fit residual for approximating a set of points by its planar primitive, and critically, (ii) *irregularity cost:* the irregularity of the mutual arrangement of the output planes. A conventional *spatial* term ensures segmentation smoothness.

To achieve this, one simple approach is to first fit a set of planes to approximate the input data, then try to discover potential inter-plane relations, and finally, conform to the extracted relations using a constrained-fitting approach. This, however, assumes that the extracted relations are mutually consistent (*i.e.,* do not contradict each other). Alternatively, one can progressively build a set of consistent (potential) relations by selectively adding relations one at a time. Such a greedy selection strategy can easily result in a catastrophic failure by committing to an erroneous relation early on. Instead, we formulate a global optimization to *simultaneously* extract a set of primitives along with their relations. Our main observation is that instead of committing to any solution in the early stage, and hence introducing a bias, it is better to defer decisions to a later stage.

Actual inter-plane relations in the real-world are often concealed by flaws and bias in the acquisition stage. Hence, we first hypothesize possible relations and create additional primitive planes as speculative explanations for the raw points. In other words, we first *expand* the set of candidate primitives by adding hypothesized planes, and then repose the RAP extraction problem as a *selection* problem. Surprisingly, first significantly expanding the candidate set of planes and then formulating a constrained optimization to extract the RAP results in a simple, robust, and scalable algorithm.

The algorithm proceeds in three simple stages: First, starting from a point set $\mathcal{S}$, as *initialization* we generate a set of primitive planes $\mathcal{P} := \{P_i\}$ by locally fitting planes to the input. Then, in the *candidate generation* stage, each pair of planar primitives creates additional candidate planes that are added to the primitive set $\mathcal{P}$ to form the enriched set $\tilde{\mathcal{P}}$. Finally, in the key *RAP selection* stage, we formulate an energy minimization to select the RAP from the enriched set $\tilde{\mathcal{P}}$ as the final abstraction of the input data $\mathcal{S}$.

This results in the simultaneous extraction of primitives along with their inter-relations. We run the algorithm over coarse-to-fine scales, allowing the primitive arrangements selected in the coarse scales to influence and regularize the solution at finer scales. We describe the core algorithm next.

4 Algorithm

Starting from a pointcloud $\mathcal{S}$ of a man-made scene, our goal is to abstract the scene as regular arrangements of planes. As part of the input, we expect to know a priori whether certain inter-plane relations, such as 45°, contribute to a user-specified definition of "regular". The proposed algorithm runs in three main stages: (i) initialization, (ii) candidate generation, and (iii) RAP selection. RAP selection optimizes the balance between explaining $\mathcal{S}$, and imposing inter-plane regularity. We now describe these steps of the algorithm, and give more specific implementation details.

4.1 Initialization

We oversegment the point set $\mathcal{S}$ through simple region growing. The aim is to group nearby points, with consistent normals into patches. If needed, $\mathcal{S}$ is first turned into a set of oriented points by using local PCA analysis. The complete oversegmentation of $\mathcal{S}$ into a set of patches $\{S_i\}$ proceeds from the bottom up. Each point j within a Euclidean distance ρ of point i is grouped into patch S_i if its orientation differs from S_i's by less than τ. This process repeats, expanding the search-area to within ρ of any new point in the group. At each iteration, we compute the least-squares fit of a "local" plane P_i. These planes have finite extent, clipped to a bounding box based on the projection of points in S_i. We thus obtain our initial set of candidate primitive planes $\mathcal{P} := \{P_i\}$, see Figure 3a.

4.2 Candidate generation

The planar patches detected in the initialization stage are based on local fitting, and hence can easily be biased by noise and outliers (see Figure 3a). Robustly detecting relations and regularity among such noisy shape fragments is difficult, especially since bias may occur gradually over long distances. Instead, we explicitly generate additional planes as speculative "cousins", arranged relative to the initial set, according to the expected inter-plane relations. Most of these speculative patches will ultimately be rejected, but they help to recover some planes that were undersampled or noisy. Together with the initial planes, this expanded set of candidates allows us to pose the search for a good global RAP as a selection problem.

To be specific, each plane P_j in the initial set $\mathcal{P}$ was created to explain one patch of oriented points S_j. However, S_j may be better explained, in a global sense, by an alternative plane that comes from, *e.g.,* rotating plane P_i by 90° and translating it to S_j. We denote such a new plane as $P_{i\to j}$, and it serves as a candidate alternative to the initial $P_j \equiv P_{j\to j}$. Thus, each patch S_j is now associated with a set of candidate primitives $\{P_{1\to j}, P_{2\to j}, ..\}$, where $P_{j\to j}$ is an alias for P_j. We re-estimate the finite extents of the other clones $P_{i\to j}$ based on the points in S_j. In practice, the number of possible inter-plane relations (i.e., $\Theta = \{0, \pi/2\}$) further multiplies the number of ways each P_i can be rotated onto S_j, but we keep the one with the best fit. The whole purpose of the candidate generation step is to ensure that the sought RAP configuration *exists* as a subset of the generated enriched candidate set $\tilde{\mathcal{P}} := \{P_{i\to j}\}$.

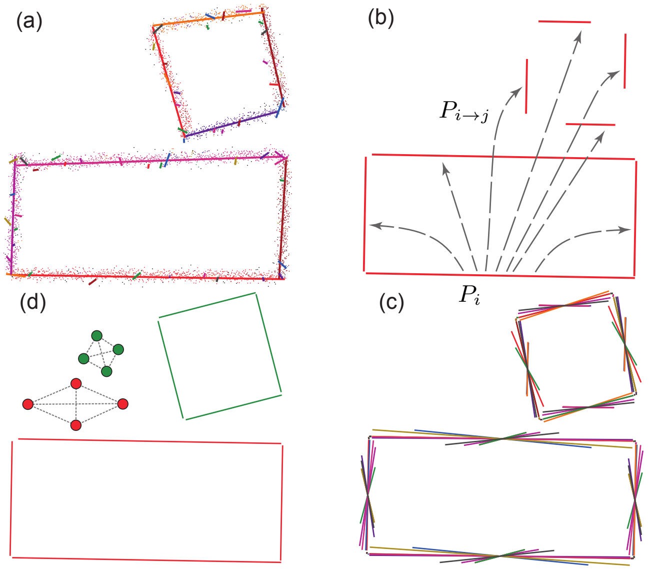

Figure 3: *We oversegment an input pointcloud into (a) point patches S_i along with their planar approximations P_i. (b) Each such primitive generates candidate primitives at the centroid of other patches, i.e., P_i generates $P_{i\to j}$ at the centroid of any other primitive P_j, creating (c) a set of enriched candidates $\tilde{\mathcal{P}}$. (d) Finally, via a selection formulation, we extract a set of regularly arranged planes (RAP), i.e., a set of planes along with their relation graph (shown inset) by balancing between global regularity and faithfulness to the local data. Input angles: $\theta = \{0, \pi/2\}$.*

As an example, we visualize all the candidate subsets in Figure 3c, where the sides of the rectangles have proposed new candidates for all the patches in both rectangles. Figure 3b shows the generated candidates $\{P_{i\to j}, j = 1, 2, \dots\}$ from just one plane P_i. Details of how a user-selected family of plane relations is used to generate the enriched candidate set in a scalable way, are given in Section 5.

4.3 RAP selection

We are now ready to formulate the RAP extraction problem as a subset selection problem from the enriched candidate set $\tilde{\mathcal{P}}$. We represent the selection of any one candidate $P_{i\to j} \in \tilde{\mathcal{P}}$ by a corresponding binary indicator variable $\chi_{i\to j} \in \{0, 1\}$. The binary vector $[\chi_{i\to j}, \dots]$ corresponds to a selection of candidate planes, a RAP.

Formulation. We pose the RAP extraction problem as an energy minimization using these binary selection variables. We want to balance simultaneously explaining the scene in a data-faithful way, and selecting an as-regular-as-possible arrangement of planes.
We encode this as a weighted combination of three terms:

$$\{\chi_{i\to j}\} = \underset{\{\chi_{i\to j}\}}{\operatorname{argmin}}\, E := \lambda\, E_{data} + (1-\lambda)\, E_{irr} + E_{spat}, \quad (1)$$

where $\lambda \in [0,1]$. To ensure that we assign each patch S_j at least one primitive candidate $P_{i\to j}$ to explain its data points, we require $\sum_i \chi_{i\to j} \geq 1 \ \ \forall j$. For example, with $\lambda = 1$, all $\chi_{j\to j} = 1,\ j = 1, 2, \ldots$, and the rest of the variables are 0, *i.e.,* only the original locally fit candidate planes get selected.

Data cost. Potentially, many planar candidates can be indicated as trying to explain each patch. We compute the total data fitting error as the sum of the individual data fitting residuals. With $E_d(P_{i\to j}, S_j)$ denoting the residual cost of abstracting patch S_j by $P_{i\to j}$, hence $E_{data} := \sum_j \sum_i \chi_{i\to j} E_d(P_{i\to j}, S_j)$, see Eq. (6).

Irregularity cost. Even a bad choice of arrangements of planes may satisfy the constraints and have a low data cost. To also encourage *regularity* in the arrangement of planes, it seems natural to construct an undirected irregularity measure $Irr(\cdot,\cdot)$ between every pair of planes present in the proposed RAP. Similarly to Pham et al. [2014], we formulated this as $E_{irr} := \sum_{j,i,l,k} \chi_{i\to j}\chi_{k\to l} Irr(P_{i\to j}, P_{k\to l})$, measuring the irregularity of any selected arrangement encoded by the indicator variables. By construction, certain pairs are perfectly compatible, for example, $Irr(P_{i\to j}, P_{i\to i}) = 0$, since both are generated from P_i using an a priori known inter-plane relation, hence these are automatically favored for selection.

Figure 3d shows an example of the selected RAP that prefer a perfectly regular arrangement, even at the expense of a slightly higher data cost ($\lambda = 0.5$ in this example). This is a desired behavior in such a balanced setting. Once regularized, a RAP can be visualized as a simple graph, with selected planes $P_{i\to j}$ as nodes. The nonzero indicator variables encode which inter-plane relations are part of the regularized RAP, as shown by the graphs in Figure 3d.

The scene in Figure 5a has its RAP depicted in Figure 5c, where the irregularity term has generated such pairwise potentials, that all 12 planes were interconnected perfectly. As a consequence, we see in Figure 5a that the data of the rotated square (data potentials not pictured) has been overruled by the axis-aligned majority. This is a common problem in scenes with large relative differences in the number of witnesses for the underlying orientations. The construed normalization variable in [Pham et al. 2014] is an attempt to resolve this problem. However, even moderately complex scenes like Fig. 2 require more control and respect for diversity than this allows for.

4.4 Irregularity revisited

In Figure 5a, we have just seen that the naive irregularity term E_{irr} can be too aggressive in encouraging *all* planes to be related to each other. Similar unaligned objects in a Manhattan-world drove the work of Straub et al. [2014], which translates to RAPTER with input angles $\Theta = \{0, \pi/2\}$. We refer to a set of orientations that, given the input relations, are internally perfectly regular as a *frame*.

Here, we redesign the E_{irr} cost to tolerate diversity. Obviously, the clusters of planes or objects that should be regularized separately are not known in advance. So we introduce a second level of indicator variables. For each internally-regular coordinate frame, we add a binary auxiliary variable in the optimization $\chi_i \in [0,1]$.

Tolerable irregularities. The irregularity cost is intended to encourage abstractions that cope with low quality data and to align even distant patches. The regularization can, however, result in undesired simplification in areas with high data fidelity. All cross-cluster plane pairs contribute to E_{irr}, so a sole rotated table in a room is literally punished from all sides. If we fail to distinguish between irregular arrangements vs. clusters of regular arrangements, we end up penalizing both. Instead, we desire a reconstruction like Figure 5b. The new auxiliary variables $\{\chi_i\}$ signpost groups of planes that are regular among themselves, *i.e.,* where one χ_i represents its own frame. Note that even distant objects may be part of the same frame. We reformulate the irregularity energy so that the optimized RAP for Figure 5a looks like Figure 5b. With the new irregularity, we observe that the energy of each frame encourages mutual regularity internally, but different frames only pay the price for being misaligned once; see the bold edge connecting the indicator variables' hexagonal "metanodes." We redesign the irregularity term, so

$$E_{irr^\star} := \sum_{i,k} \chi_i \chi_k Irr(P_{i\to i}, P_{k\to k}). \tag{2}$$

The original indicator variables $\chi_{i\to j}$ reveal which planes are selected as frames, so $\chi_i = \max_j \chi_{i\to j}$. Essentially, in the new irregularity measure $E_{irr^\star}$, the cost for choosing a new frame becomes less dependent on the number of similarly oriented primitives in the solution. If any of the candidate planes created from an initial locally-fitted plane in the candidate generation step is chosen, the corresponding auxiliary variable should get activated. We encode this behavior with the following constraints.

Constraints. An auxiliary node χ_i representing frame i contributes to the irregularity cost if any of the candidates $\{P_{i\to j}\}$ generated by the initial plane primitive $P_{i\to i}$ is selected. For each initial (plane) orientation i, we encode the max-condition as a single quadratic constraint of the form

$$\sum_j (\chi_{i\to j}\chi_i - \chi_{i\to j}) \geq 0 \ \ \forall i. \tag{3}$$

Thus, we arrive at the final updated formulation as

$$\begin{aligned}&\{\chi_i\}, \{\chi_{i\to j}\} = \\ &\underset{\{\chi_i\},\{\chi_{i\to j}\}}{\operatorname{argmin}} \ E := \lambda\, E_{data} + (1-\lambda)\, E_{irr^\star} + E_{spat},\end{aligned} \tag{4}$$

where $\lambda \in [0,1]$, subject to the constraints $\sum_i \chi_{i\to j} \geq 1\ \forall j$, and Equation 3. Note that the minimization produces the RAP abstraction, *i.e.,* both the selected set of planes and their inter-relations.

Spatial smoothness. Our main contribution comes from the global regularity of frames, where spatially distant objects are able to influence the regularity of the approximation. Since we start from an over-segmentation, we use a spatial smoothness term to encourage neighboring patches to pick the same plane orientation. We increment the energy with a fixed penalty for non-smoothness,

$$E_{spat} := \sum_{j,i,l,k;\ j\neq l} \chi_{i\to j}\chi_{k\to l} neigh(S_j, S_l) C_{spat}, \tag{5}$$

where the fixed spatial penalty $C_{spat} = (1-\lambda)/10$ is related to the irregularity weight. See Section 5 for estimation of patch proximity.

We need no additional term to limit the complexity of the arrangement (*e.g.,* number of selected planes $\sum \chi_{i\to j}$) as superfluous candidates get penalized by the total energy cost E. We proceed with details of the implementation and optimization.

5 Implementation Details

In this section, we provide various implementation details, comment on the optimization, and discuss necessary modifications to transition to 3D and for handling very large scenes. Although we detail the specific functions used, they could likely be replaced by comparable functions or cost measures leading to similar performance.

Initialization. For the initial oversegmentation of the input pointcloud, we progressively gather points if they are neighboring and

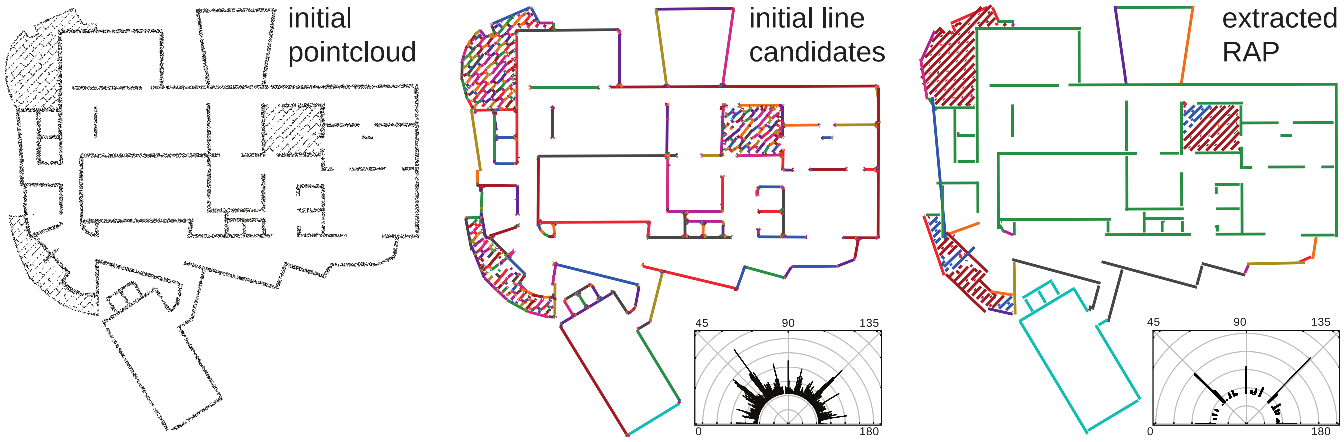

Figure 4: *Starting from a pointcloud obtained by digitizing a floorplan our algorithm takes the initial line candidates to extract regular arrangements. Corresponding angular plots show the normal distribution of output points. Regularity is masked in the noisy initial plot (spread distribution), but is visible in the final extracted RAP (clustered distribution). Final line segments are colored based on their corresponding orientation frames, same color denotes lines mutually parallel or orthogonal (compare to Figure 8). Input angles $\theta = \{0, \pi/2\}$.*

have comparable orientations. We use the approach proposed in Yan et al. [2014]. As angle threshold we use $\tau = \pm 15^\circ$ and a point distance of ρ. Apart from the list of possible plane relations, ρ is the main user-selected parameter of the system, and represents measurement-tolerance. Each new scene is scaled to fit the unit cube. Then ρ is set so that structures with size above this threshold will be preserved, whilst other, smaller variations will be simplified. The resolution of the RAP can thus be modified according to the scene characteristics and application context.

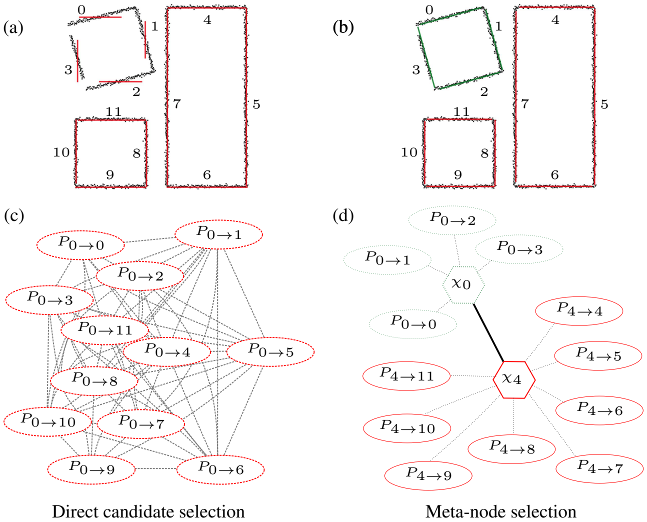

Figure 5: *The direct pairwise formulation leads to an undesired over-regularization (a) as non-dominant orientation frames easily get masked by dominant ones. A single frame inspired by $P_{0\to 0}$ gets selected (c), because the number of corresponding pairwise costs for across-frame connections relate exponentially to clique sizes. Our irregularity term (Section 4.4) can preserve distinct frames that represent only a minority of the points (b) in the pointcloud by introducing intermediate metanodes (hexagons) as shown in (d). Dashed lines and colors denote shared frames, the solid line denotes non-zero irregularity costs. Input angles $\Theta = \{0, \pi/2\}$.*

Data term. We measure the data cost in terms of the total residual error of abstracting all patch S_j's points $\{p_h\}$ by the primitive plane $P_{i\to j}$, as shown in Equation (6). Each candidate plane has a fixed cost determined by its assignment to a patch. Our plane primitives are finite, its bounding box encloses all points p_h within ρ distance from the infinite plane. This means that when the point p_h, projected onto the plane, falls inside the patch's bounding box, the distance $d()$ is the typical point to infinite-plane distance. When p_h falls outside the bounding box, $d()$ is the Euclidean distance to the nearest point inside. Therefore, our data cost is

$$E_{data} := \sum_j \sum_i \chi_{i\to j} E_d(P_{i\to j}, S_j) \quad \text{where,} \tag{6}$$

$$E_d(P_{i\to j}, S_j) = \frac{1}{|\{p_h \in S_j\}|} \sum_{p_h \in S_j} d\left(p_h, P_{i\to j}\right)^2 . \tag{7}$$

We normalize the per-patch residual error because patches have different numbers of points p_h. Point to finite plane distance provides robustness towards the smoothing effect of this normalization.

Irregularity. The most basic, yet surprisingly common regularity in an arrangement of planes is parallelism. We define irregularity as a function of the unsigned angle between the normals of two planes $Irr(P_i, P_j) := f(\angle(\mathbf{n}_i, \mathbf{n}_j))$, with $f()$ defined below. Regular arrangements contain planes with low irregularity, *i.e.,* the angle between their normals is close to 0. When the angle is 0, we think of this as a *perfect* relation. By construction, generated candidates are perfectly related with their generators.

Candidate generation. For each pair of planes $P_{i\to i}$ and $P_{j\to j}$, we generate two new planes. $P_{i\to j}$ comes from translating $P_{i\to i}$ to the centroid of S_j and rotating by one of the allowed relations (see Possible relations below), and similarly for $P_{j\to i}$ from $P_{j\to j}$. The coarse-to-fine iterations (see Scalability and Algorithm 1) mean that the output of a selection step serves as input for successive generation steps, *i.e.,* $P_{i\to j}$ can spawn $P_{i\to k}$ at patch S_k.

Pairwise term. Our main goal is to discover perfect relationships in the form of extracted RAP. In our implementation, we used $f(x) := 1 - \exp(-\delta x)$ with $\delta = 3$, to ensure that perfect regularity is encouraged, while still respecting minority frames.

Spatial smoothness. We use a spatial smoothness term to encourage neighboring patches to pick the same orientations allowing coplanar patches to be identified. Specifically, we check if the minimum distance between a pair of patches is small by estimating the minimum distance between their points, and if they have comparable orientation up to τ (angle tolerance), *e.g.,*

$$neigh(S_j, S_l) := \mathbf{1}\left(\min_{\mathbf{p}_g \in S_j; \mathbf{p}_h \in S_l} \|\mathbf{p}_g - \mathbf{p}_h\| < 2\rho\right) \cdot \mathbf{1}\left(\angle(\mathbf{n}(S_j), \mathbf{n}(S_l)) < \tau\right). \quad (8)$$

Quadratic programming. The RAP extraction problem amounts to a quadratic optimization problem with quadratic constraints. Note that the quadratic constraints are critical, as explained in Section 4.4 and Figure 5. We used a mixed integer nonlinear program solver [Bonami et al. 2008] that relies internally on a modified (for nonlinear problems) branch-and-cut algorithm, and an interior-point based LP solver [Wächter and Biegler 2006] for the relaxed sub-problems. We provided the solver with analytically computed second-order derivatives for the objective and constraint matrices. Note that the formulated constraints (and objectives) are non-convex, so the retrieved solutions are not guaranteed to be optimal. However, the quality of our output relies crucially on *robustly* retrieving a RAP, and we have not had problems with convergence in practice. Typically, for smaller problems, we ended up with 500 variables, while for larger problems, we had many more variables that we tackled by splitting the problem into groups of approximately 2300 variables, see Algorithm 1. We also used [Rusu and Cousins 2011; Schenk and Gärtner 2004] in our implementation.

Possible relations. Supporting other relations in RAPTER is relatively easy. In short, for any such relation one has to appropriately adapt the definition of regularity (irregularity in our case), and adjust the candidate generation step. In our implementation, the system supports planes that are

(i) Parallel: These are the default relation, require translation, and no further modifications to the energy terms.

(ii) Orthogonal: For irregularity, we simply measure the difference of angles between the planes from target angle $\pi/2$. For candidate generation in 2D, we translate the planes to their target location (as before), and rotate by $\pi/2$. In 3D, it is a bit more involved. Given two planes P_i and P_j with respective normals $\mathbf{n}_i$ and $\mathbf{n}_j$, we create $\mathbf{n}_{ij} = \mathbf{n}_i \times \mathbf{n}_j$. Then, we set the direction of $P_{i\to j}$ as $\mathbf{n}_i \times \mathbf{n}_{ij}$, and $P_{j\to i}$ as $\mathbf{n}_j \times \mathbf{n}_{ij}$. The rest of the setup is unchanged.

(iii) Other angles: For other generator angles Θ^*, we proceed just as above, but use multiples of Θ^*, instead of $\pi/2$. As a result, we get the set of input angles $\Theta = \{k\Theta^*\}$, $k = 1, 2, \ldots$; *s.t.* $k\Theta^* < \pi$. We adapt Irr to be $Irr(P_i, P_j) := \min_k \ f(|\Theta_k - \angle(\mathbf{n}_i, \mathbf{n}_j)|)$.

(iv) Coplanar: To model coplanarity, for any two planes P_i, P_j with the same (unsigned) orientation $\mathbf{n}_i = \mathbf{n}_j$, we set $Irr(P_i, P_j) := f(d(P_i, P_j))$ based on the offset distance $d()$ between the planes. Planes are simply mutually copied across in the candidate generation stage. In practice, other relations are explored together to optimize a RAP, that then seeds a coplanarity-only RAP, where we simplify (*i.e.,* merge) adjoining coplanar planes.

Scalability. For better efficiency, we make following modifications: (i) In the candidate generation stage, we restrict candidate generation to pairs of planes with low to medium irregularity. Essentially, we put a threshold on the angle (or offset difference) to reduce the number of generated candidates. The same parameters are used, as for the initialization ($\tau = \pm 15°$, see Table 1 for ρ).
(ii) We propose a coarse-to-fine RAP extraction workflow. First, at a coarse level, we only consider the larger initial planes (based on their area). Once we extract a corresponding RAP, we freeze these relations, *i.e.,* they are not allowed to change further. Then, we bring in the next level of patches, proceed as before, but allow the earlier RAP to also contribute in the candidate generation step and during selection. Note that we keep the older frame metanodes in place for the new patches. Essentially, relations detected at the coarse level can influence the ones lower down. However, the lower levels have no influence on the higher (*i.e.,* coarser) levels. This has proven to be effective because larger planes are more well-sampled. We go down in scale by factors of 2 as detailed in Algorithm 1. In line 16, we show one strategy to control the number of variables in the quadratic program by gradually introducing generated candidates. A strategy, that directly controls the number of metanodes (~quadratic terms) might be even more useful in practice.

Algorithm 1: RAP extraction by RAPTER

Input: Oriented points grouped into patches $\{S_j\} \in \mathcal{S}$, Local fits $\{P_{j\to j} \in \mathcal{P}\}$, relation generators Θ^*
Output: RAP $\mathcal{P}^* = \{P_{i\to j}\}$, Regular relations $\{\langle P_{i\to j}, P_{i\to k}, \textit{rels}\rangle\}$, Point assignments $\{p_h \to P_{i\to j}\}$

```
   /// (1) Initialization
1  $\mathcal{P}^0 := \{P_{j\to j}\} \in \mathcal{P}$ // Sort initial fits by ↓ area
2  $\mathcal{P}^* := \emptyset$ // Initialize set of selected candidates
3  $\theta := \frac{area_{max}}{area_{min}}\rho$ // Estimate area threshold

4  while $\mathcal{P}^0 \neq \emptyset$ do
       /// (2) Candidate generation
       // Select sufficiently large initial fits
5      $\mathcal{P}_\theta := \{P_{j\to j}, \textit{s.t.}\ \forall_j\ area(P_{j\to j}) > \theta\} \in \mathcal{P}^0$
6      $\mathcal{P}^0 := \mathcal{P}^0 \setminus \mathcal{P}_\theta$
       // Enrich from coarser and same scale
7      $\tilde{\mathcal{P}} := \mathcal{P}_\theta \cup$ Enrich($\mathcal{P}_\theta$, $\mathcal{P}^*$) $\cup$ Enrich($\mathcal{P}_\theta$, $\mathcal{P}_\theta$)
       /// (3) Selection from enriched set
8      $\mathcal{P}^* := \mathcal{P}^* \cup \{\mathcal{P}^{*'} \subseteq \tilde{\mathcal{P}}\}$ // Minimize Equation (4)
9      Simplify nearby co-planar patches
       /// (4) Iterate
10     if No break in Enrich then  decrease area threshold $\theta := \frac{\theta}{2}$
11 return $\mathcal{P}^*$

12 Function Enrich($\mathcal{P}$, $\mathcal{P}_{fixed}$,$|\tilde{\mathcal{P}}|_{max} = 2300$)
13 for  Each pair $\langle P_{j\to j} \in \mathcal{P},\ P_{k\to l} \in \mathcal{P}_{fixed}\rangle$ do
14     if  $\min_r(|\Theta_r - \angle(\mathbf{n}_j, \mathbf{n}_k)|) < \tau$ // Min angle
15      then  $\tilde{\mathcal{P}} := \tilde{\mathcal{P}} \cup \{P_{k\to j}\}$      // Create new candidate
16     if $|\tilde{\mathcal{P}}| > |\tilde{\mathcal{P}}|_{max}$ then  break       // Limit #variables
17 return $\tilde{\mathcal{P}}$ // End of function Enrich ()
```

6 Evaluations

Datasets. We performed our experiments on various 2D and 3D input scenes. We synthetically generated the input of Figure 2 and the Blensor ([Gschwandtner 2013]) laser scan "L-house". "Stairs" (see Figure 7) was acquired using Kinect®. "Nola" (LiDAR) is from [Zheng et al. 2010], "Empire" and "Lans" are courtesy of [Lafarge and Alliez 2013], and "Euler" is from [Oesau et al. 2014]. "Euler-Cut" is a section of the larger scene. Refer to Table 1 for scene sizes, and supplementary material for high-resolution figures.

6.1 Comparisons

We compared our results to an advanced, RANSAC-based method [Schnabel et al. 2007], constrained data fitting GlobFit [Li et al. 2011a], discrete labeling based PEARL [Isack and Boykov 2012], and point set structuring [Lafarge and Alliez 2013].

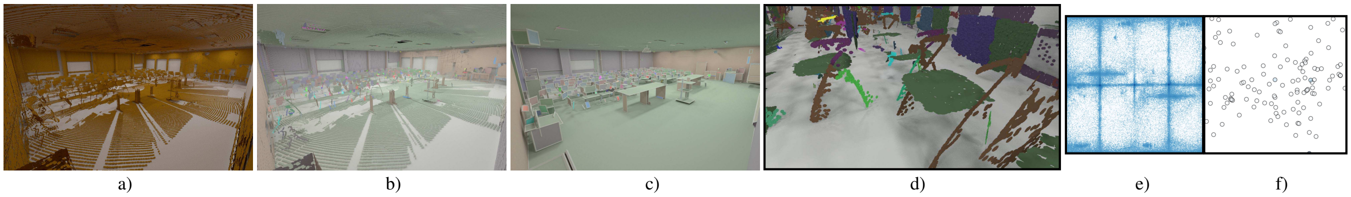

Figure 6: *Starting from acquired pointclouds (a), our algorithm extracts Regular Arrangements of Planes (RAP). The extracted planes are used to reproject the associated points (b), or approximate the input by a set of planar polygons (c) (zoom shown in (d)). The normal distributions (Cartesian projection) of the input cloud (e) are very noisy. In contrast, the extracted RAP have clean normal distributions showing the extracted regularity (f). Circles denote normal locations and their support is indicated by the color of the circles, going from white (few samples) to blue (many samples). Parallel planes are shown with the same color (figures (b)-(d)), and input angles were* $\Theta = \{0, \pi/2\}$.

Scene	#points	#initial	#prims	$\#\chi_i$	#rels	ρ
L-house	5k	67	13	2	61	0.02
Stairs	261k	300	99	2	8281	0.01
Nola	740k	15.8k	847	1	358k	0.004
Empire	1.2M	4.7k	163	17	7025	0.0025
Lans	1.3M	7.4k	490	9	22.1k	0.005
Euler-Cut	586k	4k	965	17	103k	0.004
Euler-Full	3.9M	28.7k	548	5	49.5k	0.002

Table 1: *Statistics of processed scenes (#points, #initial), and of the RAP retrieved by our method: the number of primitives #prims, representative orientations* $\#\chi_i$*, and perfect relations discovered amongst primitives in the orientation frames #rels.* ρ *shows the input feature size given a scene scaled to the unit cube.*

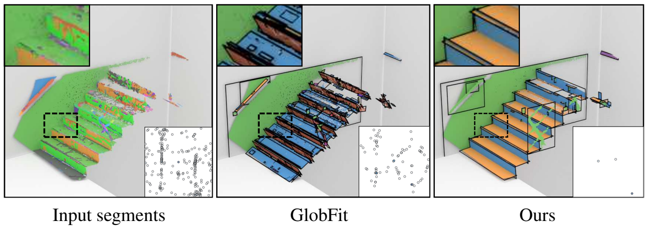

Figure 7: *Comparison to [Li et al. 2011a] on a Kinect® scan. The moderate number of input planes (300) prohibitively increases the likelihood that GlobFit will commit early to an irrelevant relation in the scene yielding over-regularization. There, stair steps were aligned to a single plane spanning the whole staircase, and risers were rotated orthogonally. In contrast, our approach correctly reconstructs the stairs, handlebars, and the wall, initialized with these 300* or *all 1500 input planes. Input angles* $\Theta = \{0, \pi/2\}$.

We used available implementations of the algorithms, except for the latter, where we asked the authors for comparison. RANSAC uses a probabilistic framework that is least sensitive to the notion of finite planes. We reimplemented the propose, expand, and refit steps of PEARL with finite plane segments using the published α-expansion library. Its dense formulation frequently made it necessary to downscale the input point clouds. Point set structuring takes an arrangement of planes as input, and benefits more from our regularization as preprocessing. Using their full pipeline on our input still yields unregularized outputs (*c.f.,* Figure 10, Table 2). Except GlobFit, other methods have no notion of global relations, and hence are out-performed very easily by our method.

GlobFit. We used an improved version of the published implementation of [Li et al. 2011a] to perform extensive experiments on our scenes. We found that: i) GlobFit heavily relies on the quality of relations it first commits to; ii) memory demands quickly rise beyond practical magnitudes. We had to sort the primitives from our initialization in decreasing order w.r.t. assigned numbers of points to achieve any output. In Stairs, Nola, and Lans we observed a degradation of the quality of the output related to the number of input planes. *E.g.* in the Stairs (Figure 7) scene, given the 15 largest planes as input, we both were successful at reconstruction with perfectly oriented planes gathered in a single frame. Using a richer set of 300 planes (of 560) GlobFit had a much lower chance of initially picking the right relations and could not recover from such mistakes. Similar problems arose in Nola (>120 planes, Fig. 11), and Lans (>300 planes, Fig. 12). Empire is heavily polluted with outliers, which required us to provide GlobFit with the 13 largest planes only (see Figure 10). Otherwise, output was a meaningless series of well-distributed, parallel planes.

6.2 Results

We evaluated our approach by reconstructing scenes with varying noise levels, sampling, complexity and regularity, both in 2D and 3D. The numbers of input planar segments are reported in Table 1, as well as the input scale parameter ρ, the number of output primitives, identified representative orientations (frames), and discovered pairwise perfect relations: $\#rels = \sum_i \binom{\sum_j \chi_{i \to j}}{2}$.

Generally, RAPTER produced more accurate segmentations of the input pointclouds, and critically, arrangements with higher regularity. Most importantly, we simultaneously preserved independent orientations for smaller groups of planes. We reorient points in the input pointcloud using the normals of the assigned planes in our output segmentation. To visualize the distribution of point normals in a scene, we create histogram plots by mapping them to a rectangle using HEALPix's Cartesian mapping. A saturated circle shows high concentration of normals, a white circle denotes less populated, but concentrated orientations, well preserved by our method. Matching colors in the outputs indicate parallel normals ($<0.01°$).

2D. We prepared a rasterized image to demonstrate vectorization, shown in Figure 4. The input was created from an RGB image by thresholding at 40% luminosity. Despite the differences in sampling density and variations in feature scale (thick and thin walls), we obtain a good reconstruction of the main layout of the room, and also manage to recover the spatial layout in ill-sampled locations, while respecting sudden local orientation changes and preserving important details. Also note the significant reduction in the uncertainty of the orientations in the image. The extracted RAP capture the different orientation frames (denoted by same color) across multiple feature scales. Due to PEARL's spatially regularized formulation, it managed to recover the well-sampled, principal directions in the drawing. In contrast, RANSAC used more primitives to explain the hatched areas than the more meaningful borders (see Figure 8).

We created the simulated scan of a room in Figure 2 to demonstrate the importance of regularization and preservation of details. We used the input angle generators $\Theta^* = \{\pi/3, \pi/2\}$ to solve the scene. Despite the apparent simplicity of the solution, note, that

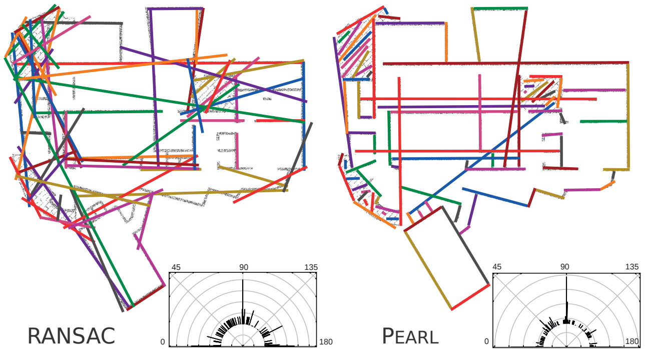

Figure 8: *Extracted lines from the input pointcloud shown in Figure 4 and angular distributions. Only a fraction of inter-line regularities are found in the noisy data. Input angles* $\Theta = \{0, \pi/2\}$.

there are four different orientation frames in the scene, therefore the challenge really is to preserve their independence. Although some triangle edges are almost parallel to rectangle edges with overwhelming popularity, we preserve the more accurate data fit due to enough evidence for existence of that orientation and the spatial regularity cost detailed in Equation (5). Our method is especially useful, when biased noise and occlusions would distract other methods bound to spatial reasoning. The resulting RAP managed to capture scene diversity without over-committing to any particular orientation, *i.e.,* room walls (red) or dining table (green).

L-house. We evaluated performance on a synthetic scan corrupted by non-uniform, sparse sampling (Figure 9). According to our output normal distribution, the six main directions were detected accurately, even small clusters in the corner were correctly oriented. In comparison, the roof planes returned by RANSAC are not perfectly aligned. PEARL performs well on this example due to spatial regularization and reliable data. GlobFit can correctly identify the reliable relations amongst the small number of input planes (91).

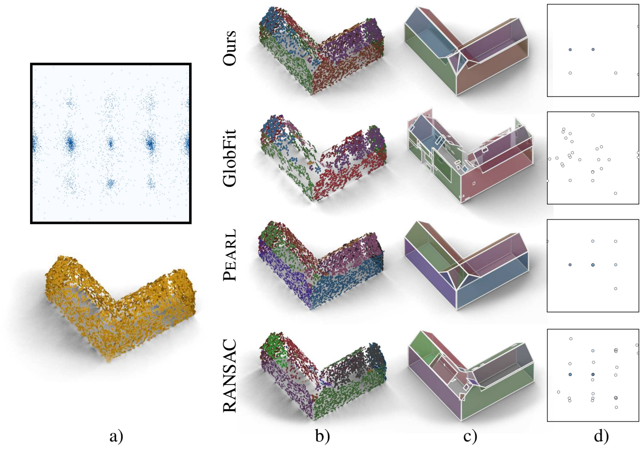

Figure 9: *Comparison to other methods on “L-house” (a). We show points reprojected to associated planes (b), and approximated by a set of planar polygons (c). Input (a-top) and output (d) normal distributions are mapped to 2D. Input angles* $\Theta = \{0, \pi/2\}$.

Empire. We illustrate how our approach can be used to reconstruct data corrupted by outliers on Empire (Figure 10). In addition to the recognised regularity of the building with three main directions, we properly detected small tilted components around edges. Note the cylinder on the top of the building, approximated by planes favoring the global arrangement orientations. RANSAC over-simplified the geometry, because of the excessive presence of outliers. PEARL managed to produce an arrangement of planes with a more compact normal distribution, however the regular structure of the building was lost during the process (see top view). Many details were missed due to the delicate balance between complexity and spatial smoothness. GlobFit did well given the largest 13 planes as input (of 4693). Given more, it could not deal with the amount of outliers, hence orientations corresponding to the smaller details at the top of the building are missing from the normal distribution.

Nola. Nola is a good example for our approach (see Figure 11). The building is exclusively composed of planar components acquired using a long-range scanner, which lead to large, occluded and ill-sampled regions. The regularity of the geometry is well exploited by our approach, balconies are properly detected and perfectly aligned. Note that all planes are either vertical (walls) or horizontal (floors). RANSAC discovered the main directions, but completely failed with smaller structures and large, slanted planes. PEARL managed to extract several horizontal planes (green balconies in top view) but failed to properly segment the geometry due to data quality. GlobFit could not recover from early commitment to some relations in its input, 120 (of 15685) most sampled patches.

Lans. In Figure 12 our approach faithfully reconstructs various shapes including a cone, an octagonal pyramid, and small non-symmetric tetrahedrons (top of the tower). The relations needed for the cone were not prescribed, hence our approach did not regularize there. The simultaneous extraction of relations and primitives allowed us to preserve small structures, such as the alcove or the

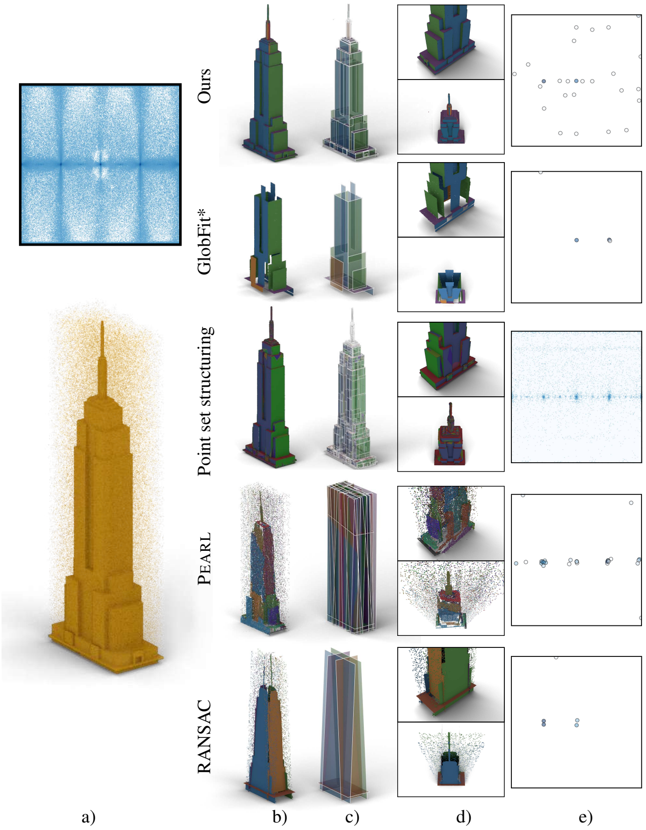

Figure 10: *Comparison to other methods on “Empire” (a). We show points reprojected to associated planes (b), and approximated by a set of planar polygons (c) (zooms in (d)). Input (a-top) and output (e) normal distributions are mapped to 2D. Main difficulty here is detecting and segmenting structures of very different sizes whilst disregarding outliers. GlobFit only had the 13 (of ~4700) most supported planes as input. Input angles* $\Theta = \{0, \pi/2\}$.

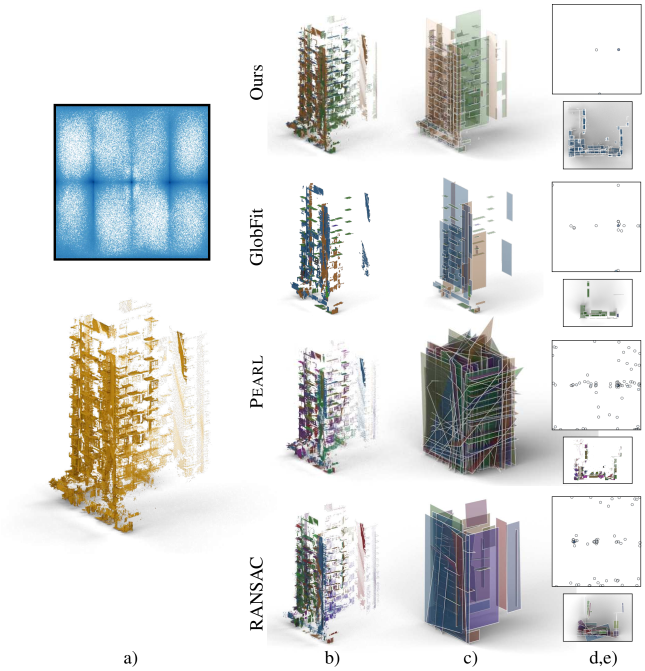

Figure 11: *Comparison to other methods on "Nola" (a). We show points reprojected to associated planes (b) and approximated by a set of planar polygons (c) (zooms in (d)). Input (a-top) and output (e) normal distributions are mapped to 2D. GlobFit had the largest 120 (of 15865) input planes. Input angles* $\Theta = \{0, \pi/2\}$.

windows in the wall, despite strong variations in sampling. Some relations were recoverable by only relying on the data (RANSAC, PEARL). RANSAC failed to differentiate small and nearby similar structures due to the conflict between low complexity and attention to detail (tower wall and roof). PEARL worked with a subsampled pointcloud (10%) due to performance reasons. GlobFit repeatedly over-regularizes the scene by enforcing relations (roof, tower top) even given just a moderate part of our input (300 of 7441).

Euler. The power of our multi-scale design is illustrated by the scenes "Euler" and "Euler-Cut" (Figures 1 and 6), containing large (walls, etc.), medium (tables), and small structures (stairs, risers, table feet). Due to the simultaneous identification of primitives and relations, our approach easily outperforms RANSAC when segmenting the geometry. For example, at the stairs landing, where corrupted data is over-merged (by RANSAC), mixing the floor and the three first stairs. Also note the robust repeatability of our approach over cluttered geometry (conference room): where many chairs are segmented similarly (brown front legs, green back legs, seats in dark green, backs two-colored). We could not run other methods (PEARL, GlobFit) on these scenes (500k-4M points).

Generally, well sampled scene parts can allow other methods to recognize some relations. However, RAPTER is robust towards sampling density and low local contrast of features, whilst capable of solving large scale scenes. Most importantly it can correctly identify non-dominant directions besides main scene orientations.

6.3 Quantitative evaluation

We evaluated our method quantitatively (Table 2) and also show the range of control offered by the regularization parameter (Figure 13). In Figure 14 we investigated robustness to initialization.

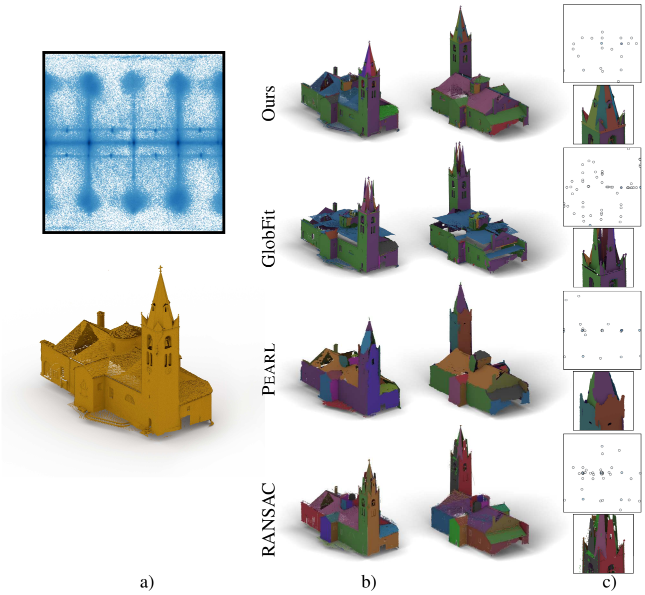

Figure 12: *Comparison to other methods on "Lans" (a). (b) Points reprojected to associated planes and zooms in (c). Input (a-top) and output (c) normal distributions are mapped to 2D. GlobFit had 300 (of 7441) input planes. Input angles* $\Theta = \{0, \pi/2\}$.

Ground truth. We retrieved ground truth triangle meshes for three scenes, face normals serve as ground truth orientation for points. We only used faces with edges $>\rho$, and ground truth points, where there is a single face within ρ distance due to lack of assignments.

Error metrics. We target to evaluate the successfully identified relationships in the scene through the relative angle of normals assigned to points by different methods (Table 2). Specifically, we compare the angle of normals of pairs of points to the angle of corresponding ground truth normals. We report the mean and standard deviation of absolute difference of relative angles: $\forall_{g,h} \ \left|\angle\left(\mathbf{n}(p_g), \mathbf{n}(p_h)\right) - \angle\left(\mathbf{n}(p_g^{GT}), \mathbf{n}(p_h^{GT})\right)\right|$. We considered only points, that were plausibly reconstructed by their assigned planes, *i.e.,* their reprojection was close ($<\rho$) to their reprojection to the ground truth mesh. We were especially interested in how relative relations were successfully recovered *perfectly*, *i.e.,* up to 0.1° compared to the ground truth relative angle of a point pair. In addition to our scalability and ability to respect diversity, numbers show a clear advantage over other methods.

Regularization. We show the amount of control and power the user has through the parameter λ, see Figure 13. When large diversity is expected in the input, or the input scans have been preprocessed, one can choose to respect the data and not enforce regularity (left side of the figure). Given prior knowledge about relations in the underlying scene, one can choose not to trust the possibly moderately or very noisy data containing biased sampling and registration mistakes. In most scenarios, the middle column would be the sought outcome. However, especially given registration errors, outcomes towards the right side show the true power of our method.

Robustness to initialization. In Figure 14 we evaluated the robustness of the algorithm to the quality of the initialization. Note, that the fragmentation of the initialization depends on the spatial threshold ρ and is only loosely coupled with the threshold parameter τ. We show, how we recover from a wide range of settings and degrade gracefully, when the the parameters were obviously ill-chosen.

Limitations. Our novel formulation with meta-nodes equipped the algorithm to become robust to the pointcloud size, the number of

Scene (#pts)	Method	Mean (SD)	$< 0.1^\circ$	Recall
L-house (5k)	RANSAC	10.6° (3.6°)	35.1%	**100.0%**
	PEARL	12.4° (3.9°)	34.2%	**100.0%**
	GlobFit	11.5° (3.8°)	73.9%	99.4%
	RAPTER (ours)	**10.1**° (**3.6**°)	**75.4%**	99.8%
Stairs (261k)	RANSAC	**7.7**°(**2.2**°)	13.3%	84.7%
	PEARL	17.6° (4.0°)	16.4%	84.5%
	GlobFit	13.3° (2.9°)	8.1%	**99.6%**
	RAPTER (ours)	9.4° (3.5°)	**65.7%**	98.2%
Empire (1.2M)	RANSAC	**2.8**°(**0.4**°)	20.5%	21.7%
	PEARL	20.8° (4.3°)	9.6%	35.4%
	GlobFit	6.3° (3.0°)	80.5%	65.8%
	[Lafarge&Alliez '13]	4.6°(2.5°)	89.8%	93.7%
	RAPTER (ours)	4.1° (2.0°)	**94.6%**	**99.1%**

Table 2: *We compare the relative angles of normals of point pairs in the output clouds and normals from ground truth triangle meshes. A point is considered recalled, if its reprojection on its assigned plane is at most at* $<\rho$ *distance from its reprojection on the ground truth mesh, higher is better. RANSAC achieves a lower mean deviation (lower is better) between the angles of normals by fitting to many well-sampled planes, some false positive primitives fit to outliers. Our high recall scores show that our precision comes with plausibly located output planes. We are especially interested in how well perfect relations were discovered, we therefore estimate precision (higher is better) of identified, perfect pairwise relative relations.*

primitives, and to be able to process large scale scenes as shown in Figure 1. However, this comes at the cost of quadratic constraints, which requires more computation time during the optimization stage. With our current implementation, runtimes vary from seconds (L-house) to a couple of hours for massive scenes (Euler) on a quad-core Intel i7-4700MQ (2.40GHz, 16GB RAM).

Our solution solves a selection problem iteratively inspired by Isack and Boykov [2012], allowing generation of primitives not only from the data but also previously computed approximations. Despite RAPTER's shown efficiency, advancing to very large scale datasets might be difficult. Due to global constraints, concurrent access to a very large database of points is required, which complicates optimisation on distributed systems. However, large scale scenes (*e.g.,* building-scale) can be processed without further optimizations.

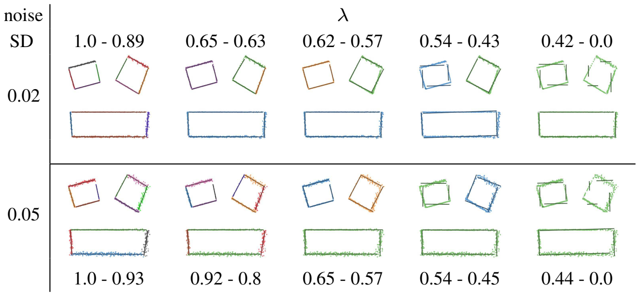

Figure 13: *Effect of regularization parameter* λ *given two levels of normally distributed noise. Data was captured from a point scanner in the center of each sub-image, with biased sampling and registration errors. The regularization parameter* λ *spans between the extremes of allowing a user to either let the data guide the reconstruction, or to enforce a set of orientations on the whole scene. Intervals of* λ *lead to the same results. For really noisy scenes, higher* λ *values are appropriate. Input angles* $\Theta = \{0, \pi/2\}$.

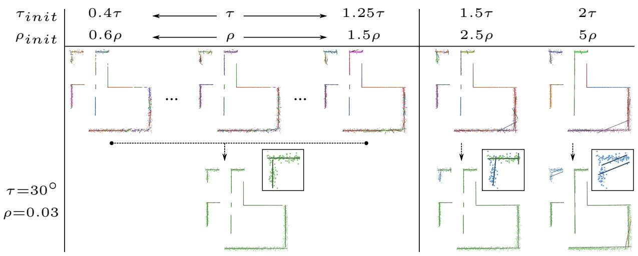

Figure 14: *Robustness to initialization. Our method is capable of producing consistent outputs given a wide quality variation of the initialization. We performed the initial segmentation using different angular* (τ_{init}) *and spatial* (ρ_{init}) *thresholds (top row). We then optimized using* $\tau = 30^\circ, \rho = 0.03$ *to retrieve the bottom row. Despite challenges presented by erroneously under-segmented point patches, our output performance degrades gracefully. Situations, where the parameters were set too high (top-right) are easily recognized. Hence, one should choose to initialize by conservatively setting the thresholds to lower values. Input angles* $\Theta = \{0, \pi/2\}$.

7 Conclusions

We presented an algorithm to abstract raw scans by regular arrangements of primitive planes by *simultaneously* extracting a set of primitives along with their inter-primitive relations. The proposed formulation favors globally consistent RAP, even at the cost of an increased data fitting error. The main novelty in the proposed formulation is to allow less represented orientation groups *not* to be masked by a more dominant scene orientation. Algorithmically, we first expand the solution space by generating potential candidate primitives, and then repose the scan abstraction problem as an instance of globally coupled primitive selection problem. The resultant algorithm runs from a coarse-to-fine scale, leading to a robust and scalable algorithm as demonstrated on many test scenarios.

An obvious next step is to add support for other primitive types (e.g., cylinders, cones). More interestingly, in the future, we would like to support other relations including equal spacing and equal length in the global formulation. One possibility is to model such second order relations by using pair of primitive pairs (i.e., quartet of primitives) in the candidate generation stage. However, the resultant candidate blowup will require rethinking the optimization formulation without sacrificing the global selection criteria.

Acknowledgements

We thank the reviewers for their comments and suggestions for improving the paper. Special thanks to Florent Lafarge for help with comparisons and datasets, Neil Smith and Hui Lin for additional datasets, David Vanderhaghe for his support, and Julian Straub for extra comparisons. We thank Simon Julier, Duygu Ceylan, Moos Hueting, Melinos Averkiou, Peter Hedman, James Hennessey, Clément Godard, Martin Kilian, Bongjin Koo and Veronika Benis for invaluable comments, support and discussions. This work was supported in part by ERC Starting Grant SmartGeometry (StG-2013-335373), Marie Curie CIG, ANR Mapstyle project (ANR-12-COORD-0025) and EU project CR-PLAY (no 611089) www.cr-play.eu.

References

ANAND, A., KOPPULA, H. S., JOACHIMS, T., AND SAXENA, A. 2011. Contextually guided semantic labeling and search for 3D point clouds. *CoRR*.

Arikan, M., Schwärzler, M., Flöry, S., Wimmer, M., and Maierhofer, S. 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM TOG 32*.

Bonami, P., et al. 2008. An algorithmic framework for convex mixed integer nonlinear programs. *Discret. Optim. 5*.

Boulch, A., de La Gorce, M., and Marlet, R. 2014. Piecewise-planar 3D reconstruction with edge and corner regularization. *Computer Graphics Forum 33*.

Chen, K., Lai, Y.-K., Wu, Y.-X., Martin, R., and Hu, S.-M. 2014. Automatic semantic modeling of indoor scenes from low-quality RGB-D data using contextual information. *ACM SIGGRAPH Asia 33*.

Chum, O., and Matas, J. 2005. Matching with PROSAC: Progressive sample consensus. In *ICCV*.

Furukawa, Y., Curless, B., Seitz, S. M., and Szeliski, R. 2009. Manhattan-world stereo. In *CVPR*.

Gallup, D., Frahm, J.-M., Mordohai, P., Yang, Q., and Pollefeys, M. 2007. Real-time plane-sweeping stereo with multiple sweeping directions. In *CVPR*.

Gschwandtner, M. 2013. *Support Framework for Obstacle Detection on Autonomous Trains*. PhD thesis, Univ. Salzburg.

Isack, H., and Boykov, Y. 2012. Energy-based geometric multi-model fitting. *IJCV 97*.

Kim, Y. M., Mitra, N. J., Yan, D.-M., and Guibas, L. 2012. Acquiring 3D Indoor Environments with Variability and Repetition. *ACM SIGGRAPH Asia 31*.

Kim, B., Kohli, P., and Savarese, S. 2013. 3D Scene Understanding by Voxel-CRF. *IEEE ICCV*.

Koppula, H., Anand, A., Joachims, T., and Saxena, A. 2011. Semantic labeling of 3D point clouds for indoor scenes. *NIPS*.

Lafarge, F., and Alliez, P. 2013. Surface reconstruction through point set structuring. In *Proc. of Eurographics*.

Li, Y., Wu, X., Chrysanthou, Y., Sharf, A., Cohen-Or, D., and Mitra, N. J. 2011. GlobFit: Consistently fitting primitives by discovering global relations. *ACM TOG 30*.

Li, Y., Zheng, Q., Sharf, A., Cohen-Or, D., Chen, B., and Mitra, N. J. 2011. 2D-3D fusion for layer decomposition of urban facades. In *IEEE ICCV*.

Lin, H., Gao, J., Zhou, Y., Lu, G., Ye, M., Zhang, C., Liu, L., and Yang, R. 2013. Semantic decomposition and reconstruction of residential scenes from lidar data. *ACM SIGGRAPH*.

Mattausch, O., Panozzo, D., Mura, C., Sorkine-Hornung, O., and Pajarola, R. 2014. Object detection and classification from large-scale cluttered indoor scans. *CGF*.

Mehra, R., Zhou, Q., Long, J., Sheffer, A., Gooch, A., and Mitra, N. J. 2009. Abstraction of man-made shapes. *ACM SIGGRAPH Asia 28*.

Nan, L., Sharf, A., Zhang, H., Cohen-Or, D., and Chen, B. 2010. SmartBoxes for interactive urban reconstruction. *ACM TOG*.

Ni, K., Jin, H., and Dellaert, F. 2009. GroupSAC: Efficient consensus in the presence of groupings. *IEEE ICCV*.

Niessner, M., Zollhöfer, M., Izadi, S., and Stamminger, M. 2013. Real-time 3D reconstruction at scale using voxel hashing. *ACM SIGGRAPH Asia*.

Niessner, M., Dai, A., and Fisher, M. 2014. Combining inertial navigation and ICP for real-time 3D surface reconstruction. *CGF Eurographics*.

Oesau, S., Lafarge, F., and Alliez, P. 2014. Indoor scene reconstruction using feature sensitive primitive extraction and graph-cut. *ISPRS Journ. Photogramm. and Remote Sensing 90*.

Pham, T. T., Chin, T.-J., Schindler, K., and Suter, D. 2014. Interacting geometric priors for robust multimodel fitting. *IEEE Transaction on Image Processing 23*.

Ramalingam, S., and Brand, M. 2013. Lifting 3D manhattan lines from a single image. *IEEE ICCV*.

Rusu, R. B., and Cousins, S. 2011. 3D is here: Point Cloud Library (PCL). In *ICRA*.

Schenk, O., and Gärtner, K. 2004. Solving unsymmetric sparse systems of linear equations with PARDISO. *Future Gener. Comput. Syst.*.

Schnabel, R., Wahl, R., and Klein, R. 2007. Efficient RANSAC for point-cloud shape detection. *CGF 26*.

Shao*, T., Monszpart*, A., Zheng, Y., Koo, B., Xu, W., Zhou, K., and Mitra, N. 2014. Imagining the unseen: Stability-based cuboid arrangements for scene understanding. *ACM SIGGRAPH Asia*.

Sharf, A., Huang, H., Liang, C., Zhang, J., Chen, B., and Gong, M. 2013. Mobility-trees for indoor scenes manipulation. *CGF*.

Shen, C.-H., Fu, H., Chen, K., and Hu, S.-M. 2012. Structure recovery by part assembly. *ACM SIGGRAPH Asia 31*.

Silberman, N., and Fergus, R. 2011. Indoor scene segmentation using a structured light sensor. In *Proc. ICCV - Workshop on 3D Representation and Recognition*.

Sinha, S. N., Steedly, D., Szeliski, R., Agrawala, M., and Pollefeys, M. 2008. Interactive 3D architectural modeling from unordered photo collections. *ACM TOG 27*.

Straub, J., Rosman, G., Freifeld, O., Leonard, J. J., and Fisher III, J. W. 2014. A Mixture of Manhattan Frames: Beyond the Manhattan World. *IEEE CVPR*.

Wächter, A., and Biegler, L. T. 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. *Mathematical Programming 106*.

Yan, F., Sharf, A., Lin, W., Huang, H., and Chen, B. 2014. Proactive 3D scanning of inaccessible parts. *ACM SIGGRAPH*.

Yumer, M. E., and Kara, L. B. 2012. Co-abstraction of shape collections. *ACM SIGGRAPH Asia 31*.

Zheng, Q., Sharf, A., Wan, G., Li, Y., Mitra, N. J., Chen, B., and Cohen-Or, D. 2010. Non-local scan consolidation for 3D urban scene. *ACM SIGGRAPH 29*.