# Imagining the Unseen: Stability-based Cuboid Arrangements for Scene Understanding

Tianjia Shao* Zhejiang University
Aron Monszpart* University College London
Youyi Zheng Yale University
Bongjin Koo University College London
Weiwei Xu Hangzhou Normal University
Kun Zhou Zhejiang University
Niloy J. Mitra University College London

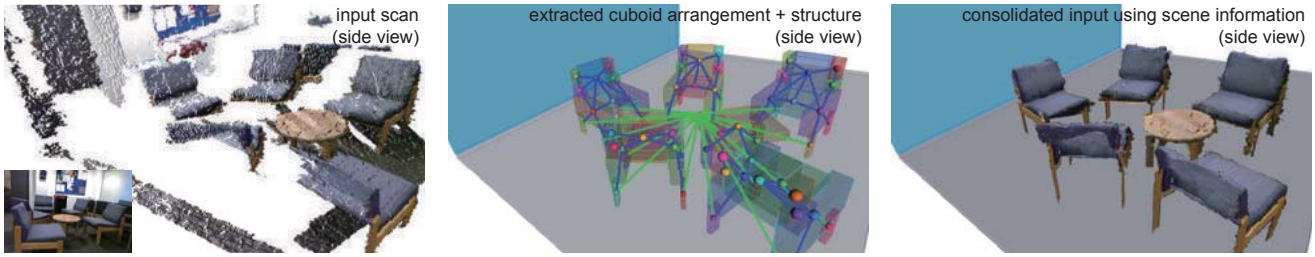



**Figure 1:** *Starting from a heavily occluded single view RGBD image (left), we extract a coarse scene structure as an arrangement of cuboids along with their inter-cuboid relations (middle) using physical stability considerations to hypothesize the missing regions/relations. The coarse structure can then be used for scene understanding and manipulation. This result was generated in the automatic mode.*

## Abstract

Missing data due to occlusion is a key challenge in 3D acquisition, particularly in cluttered man-made scenes. Such partial information about the scenes limits our ability to analyze and understand them. In this work we abstract such environments as collections of cuboids and hallucinate geometry in the occluded regions by globally analyzing the physical stability of the resultant arrangements of the cuboids. Our algorithm extrapolates the cuboids into the unseen regions to infer both their corresponding geometric attributes (e.g., size, orientation) and how the cuboids topologically interact with each other (e.g., touch or fixed). The resultant arrangement provides an abstraction for the underlying structure of the scene that can then be used for a range of common geometry processing tasks. We evaluate our algorithm on a large number of test scenes with varying complexity, validate the results on existing benchmark datasets, and demonstrate the use of the recovered cuboid-based structures towards object retrieval, scene completion, etc.

**Keywords:** box world, proxy arrangements, physical stability, shape analysis

**Links:** DL PDF WEB DATA CODE

## 1 Introduction

Acquisition devices for 3D geometry are now ubiquitous. While this has vastly simplified the data gathering process, the raw data still remains difficult to use. A fundamental problem for single view acquisition and even for multi-view consolidated scans is missing observations due to scene occlusion. This problem is particularly acute in busy and cluttered scenes (e.g., indoor environments). This limits how such raw scans can actually be used as existing shape analysis tools fail due to missing information.

Various approaches have been proposed to address the ill-posed problem of *hypothesizing* data in the occluded scene regions. For isolated objects, one can use local context [Sharf et al. 2004; Harary et al. 2013] or deform class-specific template shapes to recover the missing parts [Pauly et al. 2005; Bao et al. 2013]; for multiple objects, one can employ search-and-classify approaches [Nan et al. 2012], require multi-view acquisition [Kim et al. 2013b; Mattausch et al. 2014], or interactively capture template models using targeted acquisition [Kim et al. 2012] or modeling sessions [Arikan et al. 2013]. Such methods, however, assume access to suitable database models, or rely on much more detailed and complete scene data to reliably perform geometric matching.



**Figure 2:** *Even for scenes with significant occlusion, we as humans, can reason about the actual arrangement (image source: [Gibson 1986]). We develop an algorithm to mimic this based on physical stability of the inferred arrangement (see also Figure 16).*

*Joint first authors.

In this paper, we use physical stability considerations of static scene objects to hallucinate missing structure information from incomplete 3D acquisitions. Moreover, by considering possible geometry in the unseen part, we can better reason about the visible parts of the scans (e.g., if two objects just touch, or are fused to each other). This is similar to how we as humans regularly extrapolate seen object parts to reason about the unseen scene regions based on the physical plausibility of the resultant object stack, for example see Figure 2. Note that by avoiding the surface reconstruction or partial shape matching problems, our algorithm does not reply on availability of model templates or predefined priors (see Figure 1).

Starting from raw scans, our goal is to abstract indoor scenes as a collection of simple cuboids. We abstract indoor scenes as collections of cuboids as the 'box world' well captures the intra- and inter-object constraints common in man-made environments (c.f., [Blum et al. 1970; Gupta et al. 2010; Zheng et al. 2012]). Creating such an abstraction requires addressing the following: (i) segmenting the scan into groups of points; (ii) fitting a cuboid to each such group; (iii) hypothesizing missing cuboid geometry in the occluded regions; and (iv) determining how the cuboids are mutually arranged and connected to each other. Essentially, our goal is to determine what the cuboids are, what their spatial and geometric parameters are, and how they are relatively arranged (e.g., stacked or fixed to each other) such that they best explain the (incomplete) input data.

The algorithm starts by creating a set of initial cuboids. Based on their pairwise configurations, we extrapolate the initial cuboids to spawn a set of candidate cuboid *extensions* to hypothesize possible completions in the occluded regions. The algorithm then proceeds in two stages: a discrete stage to extract inter-cuboid connections by selecting the cuboid arrangement from a subset of candidate cuboids arrangements that best explains the input; and a continuous stage to refine the cuboid parameters to improve the stability of the current arrangement, while preserving the current cuboid connections to obtain a final cuboid arrangement. Goodness of any cuboid arrangement is assessed based on the physical stability of the configuration and how well the arrangement agrees with the input observation.

We evaluated our algorithm on synthetically scanned scenes with known groundtruth data, on typical indoor scenes obtained from benchmark datasets (about 700+ scenes from the NYU depth dataset with available groundtruth structure data), and on complex office environments with significant occlusions. The tests demonstrate that our algorithm produces high quality structure abstractions of the occluded regions. Moreover, the improvements are significant even in case of scans consolidated using multiple acquisitions (e.g., Kinect fusion). Finally, we demonstrate the use of the recovered coarse structure in object retrieval and scene completion.

## 2 Related Works

**Indoor scene capture.** Acquiring and understanding indoor scenes are valuable for various applications. A common approach is to design a classifier to label scene objects [Schlecht and Barnard 2009; Xiong and Huber 2010; Anand et al. 2011; Koppula et al. 2011]. Siberman et al. [2011; 2012] reported good performance accuracy in indoor scene labeling using learned probabilistic discriminative models for data and compatibility terms from RGBD image dataset and by considering support relationships. In another research thread, interactive methods have been developed for 3D modeling and content-aware image editing. Shao et al. [2012] segmented RGBD images with semantic labels, and retrieve 3D models corresponding to the labels from a database using a random regression forest with the depth data. In concurrent efforts, Kim et al. [2012] developed model-based algorithm to group the extracted geometric primitives from depth data into individual models, while Nan et al. [2012] interleaved segmentation and classification in region growing to extract objects from depth data. More recently Mattausch et al. [2014] propose an unsupervised algorithm to identify and consolidate repeated objects across large office scenes.

**Proxy-based scene understanding.** Instead of assuming access to all possible scene objects, an alternative approach is to explain the input in terms of ensembles of simple proxies. Li et al. [2011] and Lafarge et al. [2013] consider relations among initial RANSAC-based proxies to produce structured outputs; while, Arikan et al. [2013] combine user annotations with prior relations to create abstracted geometry. Other approaches involve encoding input scenes as collections of planes, boxes, cylinders, etc. and studying their spatial layout [Gupta et al. 2010; Lee et al. 2010; Hartley et al. 2012]. In the context of image manipulation of indoor man-made environments, Zheng et al. [2012] abstract scenes with cuboid primitives to facilitate simple yet intuitive edits. These methods, however, do not yield information about the occluded scene parts, which is the focus of this work.

Gupta et al. [2011] reason about object layouts based on human interactions in typical workspaces. In order to improve accuracy of cuboid proxy detection, a statistical deformable cuboid model has also been learned [Fidler et al. 2012], or a cuboid corner point model is learned to better detect corners in images, eventually leading to better cuboid models [Xiao et al. 2012]. Hedau et al. [2012] design a set of image space contrast-based features to better fit proxy boxes to explain scenes from single views. Physical constraints, such as penetration-free, etc. are also adopted in reasoning the position of proxies in an indoor scene [Hedau et al. 2010], while Umetani et al. [2012] use physical stability and torque limits for guided furniture design in a modeling and synthesis setting. In parallel efforts, physical validity constraints [Jia et al. 2013; Jiang and Xiao 2013; Zheng et al. 2013] have also been used for voxel-based scene parsing. The methods mostly use local reasoning on physical stability, which by itself is not sufficient to ensure global stability (e.g., all the 3 blocks in Figure 10a have to be simultaneously optimized to produce a stable stack). Kim et al. [2013a] jointly estimate a voxel-based 3D reconstruction along with voxel labelings using a voxel-CRF formulation. We also focus on proxy-based scene understanding, particularly considering global physical stability to reason about large scale missing geometry, but without access to rule-based priors or any training data.

**Shape completion.** Occlusion naturally results in missed data. Various approaches have been proposed to plausibly complete such missing regions: diffusion-based smooth surface completion [Davis et al. 2002]; example-based completion based on existing database models [Sharf et al. 2004; Pauly et al. 2005]; local context-based completion [Harary et al. 2013]; consolidating information across multiple scans and by factoring out symmetry based redundancies [Zheng et al. 2010]. Since finding an appropriate shape from model repositories is often ill-posed especially in absence of sufficient data, Shen et al. [2012] introduce a structure-based approach to extract suitable model parts and fuse them together to form high-quality models. Note that in these cases the resultant model, retrieved as a whole or as an assembly of parts, remains plausibly connected based on the connectivity inherited from the source models. The methods begin to fail in presence of significant clutter and occlusion.

## 3 Overview

Our goal is to create an arrangement of cuboids as a structural proxy for an incomplete scan of a cluttered scene. By *arrangement of*

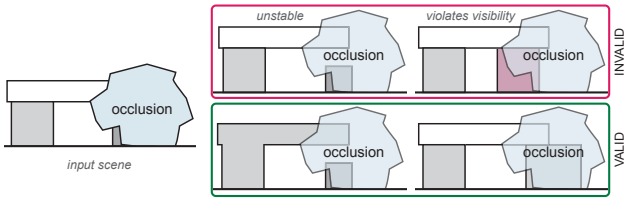



**Figure 3:** *Based on incomplete scene observation (left), we can imagine multiple completions. Top-right shows invalid completions: an unstable stack of three cuboids, or a stable stack of three cuboids that violates the visibility constraint. Bottom-right shows valid completions: two fused cuboids and a small cuboid, or a stack of three cuboids consistent with the visibility constraint.*

*cuboids*, we refer to a set of cuboids with their explicit pairwise contact relations, e.g., two cuboids touch, or are fused to each other, along with geometric attributes for each cuboid.

Starting from the raw data (e.g., single view RGBD data, or Kinect fusion data), we first create a set of initial cuboids (see Section 4.1). These initial cuboids, however, provide little information about the occluded scene regions. We make the important observation that pairs of such cuboids that intersect when extrapolated into the occluded regions provide potential geometry hypotheses for the occluded regions (see Figure 3). Hence, we generate multiple hypotheses for missing geometry by extending the initial cuboids into the occluded regions. We refer to such extrapolations as *cuboid extensions*. Note that each cuboid can have multiple potential extensions based on its contact types to the other cuboids. In this work, we consider no-contact, fixed contact, and touching as contact type relations. In Section 4.2, we describe how we enumerate possible cuboid arrangements $\{A_1, A_2, \dots\}$ where, each arrangement $A_i := \{B_1, B_2, \dots\}$ consists of a set of (extended) cuboids.

Working under the assumption that the scanned scenes consist of static objects, we expect any valid cuboid arrangement to be physically stable, or nearly-stable to account for errors due to coarseness of the initial cuboid estimates. Hence, we measure stability of any candidate cuboid arrangement $A_i$ under the respective inter-cuboid contact types and discard the unstable arrangements (see Figure 3). We propose a branch-and-bound algorithm to avoid explicitly checking the exponential set of possible cuboid arrangements.

Finally, we refine the cuboid parameters of the stable arrangement(s) while retaining current contact constraints using a quadratic programming formulation (see Section 4.3). We prefer the physically stable cuboid arrangement that requires fewer fixed joints and necessitate minimal volume extension as the simplest solution in terms of geometry that is hypothesized. In the end, we recover a physically valid arrangement of cuboids with both the inter-cuboids interaction types and the respective cuboid parameters.

# 4 Algorithm

Our goal is to infer a physically valid arrangement of cuboids to hypothesize geometry in the missing-data regions in an incomplete 3D point cloud $P$ (e.g., a single view RGBD image). By physical stability, we require the cuboid arrangement to be globally stable. This is achieved by creating arrangements of cuboids with suitable cuboid geometry and inter-cuboid contact relations (see Figure 4).

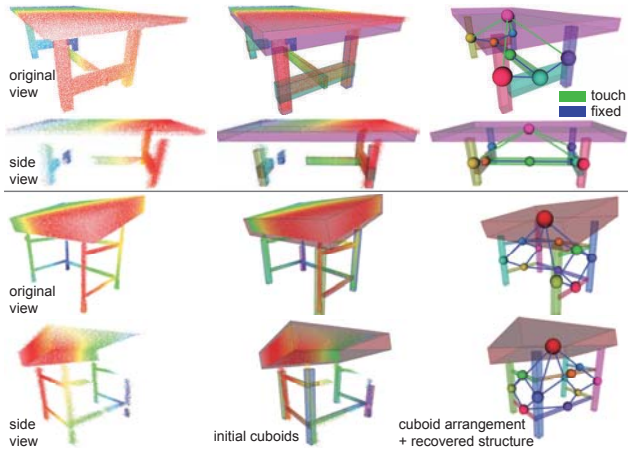



**Figure 4:** *Starting from incomplete scans (left), our algorithm starts by creating a set of initial cuboids (middle) that are optimized, both in their geometric parameters and the inter-cuboid contact types to propose a physically-valid arrangement of cuboids (right) that is consistent with the input data.*

## 4.1 Generating Initial Cuboids

We detect and remove the points in $P$ associated with the ground and wall planes, group the remaining points into separate clusters of point sets, and create initial cuboids for each such point cluster. For simple scenes, the initialization proceeds automatically, while for more complex scenes, user assistance is required. The generated cuboids are then refined to better align with the image space edges, where available. We now provide further details.

*Ground/wall planes:* We scale the input point set $P$ to fit inside a unit sphere and align the z-axis to the input up-direction. We use RANSAC to detect the dominant planes [Schnabel et al. 2007] in the input scene and sort the planes based on their size (i.e., area of extent). We mark the planar segment with upward normal and lowest z-value as the ground plane, and the other significant (based on an area threshold) planes that are orthogonal to the ground as walls (see [Kim et al. 2012]). Optionally, the user can mark the ground and wall planes. We remove the points associated with the ground and the wall planes, and group the remaining ones using a connected component analysis (using spatial proximity) as object clusters, say $\{C_1, C_2, \dots\}$. Note that each object cluster $C_i$ denotes a set of points.

*Notation:* We fit initial cuboids to each object cluster $C_i$. First we introduce some notations. A cuboid is an oriented box associated with a local coordinate frame. We represent a cuboid $B$ by its center $\mathbf{c}$, a local coordinate frame $\mathbf{F}$, and $(s_x, s_y, s_z)$ as the three size parameters along the three local coordinate axes. Note that along any axis a cuboid can be extended differently in the positive and/or the negative directions.

*Automatic cuboid creation:* For each object cluster $C_i$, we fit a bounding box aligned to the ground plane (see Figure 5-top). We orient the bounding box based on the best fitting rectangle to the points $p_j \in C_i$ projected to the ground plane. We voxelize (using 0.005 as cell size) and identify the occupied cells. Our goal is to fuse these voxels to form a set of non-overlapping rectangular slabs, each of which is nearly fully occupied. We take a greedy approach by fusing the occupied cells to form rectangular slabs if they contain more than 70% occupied cells (wrt. to the respective slab volumes). In the figure, the cuboids are marked as $1, 2, \dots$ based on their order of appearance. In case of multiple candidates we prefer the

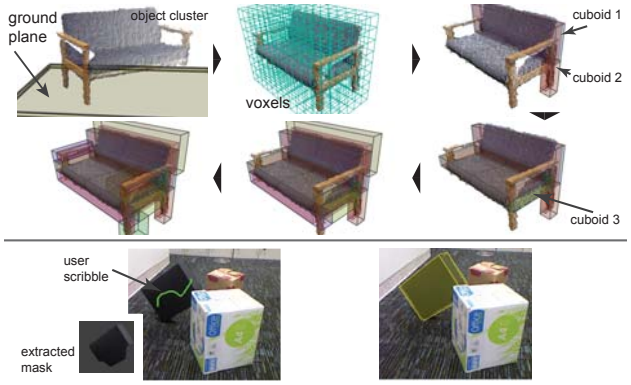



**Figure 5:** *(Top) In the automatic mode, cuboids are progressively fitted to an object cluster $C_i$ using a greedy approach. (Bottom) Based on user scribbles, a mask is produced from the RGBD data using depth-augmented grab-cut, and a cuboid is fitted to the corresponding points.*

one with the highest occupancy, and recurse over the remaining cells. In the end we have a set of cuboids for each object cluster.

*User interaction:* Optionally, the user can scribble strokes on the input image to guide segmentation (see Figure 5-bottom). This mode is used in scenes where objects come too close and fail to be separated using connected component analysis. Based on the strokes, we use a depth-augmented version of the original grab-cut segmentation [Rother et al. 2004] to select a group of points or mask. We run RANSAC on the selected points to generate candidate planes. The largest of the planes is selected as the primary plane, and the second largest plane is made orthogonal to the primary one. We extract the bounding box determined by these orthogonal directions (third direction is the cross product of the two plane normals). In our tests, such interactions were only necessary in regions of clutter and/or with slanted objects.

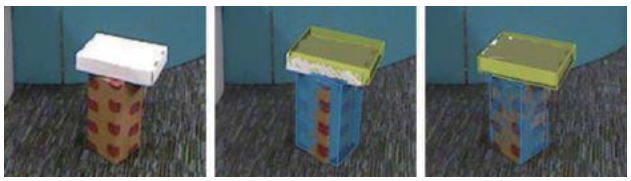


**Figure 6:** *The initial cuboids (middle) are adjusted to better align the original image edges in the RGDB data (left) to produce refined cuboid candidates (right).*

*Image-guided cuboid improvement of initial cuboids:* The cuboids generated, either automatically or semi-automatically, are based on the noisy point sets and hence are often misaligned to the image space edges. We refine the cuboids based on available image edge information (see Figure 6). Visible corner edges of the cuboids are mapped to the sufficiently close image edges, which are detected using Canny edge-detector (small edges and outliers are discarded). Given a group of cuboids $\{B_1, B_2, \dots\}$, we minimize:

$$\min_{\{B_i,T_i\}} \sum_i \sum_j \|\mathbf{P}(T_i(\mathbf{v}_j^s)) - \mathbf{t}_j\|^2, \qquad (1)$$

where $\mathbf{P}$ is the known camera projection, $\mathbf{v}_j^s$ is a sampled point on the visible corner edges, $\mathbf{t}_j$ is the closest point of $\mathbf{v}_j^s$ on the corresponding image edge, and $T_i$ is the unknown rigid transformation applied to the cuboid $B_i$. We update the closest points $\mathbf{t}_j$ at each iteration and use Matlab *fmincon* function to solve the nonlinear optimization in Equation 1. Note that if the offsets between cuboids and the point cluster degrade after refinement (due to spurious edges), we abandon the refinement.

**Algorithm 1** Inferring Inter-cuboid Contact Types

**Input:** $N$ initial cuboids $(B_1, ..., B_N)$
**Output:** $N$ optimized cuboids $(B_1^*, ..., B_N^*)$ and their contact relations $(J_1^*, ..., J_M^*)$

```
// Initialization of interaction graph G := (V, E)
G ← ∅
for i = 1 to N do
    V_i ← B_i
end for
for each node pair (V_i, V_j) do
    // build multi-edges e_ij^k
    e_ij^0 ← (B_i, B_j) // edge corresponding to initial geometry of B_i and B_j
    if B_i, B_j can potentially touch then
        for k = 1 to 3 do
            e_ij^k ← (B_i^{ij,k}, B_j^{ij,k}) //different extensions as in Figure 7
        end for
    end if
end for

// Optimization of cuboid geometry and their relation types
N_f ← N^2        // fixed joint number
δV ← π           // extended volume
while 1 do
    Gather a new combination of potential cuboid extensions (e_ij^k, ..., e_mn^l)
    if no more edge combination then
        break
    end if
    Update cuboids (B'_1, ..., B'_N) based on the extensions on potential edges
    if cuboids penetrate then
        continue
    end if
    Calculate current relations (J'_1, ..., J'_M), count of fixed joints N'_f, and volume
    extension δV' after interaction type pruning based on physics stablity
    if unstable according to Equation 3 then
        continue
    end if
    if N'_f ≤ N_f and δV' < δV, then
        N_f ← N'_f
        δV ← δV'
        (B_1^*, ..., B_N^*) ← (B'_1, ..., B'_N)
        (J_1^*, ..., J_M*) ← (J'_1, ..., J'_M)
    end if
end while
```

## 4.2 Inferring Inter-cuboid Contact Types

The initial cuboids do not occupy the occluded regions. In the key stage of the algorithm, we *extend* these cuboids into the occluded regions with the constraint that the arrangement (configuration) of the extended cuboids is physically stable. However, there are many ways of extending the cuboids into the occluded region. We first create candidate extensions and then pose the problem of selecting extensions such that the resultant arrangement of extended cuboids is physically stable. We use the formulation by Whiting et al. [2009] to assess physical stability in terms of the (unknown) cuboid dimensions. We now explain the individual steps (see Algorithm 1).

*Representing contact types:* We encode the discrete interaction among the cuboids as a *contact graph* $G := (V, E)$, where each cuboid $B_i$ becomes a node in $V$ and each pair of cuboids $(B_i, B_j)$ forms multiple candidate edges in $E$, one for each possible contact type (e.g., touch, fixed).

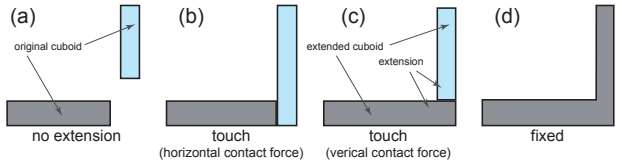



**Figure 7:** *2D Illustration of candidate cuboids generation cases for different contact types. Starting from the initial cuboids (a), we generate different candidate extensions: no extension (a), touching cuboids (b,c), and fused cuboids (d).*

We make an important observation that the rough geometry of the cuboids is largely determined by *how* they interact with each other and provide good extension candidates for the respective cuboid parameters. Each candidate cuboid pair amounts to possible contacts corresponding to different extensions for no-contact (i.e., disjoint), touching, or fixed (i.e., cuboids are fused together). Each such contact edge implicitly suggests cuboid geometries based on the respective cuboid extensions (see Figure 7). Say, any pair of cuboids $(B_i, B_j)$ is extended to be $(B_i^{ij,k}, B_j^{ij,k})$, where $k$ indicates the joint type. For example, edge type with no extension ($k = 0$) amounts to retaining the original cuboids, i.e., $B_i^{ij,0} = B_i$. Note that a cuboid $B_i$ can have multiple extensions proposed by different cuboids.

*Potential contact-based cuboid extensions:* For any candidate pair $(B_i, B_j)$, we extend the cuboids. Without loss of generality, assume $B_i$ is extended to be in contact with $B_j$. We first determine the extension direction through space partition. The extension length is initialized with the largest corner point to plane distance, i.e., the corner points of $B_j$ to the cube plane of $B_i$ attached to the overlapped half space. Figure 7 shows the different extension cases. Note that the type of extension also determines the force bearing faces, when applicable, which is later used during evaluating physical stability.

*Optimization formulation :* Let $\chi_i^{ij,k} = 1$ denote the corresponding extensions are selected ($\chi_i^{ij,k} = 0$ otherwise), which means, the $k$-th contact type for the cuboid pair $(B_i, B_j)$. A trivial solution is to select fixed contact types for all the cuboid pair relations to propose a physically stable solution. However, this is overly conservative. Instead, our goal is to propose a physically stable arrangement of (extended) cuboids with minimal number of fixed contacts, i.e., $\min \sum_{i,j,k} \#(\chi_i^{ij,k} = \text{fixed joint})$.

Further, the selected edges should also respect the following conditions: (i) $\chi_i^{ij,k} = \chi_j^{ij,k}$ and (ii) $\sum_k \chi_i^{ij,k} = 1$ for all the edges in $e_{ij}^k \in E$. The first constraint ensures that compatible cuboids are extended, while the second one ensures that only one type of interaction is selected for each cuboid pair relation.

The combination of all possible contact relation candidates is very large. An exhaustive search can take a prohibitive large amount of time to find the optimal solution, which is not acceptable in interactive applications. First, we prune the possible pairs of cuboids if their extensions violate the visibility constraint, i.e., if the extended proxies fall in the visible regions as informed by the source depth data. Then, we prune extensions using a branch-and-bound approach as described next.

*Pruning the solution space:* We prune contact relation types based on an important observation: If a candidate cuboid arrangement is unstable even with some of its contact relations marked as fixed, it cannot be made stable by changing the marked fixed type relations to touching or no-contact. Therefore, starting with root node where all the interaction types are set to be fixed, we expand the node as a search tree by changing relations at one edge from fixed to other types. If an expansion results in an unstable arrangement, the whole subtree is pruned to save unnecessary computation (see Figure 9).

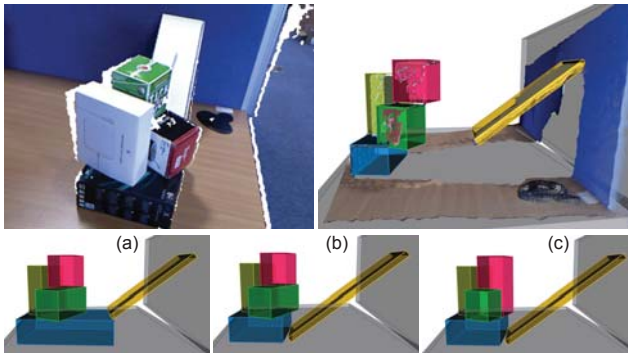



**Figure 8:** *A single view RGBD scan (top-left) can result in a physically implausible arrangement of initial cuboids (top-right), which can have multiple explanations in terms of arrangements of extended cuboids (bottom). Our formulation favors the one with the fewest number of fixed joints; and the smallest overall extension volume (c).*

*Assessing global stability:* Note that we cannot assess physical validity simply based on local reasoning. For example, in Figure 10a, all the pairs of touching cuboids by themselves are physically unstable, but the arrangement as a whole is still stable; while in Figure 10b, it is the other way round. Such scenes were common in the complex scenarios scanned and analyzed in Section 5.3. See also, Figure 15. Hence, in order to evenly quantify stability for such arrangements, we have to simultaneously consider the effects of all the cuboids.

*Assessing physical stability:* The stability of a cuboid arrangement is judged through its static equilibrium. This amounts to the set of contact forces between the cuboids to balance the arrangement under self load. Similar to [Whiting et al. 2009], we treat each cuboid as a rigid block, and position a contact force at each contact point on the interface. The number of contact points on the interface can vary according to the contact cases (1 for point contact, 2 for edge contact, and 4 for face contact). The contact forces are then parameterized at each interface face between two touching cuboids as the normal force $\mathbf{f}_n$, and the friction forces $\mathbf{f}_u$ and $\mathbf{f}_v$ (aligned to the normalized face directions $\mathbf{e}_n, \mathbf{e}_u, \mathbf{e}_v$, respectively). Since the interface forces are encoded in the local coordinate system, only 3 scalars are required in the final computation to capture their respective magnitudes. If two cuboids are fused to each other, we treat

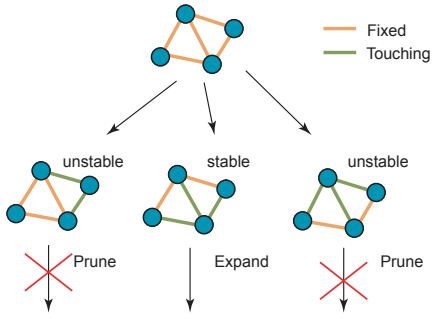



**Figure 9:** *Branch-and-bound approach to prune infeasible contact relation type assignments.*

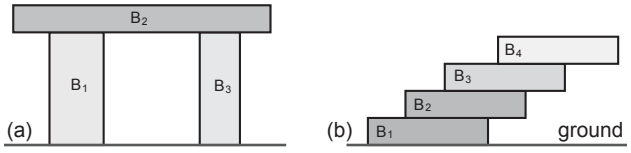



**Figure 10:** *In order to evaluate physical stability, simply investigating pairwise proxy interactions is insufficient. (Left) Although each pair of touching boxes is by itself unstable, the arrangement of all the boxes is stable. (Right) Although each pair of touching boxes is stable, the arrangement of all the boxes is unstable.*

them as a single rigid body and ignore their contact forces.

In static equilibrium, the net force and torque acting on the assembly should be zero. Gathering the constraints at each cuboid yields a linear system of the form,

$$\mathbf{D}\mathbf{f} + \mathbf{w} = 0, \tag{2}$$

where $\mathbf{w}$ captures the self weights of the cuboids (no external force is considered), $\mathbf{D}$ captures the coefficients of the force/torque equilibrium equations, and $\mathbf{f}$ captures the stack of (unknown) forces at the interface faces (see Figure 11).

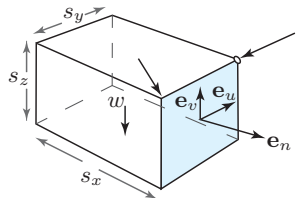



**Figure 11:** *For physical stability an arrangement of cuboids should satisfy zero force and zero torque conditions. The condition can be expressed in terms of the (unknown) forces at the interface faces and the weight $w$ of each cuboid expressed in terms of its (unknown) dimensions. In this figure, we highlight one interface face in blue. Note that the cuboid weight at its center of mass is given by $w = \rho s_x s_y s_z$ for a fixed density $\rho$.*

In order for a cuboid arrangement $\mathbf{A}$ to be physically stable, each $i$-th interface face forces $\mathbf{f}_n^i$ should be non-zero to act as compression force, and $\mathbf{f}_u^i$, $\mathbf{f}_v^i$ should satisfy the static friction law (c.f., Section 3 in [Whiting et al. 2009]). Note that the cuboid weights are parameterized in terms of the cuboid dimensions. In order to account for inaccuracies, we seek a least norm solution to Equation 2 as:

$$\mathbf{E}_s(\mathbf{A}) := \min_{\mathbf{f}} \|\mathbf{D}\mathbf{f} + \mathbf{w}\|^2 \; s.t. \; \mathbf{f}_n^i \geq 0 \text{ and } |\mathbf{f}_u^i|, |\mathbf{f}_v^i| < \mu \mathbf{f}_n^i, \tag{3}$$

where $\mathbf{f}_n^i \geq 0$ indicates the compression force constraint, and $\mu$ is the static friction coefficient, default 0.7 in our system. If the minimized energy $\mathbf{E}_s$ of Equation 3 is below a certain threshold (1e-3 in our tests), we deem the cuboid arrangement with the assigned interaction type information to be potentially stable, and further refine the cuboid dimensions as described next.

### 4.3 Refining Cuboid Parameters under Contact Constraints

Finally, we refine the cuboid parameters to improve the stability score of the arrangement $\mathbf{A}$, while preserving the connectivity information encoded by the contact graph extracted in the Section 4.2. We improve the current stability score using gradient descent. We numerically estimate gradient of the energy with respect to each current cuboid $B_i$, and greedily select the direction with the maximum norm, i.e., $\text{argmax}_i \, \|\nabla \mathbf{E}_s(\mathbf{A})|_{\gamma_{B_i}}\|$ evaluated at the current configuration $\gamma_{B_i}$ of the cuboids $B_i$. Note that we only vary parameters for one cuboid at a time. We take a small step along the negative of the normalized direction with all the refinements for the other cuboid parameters set to zero. We denote this full configuration vector as $\Gamma_{\mathbf{A}}$. We project the solution to preserve the active interactions as specified by the corresponding contact graph. Specifically,

$$\alpha^* := \underset{\alpha}{\text{argmin}} \|\mathbf{E}_s(\mathbf{A} - \alpha\Gamma_{\mathbf{A}})\|^2 \text{ s.t. } f_i(X) = 0 \;\; \forall e_i \in \tilde{E}, \tag{4}$$

where $\tilde{E}$ denotes the *active* edge set in the current contact graph $G$, and $f_i$ denotes the contact constraints (e.g., the corresponding cuboids touch) as indicated by the corresponding contact type thus resulting in a QP formulation. For example, if two cuboids touch, we add a constraint to ensure that the corresponding interface faces remain touching. At the end of each iteration, we update the cuboid parameters as $\mathbf{A} \leftarrow \mathbf{A} - \alpha^*\Gamma_{\mathbf{A}}$. We refine the parameters for each plausibly stable arrangement. In case of multiple solutions having same number of fixed contacts, we pick the one with the smallest overall extension volume. For example, in Figure 8 the bottom-right solution is selected.

## 5 Evaluation and Results

We processed synthetic and real scenes from recorded and public datasets in order to measure the quality of the optimized proxy approximation. Both the connectivity and geometry of their arrangements were evaluated (when suitable groundtruth was available). We calculated precision, recall, and f-measure for the connectivity validation, and used an $L_1$ norm based metric to evaluate the improvement of the proxy dimensions.

**Datasets:** We tested our algorithm on four categories of scenes: (i) synthetic scan simulations allowed us to evaluate robustness against sampling variations, initialization quality, and increasing amount of occlusion due to changing view-points; (ii) 700+ scenes (with available groundtruth) were evaluated from the NYU2 dataset [Silberman et al. 2012]; (iii) a variety of indoor scenes with different levels of occlusion and complexity were recorded to evaluate retrieval capabilities in increasingly occluded environments; and (iv) Kinect Fusion type scenes were recorded to show the increasing importance of our solution when the scene complexity increases. We also assessed the suitability of the recovered cuboid-based structures for scene completion and manipulation tasks (see Section 6).

**Evaluation metrics and ground truth.** We evaluated how well we can recover the structure information in the scenes, and how much this improves the approximation accuracy of the recorded objects. The structure information was encoded in a support graph similar to [Silberman et al. 2012]. We compared the initial and optimized structure graph to a ground truth graph. Our optimized graph, as described in Algorithm 1, contains edges between two proxies labelled "touch" and "fixed." The ground truth graph was created manually (or automatically in case of scenes from NYU2) to encode the correct relationships using these labels. Walls, floors, ceilings were assumed to be objects with zero mass. Hanging relationships were labelled "fixed." The initial graph was assumed to contain solely "touching" edges. Finally, we calculated initial and optimized precision-recall (PR) ratios, and converted them to f-measure ($F_1 = \frac{200PR}{P+R}$). A summary of our results can be found in Table 1.

For those scenes where we have access in the real world (to perform measurements by removing occlusion), we evaluated the accuracy

**Table 1:** *Quality evaluation for structure graphs.*

| | | INITIAL | | OPTIMIZED | |
|---|---|---|---|---|---|
| TYPE | #Scenes | PR (%) | $F_1$ | PR (%) | $F_1$ |
| Synthetic | 12 | 42.1 / 26.7 | 32.2 | 96.2 / 84.7 | 87.2 |
| NYU2 | 700 | 34.7 / 47.3 | 40.0 | 55.3 / 70.8 | 60.5 |
| Recorded | 20 | 70.1 / 49.5 | 56.5 | 95.7 / 96.6 | 95.9 |
| Kinfu | 2 | 37.2 / 28.2 | 32.1 | 96.4 / 84.0 | 89.7 |

gain of the proxy approximation. We compared the $L_1$ norms of the proxy extents compared to the ground truth both in the initial and optimized scenes (Equation 6). The threshold $\epsilon = 5\%$ was used to filter changes that could have been induced by sensor noise. The error improvements were summed weighted by the ground truth proxies' proportion in the scene as:

$$e_i = \frac{|s_i - s_i^{(GT)}|}{s_i^{(GT)}} \; ; \; c_i = \begin{cases} s_i^{(GT)}, & \text{if } |e_i^{init} - e_i^{optimized}| > \epsilon. \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

$$Error(scene_k) = \frac{\sum_{j=1}^{\#proxies} \sum_{i=1}^{3} c_i \, |e_i^{init} - e_i^{optimized}|}{\sum_{i=1}^{3j} c_i}. \quad (6)$$

To collect the ground truth, we imagined the smallest bounding proxy of the real world objects and physically measured the three side lengths using measuring tape. We denoted these $s_i^{(GT)}$ and compared them to the measured side lengths in each setting ($init$, $optimized$), denoted by $s_i$ in Equation 5.

### 5.1 Synthetic Data

We modeled scenes (chair, stacking boxes, table with chairs) to evaluate the stability of the solution regarding the quality of initialization, arrangement complexity and change of viewpoint. The models were then virtually scanned using Kim et al. [2012] and then initial cuboids estimated as described in Section 4.1. We evaluated the quality of the structure graph compared to the ground truth using $F_1$ score (best being 100%).

**(i) Robustness to initialization.** In order to evaluate the robustness of our approach towards the quality of initialization, we synthetically perturbed the parameters of the initial cuboids (within $\pm 5\%$ of original size, and within $15^\circ$ of original orientation). We discard any perturbation if it results in intersecting cuboids. We then used such arrangements as input to our core algorithm and got consistent results with respect to ground truth structure graphs over several runs (about 20) for each of the bookshelf and chair scenes (Table 2).

We also tested the robustness towards random perturbation of proxy densities. We changed the densities of up to 20% of the cuboids selected randomly, each up to $\pm$5x. We found the desks and sofas to be more fragile allowing 2-3x changes, while other scenes (aligned box stacks) to be more robust (5-7x variations) over 20 runs. Note that we switched off image-based refinement for these experiments as due to the perturbation the recovered proxies might not be containing their generating points.

**(ii) Robustness to change of viewpoint.** We evaluated the effect of varying occlusion due to different view-points by re-sampling the top-right scene in Figure 12 from different views. The most significant challenge was due to the lack of an initial proxy when no corresponding point samples was recorded. This occurs when an object part is entirely occluded in the scene. Some errors are inherited from the decaying sensor accuracy when targeting surfaces from skew view-angles.

**Table 2:** *$F_1$ scores (initial $\rightarrow$ optimized) of optimization after perturbation of initialization. The initialized proxies were randomly rotated around their centroids.*

| scene | $0^\circ$ | $5^\circ$ | $10^\circ$ | $15^\circ$ |
|---|---|---|---|---|
| chair_01 | 20.0→100 | 20.51→100 | 20.51→97.8 | 17.14→87.8 |
| chair_02 | 21.43→100 | 20.69→100 | 14.29→97.5 | 14.29→94.7 |

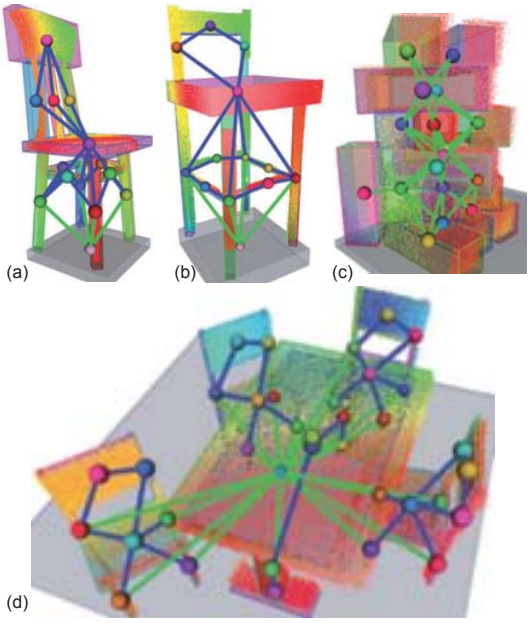



**Figure 12:** *Synthetic scenes created and sampled to evaluate stability. Robustness to perturbation of initialization (a,b), arrangement complexity (c) and change of view-point (d) were measured.*

In total we simulated 5 different viewpoints, the best $F_1$ score improvement was $25.5 \rightarrow 90$. Our method has two solutions to resolve inconsistencies. If enough redundancies were present, some proxies (legs of chairs and tables) can be hallucinated based on symmetry. In other cases an existing proxy is extended to make the scene stable (Figure 8). This minimally resulted in one false positive and two false negative edges in the structure graph.

**(iii) Robustness to arrangement complexity.** We evaluated how well our method handles densely connected structure graphs. Synthetic scenes with 6, 9 and 18 stacking boxes were scanned and optimized. Additionally the connectivity of real recordings (Figures 14 and 16) were also compared in Table 3. The 90+ $F_1$ scores show that our method can successfully handle both densely connected structure graphs (median valence 5 in Figure 13) and large number of proxies (56 in Figure 14).

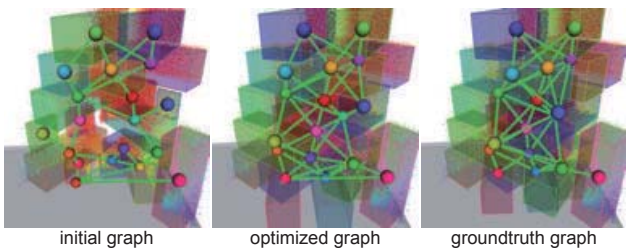



**Figure 13:** *Robustness to arrangement complexity. A scene with 18 proxies was reconstructed with $F_1$ score 90.24. The structure graph has 41 edges, median valence is 5. Some of the edges are missed because of the minimal extension principle in areas of occlusion.*

**Table 3:** *Robustness to arrangement complexity.*

| | Fig. 13 | Fig. 16 | Fig. 14 |
|---|---|---|---|
| # proxies | 18 | 17 | 56 |
| # contacts | 41 | 29 | 64 |
| max valence | 6 | 11 | 11 |
| median valence | 5 | 2 | 2 |
| average valence | 4.55 | 3.35 | 2.32 |
| min valence | 1 | 1 | 1 |
| $PR(\%)$ init. | 85.19 / 56.10 | 58.82 / 34.48 | 15.56 / 10.77 |
| $PR(\%)$ optim. | 90.24 / 90.24 | 100.0 / 89.66 | 98.33 / 90.77 |
| $F_1$ init. | 67.65 | 43.48 | 12.73 |
| $F_1$ optim. | 90.24 | 94.55 | 94.40 |

### 5.2 Performance on NYU2 Dataset

We tested our algorithm on a wide range of scenes (700+) from the NYU2 dataset. For initialization, we split each depth image into components based on the available segmentation masks, and fitted initial cuboids as described in Section 4.1 aligned to the ground plane. A large fraction of the scenes contain simple relationship hierarchies, such as object-on-floor or object-on-object-on-floor. A small ratio of these objects have incorrect structure due to occlusion. Stacked objects, when present, were vertically aligned, making the cuboid-based completion rather simple. We left out scenes with transparent objects.

**Ground truth structure graph.** We evaluated the correctness of our scene analysis by comparing the optimized scene graphs to the ground truth support graphs using updated versions of [Guo and Hoiem 2013]. These graphs are manually created, and contain semantic information (some proxies are much larger then their pointclouds, since a certain type of furniture and its size is assumed). Some of the proxies come with the labels "floor," "wall," or "ceiling." Further, several scenes have multiple floor regions (from NYU2) but only one floor proxy. This produces ambiguous correspondence. For the remaining scenes we extracted the support graph from the NYU2 dataset and matched it to the ground truth proxies. We used the resulting support graphs as ground truth.

**Automatic evaluation.** The ground truth graphs automatically retrievable with the method above contained 40%-85% of the edges a manual annotator would insert, influencing the scores reported in Table 1. Our algorithm was designed to resolve proxy intersections, which assumes a constant approximation scale and no "inclusive" relationships. We pruned those "touching" edges both from the initial and optimized structure graphs before evaluation, resulting in higher precision but lower recall scores. (All the results are provided as part of supplementary for visual inspection.)

### 5.3 Cluttered Indoor Scenes

We obtained single view Microsoft Kinect® scans of several (20+) test scenes with increasing complexity. These scenes were focused on situations with larger amount of missing information compared to the NYU2 dataset. Due to occlusion, recovering the structure graph and geometry posed a significantly higher challenge. The objects in these scenes are often locally unstable, but globally stable requiring a global analysis. For example, see the multiple cuboids in case of the sofa objects. Recorded scenes contained: vertically stacked box arrangements, slanted boxes, non-cuboid objects, single/multiple pieces of office furniture; and scenes with significant clutter and cyclic support relationships (Figure 16). The scenes had 2-23 (average 8) objects and 2-29 (average 10) touching and fixed contacts. In addition to estimating the validity of the structure graphs (Table 1) we evaluated the correctness of the geometric information recovered by physically examining the scenes and comparing them to the optimized outputs. The comparison showed that our method improved the proxy approximation from 65% to 97% averaged over all the recorded scenes of varying complexity. This means that where our method made significant (>5%) changes to the scene geometry, we reduced the approximation error by 32% on average (see Figure 17 and supplementary materials for details).

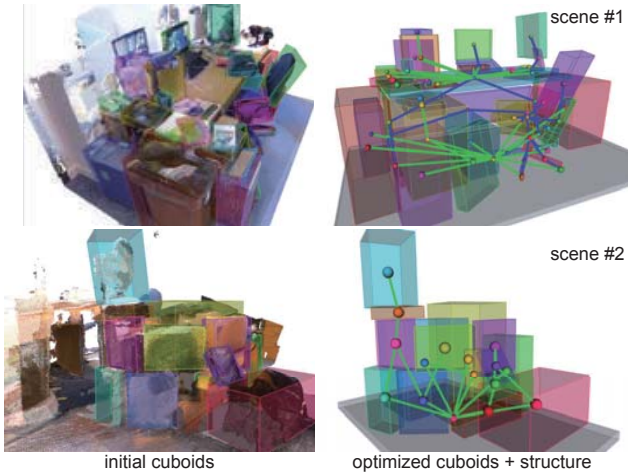



**Figure 14:** *Initial and optimized cuboids with extracted structure for two multi-view recordings of cluttered scenes (see supplementary material for details).*

### 5.4 Multi-view Recordings

In order to investigate, whether the addressed problems of occlusion can be resolved by easily accessible 3D reconstruction systems, we captured 2 scenes (∼1000 RGBD frames each) from as many viewpoints as accessible. We used PCL's Kinfu LargeScale implementation. The experiments show that even when such an involved reconstruction system is at hand, the main problems persist. Parts of a convoluted scene can be occluded from every view-point or it can be hard, impractical or impossible to make a recording from view-points that would reveal the necessary structure details of the scene. Creating the ground truth support graph for a scene containing 50+ objects takes ∼1 hour by disassembling the scene for physical measurements. Our algorithm converged in 20 seconds up to 98.3% precision and 90.8% recall ($F_1$ : 94.4) (Figure 14).

### 5.5 Comparison

In a related approach, Jia et al. [2013] proposed a 3D volumetric reasoning algorithm for parsing RGBD images with 3D block unit-



**Figure 15:** *Output of Jia et al. [2013] on a scene with severe occlusion. Stable cuboids are shown in blue and unstable ones are shown in red. Given the ground truth segmentation, initial boxes are fitted after adjusting their orientation based on the supporting relations (left). In order to reach the physical stability, the algorithm merges neighboring segments during the optimization (middle). The optimized result is to merge all the regions to be stable (right).*

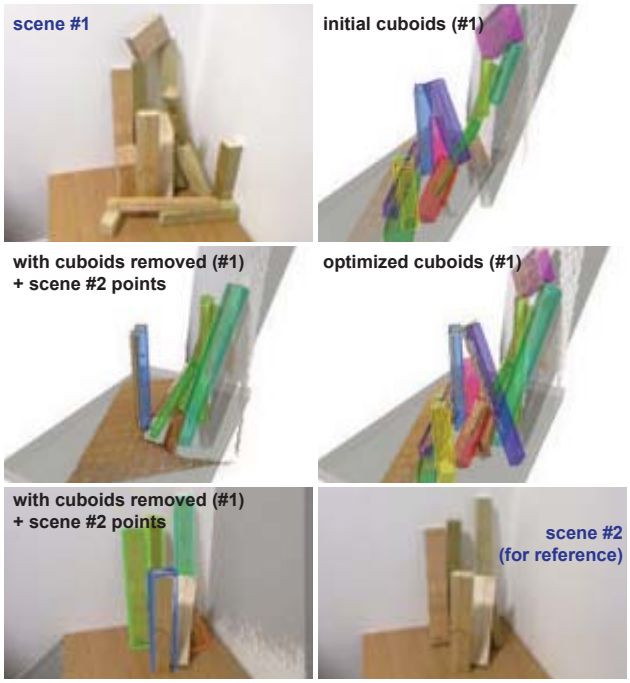



**Figure 16:** *Initial cuboids obtained from scene #1 are optimized to produce a physically stable cuboid arrangement. To show clearly how well the optimized boxes are aligned to groundtruth, we progressively remove some cuboids from the optimized cuboid arrangement (virtually) and compare with groundtruth obtained by physically removing the corresponding wooden blocks from the real world scene (scene #2) . Last row shows how well the hypothesized cuboids follow the groundtruth in occluded parts.*

s. At a high level, we both use stability considerations, but with different goals: we for *reasoning about unseen parts*, and they for image segmentation and interaction (support and stability). Hence, under severe occlusions as for heavily occluded parts, their algorithm does not perform very well. For example, see Figure 15. One interesting future direction is to take their output as an initialization to our system and hypothesize the missing parts of the scene.

### 5.6 Performance

**User interaction.** While for simple scenes automatic cuboid generation is sufficient, for more complex scenes the user may be required to delete spurious boxes or annotate scribbles to guide the initial cuboid generation. When necessary, user interaction took 10-30 secs, and proved to be simpler than annotating in 3D.

**Timing.** We tested our algorithm on single view RGBD images captured by a Microsoft Kinect camera. Once the initial cuboids were generated, the algorithm took 10-20 seconds to converge depending on the scene complexity (for scenes with $<$30 cuboids). All the results are included as supplementary material.

## 6 Applications

**(i) Structure-guided completion.** Once we have good-quality scan of a particular type object and its complete cuboid structure in a single view RGBD image, the structure-guided completion can help transfer the scan guided by its structure to complete the missing information of partial scans of same type objects.

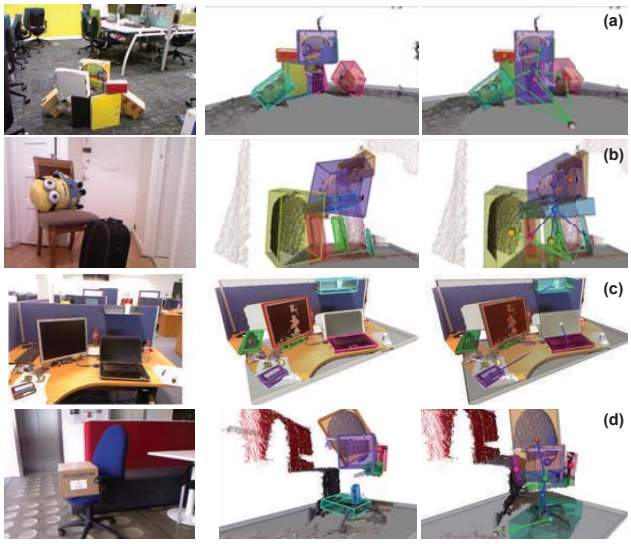



**Figure 17:** *Algorithm results on various scenes with occlusion and stacking. Ground planes (or desktop) are shown in gray. Scribbles were used for the slanted objects; rest of the algorithm runs automatically. Left-to-right: Input RGBD scan; initial cuboids and optimized cuboids along with extracted contact graphs (back view).*

The completion is done by registering the good-quality scan to partial scans assisted by the cuboid structure. Let $\mathbf{A}^1 = \{B_i^1, i = 1, \ldots n\}$ denote the cuboid structure of the good-quality scan and $\mathbf{A}^2 = \{B_j^2, j = 1, \ldots m\}$ the extracted cuboids for the partial scan. We established a correspondence between the two cuboid arrangements using a spectral matching approach proposed in [Leordeanu and Hebert 2005] by comparing cuboid dimensions in part-level comparisons. Assuming all the objects are on the ground plane, we prune the edge pairs when their height and length differences are larger than 0.05m and 0.1m, respectively. Afterwards, for two candidate corresponding cube edges, we project them to the ground plane and solve for the aligning rigid transform by assuming the correspondence of the projected end points of the two edges. The computed transformation is then refined via ICP using the 3D points. Figure1-right shows an example of scan completion using the sofa scan along with its extracted cuboids-based structure.

**(ii) Model retrieval.** Completed cuboid structures can also be used to retrieve high-quality 3D models for the corresponding low-quality point clouds. Instead of learning a mapping between local depth patches and 3D model labels [Shao et al. 2012], we directly employ the completed structure as a global context to assist shape matching. The model with highest matching score is selected as the most similar model. We first get an initial alignment between repository models, retrieved using object keyword search, and the target structure by aligning their upright orientation [Shen et al. 2012]. We assume repository models to be upright oriented. For the cuboid structure, the up direction is the same as the ground normal, and the right direction is selected from any horizontal edges of a randomly picked cuboid. The model is then translated and scaled to fit into the bounding box of the structure. Finally we sample rotation angles about the up axis of the scene to roughly estimate an alignment. Matching score is estimated based on model vertices that are covered by the cuboid structure. Specifically, matching score is defined as the sum of the matching ratio of covered vertices and the uniformly sampled points inside the cuboid structure, and the matching ratio of covered vertices and all the model vertices. We

refine the alignment between the point clouds and the 3D model using ICP, and use the model with the best alignment score. Figure 18-right shows three retrieval results.

**(iii) Image manipulation.** RGB images corresponding to the original depth scans can be edited using the abstracted structure (i.e., ground, wall planes and physically stable cuboid arrangement). In each case, we created a background image by identifying and deleting image pixels corresponding to the points in $P$ (in 3D) that fall inside or are close to the completed cuboids. We ray-cast the retrieved object models, which are interactively re-positioned (by moving on the ground plane), to create the foreground layer including the shadow map on the ground plane (default light from above). Holes in the background layer were completed using PatchMatch and composited with the rendered foreground layer. Note that the original camera view is used for such edits. Figure 18-bottom row shows few examples.




**Figure 18:** *Based on single view RGBD scans (left), our algorithm extracts a set of stable arrangement of cuboids (middle). The arrangement is then used to retrieve matching 3D meshes with cuboid decomposition and used for scene modeling (right). (Bottom row) Retrieved models can be repositioned, rendered, and combined with the original RGB to synthesize novel manipulated images.*

## 7 Conclusion

We presented an algorithm to discover coarse structure from partial scans of cluttered scenes with significant occlusion. Structure is captured in the form of abstracted cuboids and how they are mutually arranged. Starting from an initial set of cuboids, semi-automatically created, the algorithm proposes several possible extensions into the occluded regions and selects the one that is physically stable and requires minimal extension. We evaluated our framework on scenarios involving cluttered scenes with stacked and piled objects, and utilized the recovered structure for applications including scan completion and scene understanding.

*Limitations and future work:* Since we assume objects to have a fixed density, the algorithm can fail when this assumption is violated by large imbalance among object densities. One possibility would be to assign different densities based on image-based attribute classification. A natural extension of this work would be to consider other types of primitives (e.g., cylinders, spheres). Finally, the output of our algorithm being physically plausible can directly be integrated with physics-aware image manipulation and simulation systems resulting in interesting and non-trivial mix of real/virtual objects and their interactions.

## Acknowledgements

We thank the reviewers for their comments and suggestions for improving the paper. This work was supported in part by an UCL Impact award, the ERC Starting Grant SmartGeometry (StG-2013-335373), NSFC (No. 61402402), and gifts from Adobe Research.

## References

ANAND, A., KOPPULA, H. S., JOACHIMS, T., AND SAXENA, A. 2011. Contextually guided semantic labeling and search for 3d point clouds. *CoRR abs/1111.5358*.

ARIKAN, M., SCHWÄRZLER, M., FLÖRY, S., WIMMER, M., AND MAIERHOFER, S. 2013. O-snap: Optimization-based snapping for modeling architecture. *ACM TOG 32*, 1, 6:1–6:15.

BAO, Y., CHANDRAKER, M., LIN, Y., AND SAVARESE, S. 2013. Dense object reconstruction using semantic priors. In *IEEE CVPR*.

BLUM, M., GRIFFITH, A., AND NEUMANN, B., 1970. A stability test for configurations of blocks. TR-AI Memo.

DAVIS, J., MARSCHNER, S. R., GARR, M., AND LEVOY, M. 2002. Filling holes in complex surfaces using volumetric diffusion. In *Proc. 3DPVT*.

FIDLER, S., DICKINSON, S., AND URTASUN, R. 2012. 3d object detection and viewpoint estimation with a deformable 3d cuboid model. In *NIPS*, 620–628.

GIBSON, J. 1986. *The Ecological Approach To Visual Perception [Paperback]*. Pschology Press.

GUO, R., AND HOIEM, D. 2013. Support surface prediction in indoor scenes. *IEEE ICCV*.

GUPTA, A., EFROS, A. A., AND HEBERT, M. 2010. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*.

GUPTA, A., SATKIN, S., EFROS, A. A., AND HEBERT, M. 2011. From 3D scene geometry to human workspace. In *CVPR*, 1961–1968.

HARARY, G., TAL, A., AND GRINSPUN, E. 2013. Context-based coherent surface completion. *ACM TOG*.

HARTLEY, E., KERMGARD, B., FRIED, D., BOWDISH, J., PERO, L. D., AND BARNARD, K. 2012. Bayesian geometric modeling of indoor scenes. *IEEE CVPR*, 2719–2726.

HEDAU, V., HOIEM, D., AND FORSYTH, D. 2010. Thinking inside the box: using appearance models and context based on room geometry. In *ECCV*, 224–237.

HEDAU, V. 2012. Recovering free space of indoor scenes from a single image. In *IEEE CVPR*, 2807–2814.

JIA, Z., GALLAGHER, A., SAXENA, A., AND CHEN, T. 2013. 3d-based reasoning with blocks, support, and stability. In *IEEE CVPR*, 1–8.

JIANG, H., AND XIAO, J. 2013. A linear approach to matching cuboids in rgbd images. In *IEEE CVPR*.

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*, 6, 138:1–138:11.

KIM, B.-S., KOHLI, P., AND SAVARESE, S. 2013. 3D scene understanding by Voxel-CRF. In *Proceedings of the International Conference on Computer Vision*.

KIM, Y. M., MITRA, N. J., HUANG, Q., AND GUIBAS, L. 2013. Guided real-time scanning of indoor objects. *Computer Graphics Forum (Proc. Pacific Graphics) 32*, 177–186.

KOPPULA, H., ANAND, A., JOACHIMS, T., AND SAXENA, A. 2011. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*.

LAFARGE, F., AND ALLIEZ, P. 2013. Surface reconstruction through point set structuring. *CGF*.

LEE, D. C., GUPTA, A., HEBERT, M., AND KANADE, T. 2010. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *NIPS*, vol. 24.

LEORDEANU, M., AND HEBERT, M. 2005. A spectral technique for correspondence problems using pairwise constraints. In *IEEE ICCV*, 1482–1489.

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 SIGGRAPH 30*, 4.

MATTAUSCH, O., PANOZZO, D., MURA, C., SORKINE-HORNUNG, O., AND PAJAROLA, R. 2014. Object detection and classification from large-scale cluttered indoor scans. *CGF Eurographics*.

NAN, L., XIE, K., AND SHARF, A. 2012. A search-classify approach for cluttered indoor scene understanding. *ACM SIGGRAPH Asia 31*, 6, 137:1–137:10.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. 2005. Example-based 3d scan completion. In *SGP*, 23–32.

ROTHER, C., KOLMOGOROV, V., AND BLAKE, A. 2004. "grabcut": interactive foreground extraction using iterated graph cuts. *ACM SIGGRAPH 23*, 3, 309–314.

SCHLECHT, J., AND BARNARD, K. 2009. Learning models of object structure. In *NIPS*.

SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient ransac for point-cloud shape detection. *CGF 26*, 2, 214–226.

SHAO, T., XU, W., ZHOU, K., WANG, J., LI, D., AND GUO, B. 2012. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM SIGGRAPH Asia 31*, 6, 136:1–136:11.

SHARF, A., ALEXA, M., AND COHEN-OR, D. 2004. Context-based surface completion. In *ACM SIGGRAPH*, 878–887.

SHEN, C.-H., FU, H., CHEN, K., AND HU, S.-M. 2012. Structure recovery by part assembly. *ACM SIGGRAPH Asia 31*, 6, 180:1–180:11.

SILBERMAN, N., AND FERGUS, R. 2011. Indoor scene segmentation using a structured light sensor. In *Proc. ICCV - Workshop on 3D Representation and Recognition*, 601 – 608.

SILBERMAN, N., HOIEM, D., KOHLI, P., AND FERGUS, R. 2012. Indoor segmentation and support inference from rgbd images. In *ECCV*.

UMETANI, N., IGARASHI, T., AND MITRA, N. J. 2012. Guided exploration of physically valid shapes for furniture design. *ACM SIGGRAPH 31*, 4, 86:1–86:11.

WHITING, E., OCHSENDORF, J., AND DURAND, F. 2009. Procedural modeling of structurally-sound masonry buildings. *ACM Trans. Graph. 28*, 5 (Dec.), 112:1–112:9.

XIAO, J., RUSSELL, B., AND TORRALBA, A. 2012. Localizing 3D cuboids in single-view images. In *NIPS*. 755–763.

XIONG, X., AND HUBER, D. 2010. Using context to create semantic 3d models of indoor environments. In *BMVC*, 1–11.

ZHENG, Q., SHARF, A., WAN, G., LI, Y., MITRA, N. J., COHEN-OR, D., AND CHEN, B. 2010. Non-local scan consolidation for 3d urban scenes. *ACM SIGGRAPH 29*, 4, 94:1–94:9.

ZHENG, Y., CHEN, X., CHENG, M.-M., ZHOU, K., HU, S.-M., AND MITRA, N. J. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM SIGGRAPH 31*, 4, 99:1–99:11.

ZHENG, B., ZHAO, Y., YU, J. C., IKEUCHI, K., AND ZHU, S.-C. 2013. Beyond point clouds: Scene understanding by reasoning geometry and physics. In *IEEE CVPR*.