SEETHROUGH: Finding Objects in Heavily Occluded Indoor Scene Images

Moos Hueting
University College London

Pradyumna Reddy
University College London

Vladimir Kim
Adobe Systems

Ersin Yumer
Adobe Systems

Nathan Carr
Adobe Systems

Niloy J. Mitra
University College London

Abstract

Discovering 3D arrangements of objects from single indoor images is important given its many applications such as interior design and content creation for virtual environments. Although heavily researched in the recent years, existing approaches break down under medium to heavy occlusion as the core image-space region detection module fails in absence of directly visible cues. Instead, we take into account holistic contextual 3D information, exploiting the fact that objects in indoor scenes co-occur mostly in typical configurations. First, we use a neural network trained on real indoor annotated images to extract 2D keypoints, and feed them to a 3D candidate object generation stage. Then, we solve a global selection problem among these candidates using pairwise co-occurrence statistics discovered from a large 3D scene database. We iterate the process allowing for candidates with low keypoint response to be incrementally detected based on the location of the already discovered nearby objects. We demonstrate significant performance improvement over combinations of state-of-the-art methods, especially for scenes with moderately to severely occluded objects. Code and data available at `http://geometry.cs.ucl.ac.uk/projects/2018/seethrough`.

1. Introduction

For many scene understanding tasks such as creating a room mockup for VR or automatically estimating how many people a room can accommodate, it is sufficient to estimate positions, orientations, and rough proportions of the objects rather than exact point-wise surface geometry. Given a *single* 2D photograph, the goal of this paper is to select and place instances of 3D models, particularly the partially *occluded* ones, to recover the photographed *scene arrangement* or *layout* [22] under the estimated camera.

With easy access to large volumes of image and 3D model repositories and the availability of powerful supervised learning methods, researchers have investigated multiple subproblems relevant to the above goal, such as object recognition [17], localization [33], pose prediction [38], or developed a complete system IM2CAD [23] that selects and positions 3D CAD models that are similar to the input imaged scenes. While these approaches work reliably in rooms with relatively low occlusion, under moderate to heavy occlusion the methods quickly deteriorate. A common source of failure is that under significant occlusion, state-of-the-art semantic segmentation or region detection begins to break down, and hence any system relying on them also fail (see Figure 1).

Unlike images with limited occlusion where direct image-space information is sufficient, occluded scenes require a different treatment. One possibility is to train an end-to-end network to go from single images to parameterized scene mockups. On the one hand, in our experiments the networks trained with synthetic 3D scene data do not easily translate to real-world data. On the other hand, ob-



Figure 1. We present SEETHROUGH to detect objects (chairs, tables, cabinets) from single images under medium to heavy occlusion by reasoning with 3D scene-level context information and significantly improve detection rate over state-of-the-art alternatives.

taining real-world training data is difficult to scale as it requires complex annotations in 3D from single images. We propose an approach that heavily relies on 3D contextual statistics automatically extracted from synthetic scenes.

Our key insight is that typical indoor scenes exhibit significant regularity in terms of co-occurrence of objects, which can be exploited as explicit priors to make predictions about object identity, placement and orientation, even under significant inter- or intra-object occlusions. For example, a human observer can easily spot heavily occluded chairs due to the presence of other visible nearby chairs and a table, as we have a good mental model of typical chair-table arrangements.

We introduce SEETHROUGH that generates 2D keypoints from input images using a neural network, lifts the keypoints to candidate 3D object proposals, and then solves a selection problem to pick objects scored according to object cooccurrence statistics extracted from a scene database. We iterate the process by allowing already selected objects to reinforce selection of weakly witnessed occluded ones. The main conceptual novelty is combining deep learning for keypoint detection and graphical model for handling object context information. In other words, although objects are largely occluded, even partial and contextual evidence can be used for candidate generation which can in turn be pruned using object arrangement priors.

We tested our approach quantitatively on a new scene mockup dataset including partially occluded objects and show significant improvement of recognition over baseline methods on multiple quantitative measures. Although our current implementation is focused on few classes (chairs, tables, cabinets, bookshelves), the method can be retrained to other classes with appropriately annotated data.

2. Related Work

Scene mockups. 3D scene inference from 2D indoor images has recently received significant research focus due to the ubiquity of the new generation capture methods that enable partial 3D and/or depth capture. A significant amount of progress has been made following the early work of Hoeim et al. [20], first with approximating only room shape [11, 29, 27, 18], then inferring cuboid-like structures as surrogate furniture [12, 9, 41, 40, 34] and performing scene space reasoning using context models [8, 5]. However, for detailed geometry prediction, the image input is generally supplemented with additional per pixel depth or point clouds [25]. Mattausch et al. [30] used 3D point cloud input to identify repeated objects by clustering similar patches. Li et al. [26] utilize an RGB-D sensor to scan an environment in real time, and use the depth input to detect 3D objects queried from a database. In contrast, our method works only on single RGB images.

Recently, Izadinia et al. [23] presented the IM2CAD system for scene reconstruction with CAD models from a single image using image based object detection (using FR-CNN) and pose estimation approaches. Although their objective is similar to ours, the performance is bounded by the individual vision algorithms utilized in their pipeline. For example, when the segmentation misses an object because of significant occlusion, there is no mechanism to recover it in the reconstruction.

3D→2D alignment. Another way to create scene mockups is by directly fitting 3D models to the image. Pose estimation work [38, 36, 21, 27, 24, 4] also demonstrated that given object images, reliable 3D orientation can be predicted, which in turn might help with scene mockups. Lin et al. [28] used local image statistics along with image-space features to align a given furniture model to an image. Aubry et al. [4] utilized a discriminative visual element processing step for each shape in a 3D model database, which is then used to localize and align models to given 2D photographs of indoor scenes. Like most existing methods, their approach breaks down under moderate to high occlusion. Our method performs better, as other nearby objects can provide higher order information to fill in the lost information.

Priors for scene reconstruction. Scene arrangement priors have been successfully demonstrated in 3D reconstruction from unstructured 3D input, as well as scene synthesis [14]. Shao et al. [35] demonstrated that scenes with significant occlusion can be reconstructed from depth images by reasoning about the physical plausibility of object placements. Monszpart et al. [31] uses the insight that planar patches in indoor scenes are often oriented in a sparse set of directions to regularize the process of 3D reconstruction. On the other hand, based on priors between humans, Fisher et al. [15] leveraged human activity priors together with object relationships as a foundation for 3D scene synthesis. In contrast to the complex and high order joint relationships used in these works, our object centric templates are compact and primarily encode the repetition of similar shapes (such as two side by side chairs) across pose and location.

3. Overview

In indoor scenes with many objects, we observe that the environment is not important for the recognition of the unoccluded object – the shape of the object is clearly visible and immediately recognizable. However, under occlusion, the task of recognizing the object necessitates adding 3D contextual information. State-of-the-art methods based on FRCNN [33] correctly detect objects that are visible, but miss partially occluded ones (see inset figure in Section 2). However, under occlusion, the task of recognition becomes easier with more contextual and cooccurence information.

Motivated by the above insight, we design SEETHROUGH to run in three key steps: (i) an image-space

keypoint detection trained on AMT-annotated real photographs (Section 4.1); (ii) a candidate generation step that takes the estimated camera to lift detected 2D keypoints to 3D (deformable) model candidates (Section 4.2); and (iii) an iterative scene mockup stage where we solve a selection problem on a graphical model to extract a scene arrangement that proposes a plausible object layout using a common object co-occurrence prior (Section 4.3).

4. Method

4.1. Keypoint Detection

At this stage our goal is to detect subtle cues for potential object placements in a form of keypoints. A *keypoint* is a salient 3D point that appears across all objects of the same class (e.g., tip of a chair leg). We expect that a small number of (projected) keypoints will still be visible even under severe occlusions, and be useful in creating reasonable hypothesis for potential object placement. We represent this signal as: first, a *keypoint map*, a per-pixel function that indicates how likely a particular keypoint is to occur at that pixel (each keypoint has a separate map m_i), and second, *keypoint locations* which define the 2D coordinates for each keypoint. Both sets of information are used at different stages of our algorithm. We collected our own training data and trained a convolutional neural network to detect a continuous keypoint probability function, which we further use to extract candidate keypoint locations.

For each object, we picked N_k keypoints (8 for tables and cabinets each; 10 for chairs and sofas) and fine-tuned a variant of ResNet-50 neural network [17] to predict these keypoint maps in N_k output channels (see supplemental material for architecture details). We also tested the CPM architecture [37], but it yielded slightly inferior performance. While the latter focuses on keypoint detection it was pre-trained on human poses rather than general images, which is why we believe CPM did not generalized as well to our particular task (see supplemental material).

The above network predicts continuous keypoint maps $\mathbb{M} := \{m_1, ..., m_{N_k}\}$, and to extract the final keypoint locations (2D positions in the image) we used local maxima above a threshold τ_m (Figure 2). We denote the set of these keypoint locations by $\mathbb{Q} := \{Q_1, \ldots, Q_{N_k}\}$.

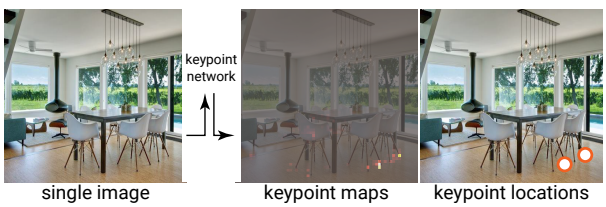


Figure 2. We trained individual neural networks on images to detect *keypoint maps*, which are then converted to 2D *keypoint locations*/thresholding and non-maximal suppression.

4.2. Candidate Object Detection

The goal of this step is to propose multiple candidate objects based on the detected keypoints. While we do not know how to group points, we observe that a very small number of keypoints (as few as two) belonging to the same object, provide enough constraints to infer the scale and the orientation of a proxy 3D object. Hence, we can generate multiple candidates even with a sparse signal under moderate to high levels of occlusions. Using these generated candidates, we can recast the global inference problem as a discrete graph optimization problem, where we only need to solve for indicator variables, selecting a subset of candidates. Thus, we want higher recall at the expense of lower precision in this step. Furthermore, in order to incorporate a slightly bigger context than a single keypoint, we select subsets of points that can compose an object. At training time we learn a deformable template from a database of 3D models, and at test time we optimize the fitting of these templates to various subsets of keypoints.

Object template. Given a database of consistently aligned 3D models M with manually labeled keypoints we use Principal Component Analysis (PCA) to project 3D coordinates of keypoints to a lower-dimensional space (we take eigenvectors $\lambda_1, ..., \lambda_k$ that explain $> 85\%$ of the variance). Our template is parameterized by a linear combination of these eigenvalues with weights $p = [p_1, ..., p_k]$ (representing offset from the mean λ_0). The final object template is defined by a weighted linear combination of the eigenvectors: $T(p) := \lambda_0 + \sum_i p_i \lambda_i$. Note that we build such models for each individual object class, but do not model the parameter correlation across class.

We formulate an optimization problem where we solve for object parameters (i.e., p) while making sure that the object aligns with the detected keypoints. To relate our 3D deformable model to 2D images, we need a camera estimate. We use a variant of Hedau et al. [19] to estimate a rotation matrix C_R with respect to the ground plane, the focal length C_f, and define the camera's location C_t to be at eye height (1.8m) above the world origin, giving camera parameters $C := [C_R, C_f, C_t]$. For each object we solve for a 2D translation across the ground plane t, azimuth θ, scale s, and 3D object template parameters p. Hence, the reprojection z_i of the i-th keypoint to image space:

$$z_i := \Pi_C \left(R_{\text{up}}(\theta)\ s\ k_i(p) + t \right), \tag{1}$$

where $k_i(p) = [T(p)]_i$ is a keypoint on the deformed template, R_{up} is a rotation around the up vector, and Π_C is a projection to the camera space.

As described next, we fit our template object in two stages: first, we propose a candidate based on a pair of points, and then, we refine these candidate parameters with respect to all keypoint maps.

(i) Initial proposals. To propose initial object candidates we sample all pairs of detected keypoints. We use a pair because it gives the smallest set to sample that provides enough constraints to extract an initial guess for object translation, scale, and orientation. For each pair, we initialize as $t = 0, \theta = 0, s = 1, p = 0$, and optimize:

$$L_{\text{init}} = \sum_{i \in \{u,v\}} \|z_i - k_i\|^2 + \underbrace{\alpha_1 \|s-1\|^2 + \alpha_2 \|p\|^2}_{\text{regularizer } (L_{\text{reg}})}, \quad (2)$$

where α_1 and α_2 are respectively the weights balancing scale and deformable template parameters ($\alpha_1 = 1$ and $\alpha_2 = 1$ in our tests).

(ii) Parameter refinement. For each of the initial proposals extracted above, we refine the fitting. Specifically, instead of considering point-locations, we define our objective with respect to soft keypoint maps m_j, maximizing the probability of template corners to align with keypoints predicted by the neural network, i.e.,

$$L = \sum_{i \in \{1,\ldots,N_k\}} \|1 - m_i(z_i)\|^2 + L_{\text{reg}}, \quad (3)$$

with L_{reg} as defined in Equation 2. If $L < \tau_u$, we add the final parameters as a candidate placement to our candidate placement set O.

Selecting a 3D mesh. For the results presented in this paper we show 3D meshes rather than object templates. Particularly, we pick the closest 3D model from our database by projecting its keypoints into the object PCA space, finding the nearest neighbor of the deformed template, and finally deforming it using the optimized parameters p. For scenes with multiple instances of the same object, we pick a consistent 3D model to place (based on inferred object size attributes).

4.3. Scene Inference

We do not expect all individual objects selected as candidates to be in the scene, since they might overlap, or have inconsistent arrangement. First, we capture scene statistics obtained from a large scene dataset with a probabilistic model, and then use the model to formulate an alternating discrete and continuous optimization.

Learning scene model. We model higher level scene statistics via a graphical model where each object is a node and edges between pairs of nodes capture object-to-object co-occurrence relationships. We used a Gaussian Mixture Model (GMM) with N_m (set to 5 in our tests) mixture components to model relative orientation δ_θ and translation δ_t of pairs of objects from a very large synthetic scene dataset [42]. We only take into account objects that are within a distance $\delta_r = 1.5m$ from each other, reasoning that far-away objects have weaker relationships. We use Expectation-Maximization algorithm to fit the GMM and add a small bias (0.01) to the diagonal of the fitted covariance matrices since objects in the database are axis-aligned.

Graph optimization. We formulate a graph labeling problem to decide which of the candidate objects should be included in the scene mockup, denoted by indicator variable $\gamma_i \in \{0, 1\}$, where $\gamma_i = 1$ iff object O_i is included. We *minimize* the objective:

$$L_{\text{graph}} := \sum_i \gamma_i U_i + \sum_{i,j} \gamma_i \gamma_j P_{i,j}, \quad (4)$$

where U_i is a unary penalty for an included object, and $P_{i,j}$ is pairwise penalty for a pair of included objects. We define the unary energy by projecting object's keypoints to the image and convolving the resulting keypoint map with a Gaussian, following the same procedure we used to create ground truth keypoint maps. This provides a location map n. And we set:

$$U_i := -\text{logit}\left(\frac{\|n \odot m_i\|_F}{\|n \odot n\|_F}\right), \quad (5)$$

where $\|\cdot\|_F$ represents the Frobenius norm, $\odot$ represents the Hadamard product, and $\text{logit}(x) = \log(x/(1-x))$. Note that since we do not expect a single placement to explain the entire keypoint location map, we setup the score as a multiplicative one, with the value only being dependent on the agreement of the actual keypoints the placement exhibits. We define the pairwise energy using the GMM model learned from the scene dataset:

$$P_{i,j} := -\text{logit}\left(GMM(\delta_\theta^{i,j}, \delta_t^{i,j})\right), \quad (6)$$

where $\delta_\theta^{i,j}, \delta_t^{i,j}$ are the relative orientations and translation of the objects o_i, o_j. Finally, we solve for the indicator variables $\{\gamma_i\}$ using OpenGM [3] by converting the above formulation into a linear program and feeding it to CPLEX [1] to find the final set of selected objects.

Refined object fitting. After selecting the set of objects, the scene mockup is ready. However, we found that our scene priors can also improve the initial object fitting results. To achieve this, we add a term from our GMM model to the regularization term (L_{reg}) in object fitting. We go through all candidate objects and re-optimize their parameters, keeping the selected objects fixed. As noted by Olson et al. [32], the structure of the negative log-likelihood (NLL) of a GMM does not lend itself to non-linear least squares optimization. Instead, we approximate the NLL of the full GMM by considering it as a Max-Mixture, reducing the NLL to the weighted distance from the closest mixture mean. We define the Max-Mixture likelihood function

$$p_{\text{Max}}(\delta) = \max_i w_i N(\delta | \mu_i, \Sigma_i),$$

where $\delta = \begin{bmatrix} \delta_t \\ \delta_\theta \end{bmatrix}$ is the relative translation and orientation of the new candidate w.r.t. the already placed object, and w_k is the weight of the kth mixture in the model. We use the sum of negative log-likelihoods of these terms for all selected objects that are within a distance of δ_r to the refined candidate:

$$-\log(p_{\text{Max}}(\delta)) = \min_k \frac{1}{2}(\delta-\mu_k)^T\Sigma_k^{-1}(\delta-\mu_k)-\log(w_k\eta_k),$$

where $N(\mu, \Sigma)$ represents the normal distribution, and η_k is the Gaussian normalization factor for the kth mixture. At optimization time, during each step we find the mixture component k^* that minimizes this function, and then optimize w.r.t. the negative log likelihood of the Gaussian of that component alone, resulting in the following term to be added to the objective function L_{reg} (Equation 2):

$$\frac{1}{2}(\delta - \mu_{k^*})^T\Sigma_{k^*}^{-1}(\delta - \mu_{k^*}). \tag{7}$$

Refined selection. Refined candidates and objects selected for the mockup can help in placing additional objects that have subtler cues. Hence, we iterate between refined fitting and refined selection processes. In the refined selection, we assume that previously selected objects cannot be removed, and add the unary term to favor placing new candidates, i.e., for each candidate placement in the second iteration, we add an extra cost to U_i (Eq. 5) as:

$$-\sum_k \text{logit}(GMM(o_i, o_k^*)), \tag{8}$$

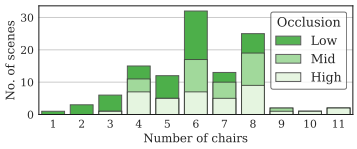
where $\{o_k^*\}$ are the objects selected at previous iterations.

5. Results and Discussion

5.1. Training and test data

(a) 2D keypoints on indoor images. We downloaded 5000 images from the HOUZZ website using keywords like living room, kitchen, dining room, meeting room, etc. We utilized the Amazon Mechanical Turk platform to obtain keypoints on the images requiring at least 3 workers to agree per image. For each image, we asked the turkers to mark the keypoints of the objects. Please refer to the supplemental for details about the web-based annotation interface. We convolved these keypoints with a Gaussian filter to simplify the CNN's task of learning smooth filters and averaged the results. These were used in addition to CAD models mockups available in the ObjectNet3D dataset [39], which we used to train keypoint detectors for sofa, cabinet, and table. We used 90,127 images containing 201,888 labeled objects for training the networks.

(b) Scene mockup groundtruth. In order to quantitatively measure the performance of SEETHROUGH and compare with alternate methods, we require a set of ground truth annotated scenes, i.e., images for which the relevant 3D CAD models have been placed manually. We are not aware of a similar dataset with mockups for 3D objects including the (partially) occluded ones. Hence, we setup another annotation tool in which an object can be placed by clicking and dragging, as well as by annotating a number of keypoints of the object, and optimizing for its location and scale. Moreover, objects can be copied and translated along their local coordinate axes, allowing for quick and precise annotation. We used the automatically estimated camera parameters for the automatic refinement, while discarding any image with grossly erroneous camera estimates. We used the tool to annotate 300 scenes (see inset for visibility statistics among randomly selected HOUZZ dataset). We found the NYU and SUN datasets to contain very limited occlusion and hence not suitable for our tests. The SUNCG dataset has more cluttered instances, but we found the scene statistics to be different compared to real-world scenes.



(c) 3D models and scenes. For our database models, we used the models from the ShapeNet [7] database and for scene statistics, we used 45K houses from the PBRS dataset [42]. While the latter comes with 400K physically-based renderings, we tried using these synthetic images to pretrain networks for predicting keypoint maps, but found that fine-tuning a variant of ResNet-50 with weights trained on ImageNet produced more accurate results.

5.2. Performance Measures and Parameters

Hyperparameters. Our optimization pipeline depends on a number of parameters that we optimized using HyperOpt [6]. We used the LOCANG measure as our objective measure. As ground truth data, we used 10 scenes fully annotated specifically for this purpose, in the same way as the data used for evaluation (see above).

Keypoint detection. We evaluated the accuracy of keypoint detection (see Figure 3) for each of the object classes. For objects with heavy occlusion, we report keypoint detection with 2 or more keypoints detected with a precision of 5 pixels or better (image size 256×256) since in those cases we have sufficient information to propose 3D candidates. For comparison, we evaluate state-of-the-art region detector FRCNN detection. Using confidence threshold of 0.5, RCNN boxes contained on 13.5% of keypoints from an

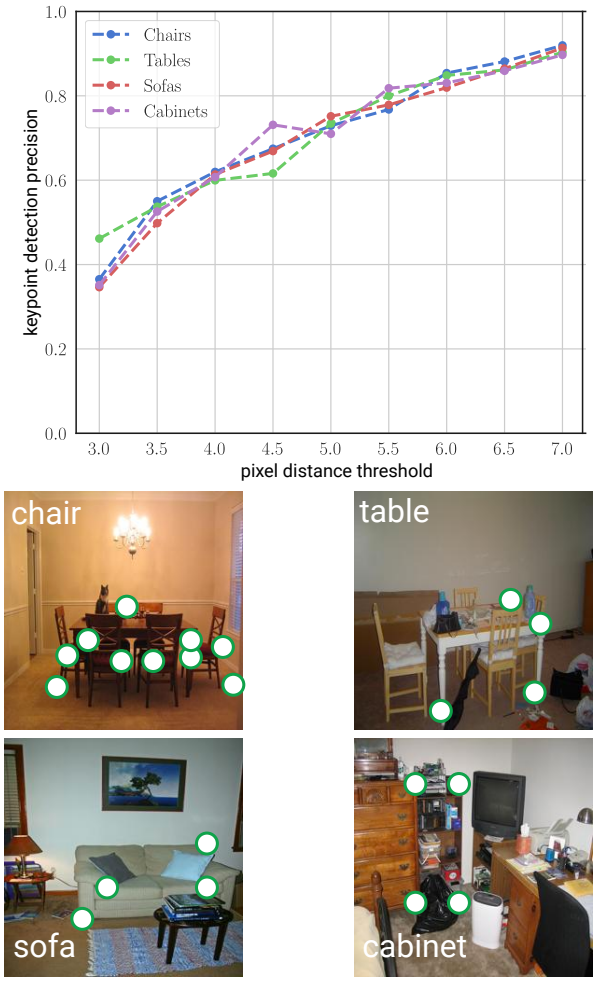


Figure 3. Image-space keypoint detection precision. Images show qualitative image-space keypoint detected for different objects.

annotated chair object against 70+% by our keypoint detection based approach.

Quantitative measures. We use *source* and *target* to denote the two scenes between which a measure is computed. We specifically do not use 'result scene' and 'ground truth scene' as the ground truth acts as a target to compute precision, and acts as source to compute recall.

We denote the objects in the source and target scene as $o_S \in S$, $o_T \in T$, respectively. We use $J_3(o_S, o_T)$ and $J_2(o_S, o_T)$ to represent the Jaccard index or *intersection-over-union* (IoU) of the bounding boxes of o_S and o_T in 3D world space and 2D screen space, respectively. Finally, given an object o_S we define the 'J_i^* correspondence' with T as the object with the MaxIoU with o_S as: $J_i^*(o_S, T) := \arg\max_{o_T \in T} J_i(o_S, o_T)$. Intuitively, this returns, for a given object, the *best matching* object from the other scene in terms of overlap.

(a) IoU3D: This measures average IoU for 3D bounding boxes around objects. Specifically, given a source scene and a target scene, we average MaxIoU across all objects in the source scene (measuring IoU overlap with the corresponding object in the target).

(b) IoU2D: Similar to IoU3D, this measure averages IoU for 2D bounding boxes around projected objects.

(c) Loc: This measures the fraction of correct locations of objects in the source scene with respect to the target. We consider every object in the source scene that has a J_3^* correspondence over a threshold τ_J to have a correct location.

(d) LocAng: Similar to Loc, this measures additionally requires the angle difference to be under a threshold τ_θ.

(e) AngDiff: This measures the average angle difference for the objects that have a correct location.

5.3. Baselines: State-of-the-art Alternatives

We are not aware of prior research focusing on producing scene mockups in the presence of *significant occlusion*. Hence, we created two baselines by combining relevant state-of-the-art methods. We convert the output of each baseline (in both cases 3D pose but 2D image space locations of objects) to our 3D scene mockup format.

(a) SeeingChairs3D. Aubry et al. [4] proposed a method to find chairs by matching so-called 'discriminative visual elements' (DVE) from a set of rendered views of 1000+ chair models with any input image. These DVEs are linear classifiers over HOG features [10] learned from the rendered views in a discriminative fashion. At training time, they are learned at multiple scales while keeping only the most discriminative ones for matching. At test time, a patch-wise matching process finds the best-matching image and rendered patch pairs, and then finds sets of pairs that come from the same rendered view (see [4] for details).

The above method outputs scored image space bounding boxes together with a specific chair model and pose. For our 3D performance measures, however, we need the output in the form of a 3D scene. Hence, we convert each set of bounding box, pose, and chair model to a 3D scene. Using our estimated camera, we optimize the location (in the xz-plane) of the 3D model without changing its pose, such that the 2D bounding box of the projected model matches as closely as possible with the detected bounding box using a least-squares formulation (solved using Ceres [2]).

(b) Im2CAD (FRCNN+3DINN). We combine a convolutional neural network (CNN) trained for image-space object detection and another CNN trained for 3D object interpretation mimicking the Im2Cad system. Specifically, we use FasterRCNN [33] to extract bounding boxes of objects from the input image and then feed these regions of interest to 3D-INN [38], which produces a templated object model consisting of a set of predefined 3D keypoints as well as a pose estimate. Since our set of keypoints is a subset of the keypoints produced by 3D-INN, we use our 3D candidate

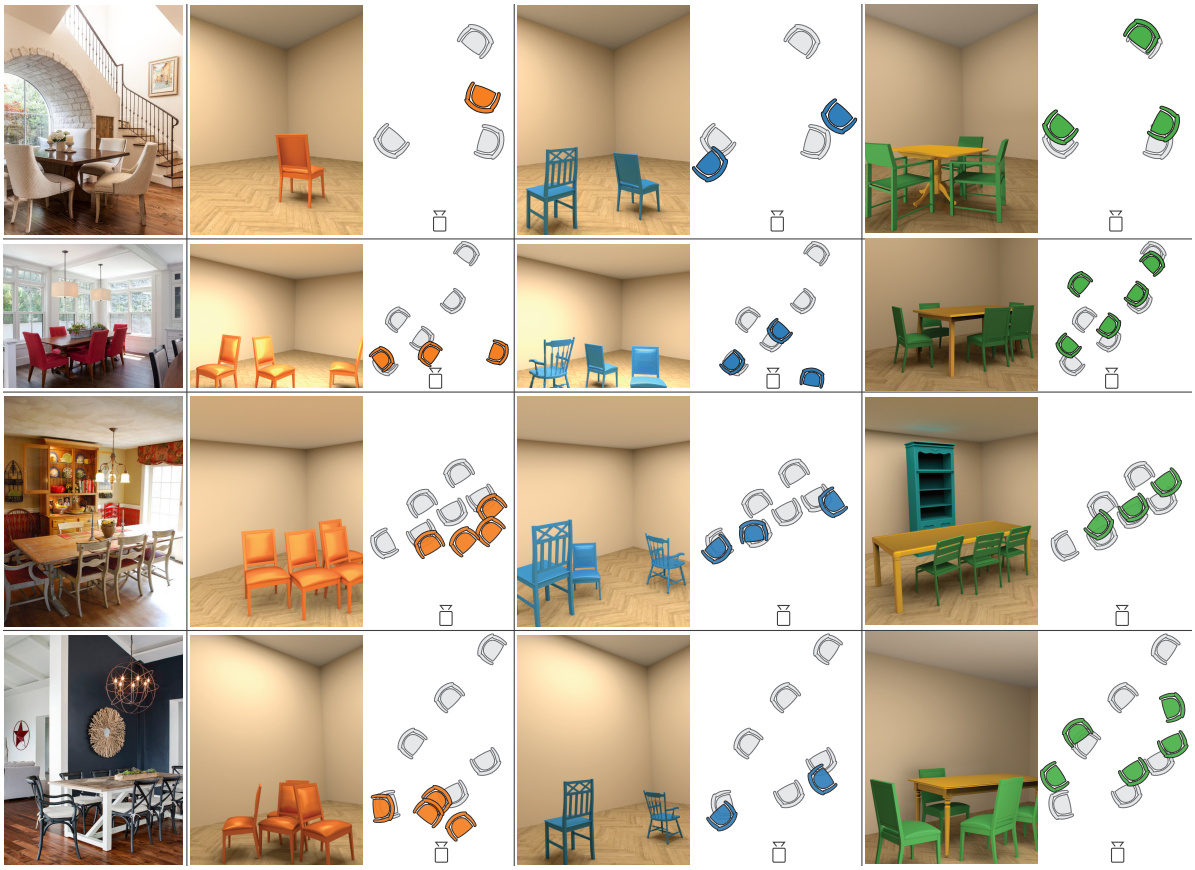


Figure 4. Qualitative comparison of the baseline methods: SEEINGCHAIRS (orange) and IM2CAD (blue) against SEETHROUGH (green). Annotated groundtruth poses (gray) are provided for reference in the top view (only chairs shown in top view to avoid clutter). Note that our approach has higher recall and correctly aligns them compared to the others.

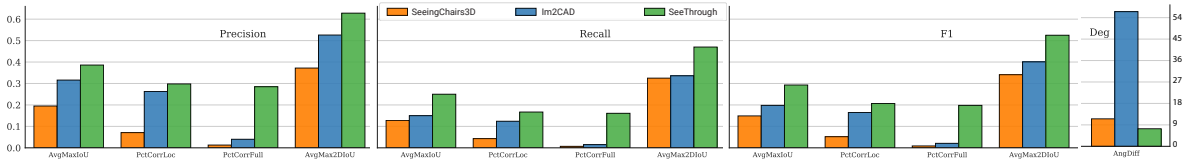


Figure 5. Quantitative performance of SEETHROUGH against the state-of-the-art method-based baseline methods. We outperform the baselines significantly across all the measures. Please refer to supplemental for the tabulated values.

generation part of SEETHROUGH to convert the extracted keypoints to a 3D object for the resultant scene mockup.

5.4. Evaluation and Discussion

We ran SEETHROUGH and the above baseline methods on the full ground truth annotated scene set (Section 5.1). A sampling of results can be seen in Figure 4. (Further visualization for 100 scenes in our groundtruth set are in the supplemental.)

The baseline methods perform well when there is no occlusion in the scene. Specifically, objects that are clearly visible are reconstructed reliably as the direct visual information is sufficient to make an accurate inference about the objects' pose and identity. However, when objects are partly occluded, the methods break down quickly. In contrast, SEETHROUGH, by incorporating co-occurrence object model, is able to recover from these situations.

This difference in performance is also reflected in the quantitative results (see Figure 5). Our method outperforms the baselines on all counts. Additionally, in Figure 6, we show how the LOCANG measure changes under varying thresholds of angle (τ_θ) and IoU (τ_J).

Performance under increasing occlusion. In order to specifically test performance under varying occlusion, we sorted the groundtruth annotated HOUZZ dataset into categories based on the extent of the visible objects. We approx-

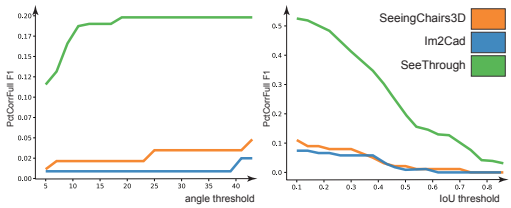


Figure 6. Performance variation according to LOCANG F1 measure for SEETHROUGH and the two baseline methods under varying angle and IoU thresholds. We perform significantly better across both the threshold ranges.

imate visibility as follows: we compute how many objects lie along view rays connecting the estimated camera location with points on a discrete grid on the image plane. We used the objects' bounding boxes for this visibility computation. Higher values denote more occlusion (as there are more objects along the view rays). Figure 1 shows that while all the three methods perform comparably under low occlusion, only SEETHROUGH continues to have a high success rate under medium to heavy occlusion.

Effect of multiple iterations. As an important feature, in Section 5.5, we demonstrate the positive utility of multiple iterations to SEETHROUGH. One of our key observations is that high-confidence objects (e.g., unoccluded objects) are easier to detect, and hence can provide valuable contextual information in reinforcing the weaker signals (e.g., partially occluded objects). This behavior results in higher detection rates using iterations and believed to be also functional in the human perception systems [13, 16].

Utility of synthetic data. We found that training on synthetic datasets [42] for predicting image-space keypoint maps led to unsatisfactory results. For this experiment, we took all renderings from 400K images that contain at least one of the annotated objects and reprojected the keypoint locations from corresponding 3D models into these renders, yielding one image/keypoint map pair as training data per render, resulting in a total of 8000 image/keypoint map pairs. We experimented with different setups: (i) network trained with only synthetic data; (ii) network first trained with synthetic data, and then refined using real data, and (iii) network trained with only real data.

The best performance on the test set resulted from setup #iii, i.e., training with only real data. One likely explanation is that training the network with the synthetic data first steers away the network weights from those that were the result of the ImageNet pretraining, which already encompass a high general understanding of real photographs.

5.5. Ablation Study

We evaluated the importance of the individual steps of SEETHROUGH to the final performance (see Figure 7 and supplemental). Specifically, we ran our pipeline on the full

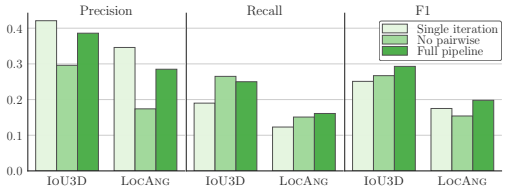


Figure 7. Ablation study evaluating the importance of the different stages of our system.

test set under two weakening conditions: (a) we disable all pairwise costs and run the remaining pipeline based solely on the keypoint location maps; and (b) we disable iterations by running the second and third stage only once, thus removing the possibility of the candidate generation stage benefiting from previously placed objects.

Discussion. Although IOU2D recall increases when disabling scene statistics (option #a), the precision goes down significantly. This is true as the pairwise costs by themselves do not propose new objects – they only make output mockups more precise by pruning objects that do not agree with others. In contrast, using only a single iteration (option #b) increases precision, but recall takes a significant hit. This is not surprising, as in the later iterations the keypoint location maps have decreased influence relative to the pairwise costs. As a result, while objects with weaker keypoint response are more easily found, false positives also become more likely. Overall, the combined IOU2D F1 measure is highest for the full SEETHROUGH as well as the LOCANG F1 measure.

6. Conclusion

We proposed SEETHROUGH, a method for automatically finding partially occluded objects in a photograph of a structured scene. Our key insight is the incorporation of higher level scene statistics that allows more accurate reasoning in scenes containing medium to high levels of occlusion. We demonstrate considerable quantitative and qualitative performance improvements across multiple measures.

Our method suffers from limitations that suggest a number of future research directions. First, we plan to extend the evaluation to more classes of objects beyond those considered. Second, one can explore higher fidelity models to better recover fine scale features in the recovered models. Finally, we would like to explore templates that can express a broader understanding of the multi-object spatial relationships including symmetry and regularity.

Acknowledgement. This work is in part supported by the Microsoft PhD fellowship program, and ERC Starting Grant SmartGeometry (StG-2013-335373). Also, special thanks to Aron Monszpart, James Hennessey, Carlo Innamorati, Paul Guerrero, and other group members for invaluable help at various stages of the project.

References

[1] IBM ILOG CPLEX Optimizer. https://www.gams.com/latest/docs/S_CPLEX.html.

[2] S. Agarwal, K. Mierle, and Others. Ceres solver. http://ceres-solver.org.

[3] B. Andres, T. Beier, and J. Kappes. OpenGM: A C++ library for discrete graphical models. *CoRR*, abs/1206.0111, 2012.

[4] M. Aubry, D. Maturana, A. A. Efros, B. C. Russell, and J. Sivic. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *Proc. IEEE CVPR*, pages 3762–3769, 2014.

[5] A. Bansal, B. Russell, and A. Gupta. Marr revisited: 2D-3D alignment via surface normal prediction. In *Proc. IEEE CVPR*, pages 5965–5974, 2016.

[6] J. Bergstra, D. Yamins, and D. D. Cox. Hyperopt: A python library for optimizing the hyperparameters of machine learning algorithms. In *Proc. Python in Science*, pages 13–20, 2013.

[7] A. X. Chang, T. A. Funkhouser, L. J. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu. Shapenet: An information-rich 3D model repository. *CoRR*, abs/1512.03012, 2015.

[8] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3D geometric phrases. In *Proc. IEEE CVPR*, pages 33–40, 2013.

[9] W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Indoor scene understanding with geometric and semantic contexts. *IJCV*, pages 204–220, 2015.

[10] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. IEEE CVPR*, pages 886–893, 2005.

[11] S. Dasgupta, K. Fang, K. Chen, and S. Savarese. Delay: Robust spatial layout estimation for cluttered indoor scenes. In *Proc. IEEE CVPR*, pages 616–624, 2016.

[12] L. Del Pero, J. Bowdish, D. Fried, B. Kermgard, E. Hartley, and K. Barnard. Bayesian geometric modeling of indoor scenes. In *Proc. IEEE CVPR*, pages 2719–2726, 2012.

[13] J. J. DiCarlo, D. Zoccolan, and N. C. Rust. How does the brain solve visual object recognition? *Neuron*, 73(3):415–434, 2012.

[14] M. Fisher, D. Ritchie, M. Savva, T. Funkhouser, and P. Hanrahan. Example-based synthesis of 3D object arrangements. *Proc. ACM/SIGGRAPH Asia*, pages 135:1–135:11, 2012.

[15] M. Fisher, M. Savva, Y. Li, P. Hanrahan, and M. Nießner. Activity-centric scene synthesis for functional 3D scene modeling. *Proc. ACM/SIGGRAPH*, pages 179:1–179:13, 2015.

[16] A. M. Fyall, Y. El-Shamayleh, H. Choi, E. Shea-Brown, , and A. Pasupathy. Dynamic representation of partially occluded objects in primate prefrontal and visual cortex. *eLife*, page e25784, 2017.

[17] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proc. IEEE CVPR*, pages 770–778, 2016.

[18] V. Hedau, D. Hoiem, and D. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. ICCV*, pages 1849–1856. IEEE, 2009.

[19] V. Hedau, D. Hoiem, and D. A. Forsyth. Recovering the spatial layout of cluttered rooms. In *Proc. ICCV*, pages 1849–1856, 2009.

[20] D. Hoiem, A. Efros, and M. Hebert. Automatic photo pop-up. *ACM SIGGRAPH*, pages 577–584, 2005.

[21] Q. Huang, H. Wang, and V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics (TOG)*, 34(4):87, 2015.

[22] M. Hueting, V. Patraucean, M. Ovsjanikov, and N. J. Mitra. Scene structure inference through scene map estimation. In *VMV*, 2016.

[23] H. Izadinia, Q. Shan, and S. M. Seitz. Im2cad. In *Proc. IEEE CVPR*, pages 2422–2431, 2017.

[24] N. Kholgade, T. Simon, A. Efros, and Y. Sheikh. 3D object manipulation in a single photograph using stock 3D models. *ACM SIGGRAPH*, 33(4):127, 2014.

[25] Y. M. Kim, N. J. Mitra, D.-M. Yan, and L. Guibas. Acquiring 3D indoor environments with variability and repetition. *ACM SIGGRAPH Asia*, pages 138:1–138:11, 2012.

[26] Y. Li, A. Dai, L. J. Guibas, and M. Nießner. Database-assisted object retrieval for real-time 3D reconstruction. *Comput. Graph. Forum*, pages 435–446, 2015.

[27] J. J. Lim, A. Khosla, and A. Torralba. Fpm: Fine pose parts-based model with 3D cad models. In *Proc. ECCV*, pages 478–493. Springer, 2014.

[28] J. J. Lim, H. Pirsiavash, and A. Torralba. Parsing IKEA objects: Fine pose estimation. In *Proc. ICCV*, pages 2992–2999, 2013.

[29] A. Mallya and S. Lazebnik. Learning informative edge maps for indoor scene layout prediction. In *Proc. ICCV*, pages 936–944, 2015.

[30] O. Mattausch, D. Panozzo, C. Mura, O. Sorkine-Hornung, and R. Pajarola. Object detection and classification from large-scale cluttered indoor scans. *Comput. Graph. Forum*, pages 11–21, 2014.

[31] A. Monszpart, N. Mellado, G. J. Brostow, and N. J. Mitra. Rapter: rebuilding man-made scenes with regular arrangements of planes. *ACM SIGGRAPH*, 2015.

[32] E. Olson and P. Agarwal. Inference on networks of mixtures for robust robot mapping. *I. J. Robotics Res.*, pages 826–840, 2013.

[33] S. Ren, K. He, R. B. Girshick, and J. Sun. Faster R-CNN: towards real-time object detection with region proposal networks. In *Proc. NIPS*, pages 91–99, 2015.

[34] A. G. Schwing, S. Fidler, M. Pollefeys, and R. Urtasun. Box in the box: Joint 3D layout and object reasoning from single images. In *Proc. ICCV*, pages 353–360, 2013.

[35] T. Shao, A. Monszpart, Y. Zheng, B. Koo, W. Xu, K. Zhou, and N. J. Mitra. Imagining the unseen: stability-based cuboid arrangements for scene understanding. *ACM SIGGRAPH Asia*, pages 209:1–209:11, 2014.

[36] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proc. IEEE CVPR*, pages 1510–1519, 2015.

[37] S. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh. Convolutional pose machines. In *Proc. IEEE CVPR*, pages 4724–4732, 2016.

[38] J. Wu, T. Xue, J. J. Lim, Y. Tian, J. B. Tenenbaum, A. Torralba, and W. T. Freeman. Single image 3D interpreter network. In *Proc. ECCV*, pages 365–382, 2016.

[39] Y. Xiang, W. Kim, W. Chen, J. Ji, C. Choy, H. Su, R. Mottaghi, L. Guibas, and S. Savarese. Objectnet3d: A large scale database for 3D object recognition. In *Proc. ECCV*, pages 160–176, 2016.

[40] J. Xiao, B. Russell, and A. Torralba. Localizing 3D cuboids in single-view images. In *Proc. NIPS*, pages 746–754, 2012.

[41] Y. Zhang, S. Song, P. Tan, and J. Xiao. Panocontext: A whole-room 3D context model for panoramic scene understanding. In *Proc. ECCV*, pages 668–686, 2014.

[42] Y. Zhang, S. Song, E. Yumer, M. Savva, J. Lee, H. Jin, and T. A. Funkhouser. Physically-based rendering for indoor scene understanding using convolutional neural networks. In *Proc. IEEE CVPR*, pages 5057–5065, 2017.