Model-Driven Indoor Scenes Modeling from a Single Image

Zicheng Liu[1,2] Yan Zhang[1,2*] Wentao Wu[1,2] Kai Liu[1,2] Zhengxing Sun[1,2]

[1]State Key Lab for Novel Software Technology, Nanjing University, China
[2]Department of Computer Science and Technology, Nanjing University, China

ABSTRACT

In this paper, we present a new approach of 3D indoor scenes modeling on single image. With a single input indoor image (including sofa, tea table, etc.), a 3D scene can be reconstructed using existing model library in two stages: image analysis and model retrieval. In the image analysis stage, we obtain the object information from input image using geometric reasoning technology combined with image segmentation method. In the model retrieval stage, line drawings are extracted from 2D objects and 3D models by using different line rendering methods. We exploit various tokens to represent local features and then organize them together as a star-graph to show a global description. Finally, by comparing similarity among the encoded line drawings, models are retrieved from the model library and then the scene is reconstructed. Experimental results show that, driven by the given model library, indoor scenes modeling from a single image could be achieved automatically and efficiently.

Index Terms: I.2.10 [ARTIFICIAL INTELLIGENCE]: Vision and Scene Understanding—Modeling and recovery of physical attributes; I.3.5 [COMPUTER GRAPHICS]: Computational Geometry and Object Modeling—Physically based modeling;

1 INTRODUCTION

With the advent of Digital Photography Age, digital images have become one of the most colorful medias. On the basis of various objects in images, people capture great creative inspiration because, various kinds of corresponding models can be found or created in the real world. Hence, from the image, it's possible to construct a scene with 3D models in a virtual world and many researchers have established geometric modeling based on images. Besides, 3D reconstruction from real world images is always an important direction in the 3D modeling field [13] [5].

Recently, with the rapid development of modeling technology, there has been an explosive growth of 3D models on the Internet. Several model libraries provide rich resources, like 3D Warehouse from Google, 3D Model Search Engine from Princeton Shape Retrieval and Analysis Group, Shape Repository from Aim@shape, etc. Using these existing resources, auto-generating or reconstructing new 3D models has become another interesting research field [3] [36] [1].

As 3D model reconstruction from a single image has always been a confusing problem, it is difficult to directly construct the scene from images. However, many existing libraries contain huge number of useful surface models, from which similar models can be retrieved to reconstruct the virtual scene. From this point of view, we present a new approach for model-driven indoor scene modeling from single image. As we consider that color images contain a great of geometric cues, so in the first image analysis stage, geometric reasoning combined with image segmentation method is applied to obtain main objects information from indoor scene images. The extracted object information can be used in the following retrieval stage. In the model retrieval stage, the idea from sketch-based retrieval [27] [11] is borrowed. In our method, since 2D objects and 3D models can all be represented by line drawings, similar models can be easily got from the model library by comparing their similarity. Specially, in line drawing representation, we propose a novel feature encoding method, which hierarchically combines global and local features together to represent a line drawing. In this way, similar 3D models can be retrieved reasonably.

*Corresponding author: zhangyannju@nju.edu.cn

Contributions In summary, our main contributions include (1) With the support of existing model libraries, an approach of 3D indoor scenes reconstruction from a single image is presented. This method reconstructs objects from the scene automatically and efficiently. (2) We propose a novel feature encoding method for line drawings. In this encoding method, not only local features but also structured global features are represented. It converts the clutter of pixel data into an ordered expression in line with the human visual perception, and also the retrieval precision is improved accordingly.

2 RELATED WORKS

Image-based 3D reconstruction Image-based modeling is always an ideal way for constructing a 3D world in people's mind. Traditional image-based modeling technologies require one to take photos from different viewpoints for one scene and then use technologies such as stereo vision to construct 3D scenes [26]. However, inconvenience of taking several photos for a scene leads to the limitations of above methods. Hence, ideas of single-image-based modeling technology would be more widely accepted, as a single scene image is quite available.

In computer vision field, traditional single-image-based modeling methods construct a 3D scene using cues like lightness, texture, focal length, etc. But these methods always have strict restriction for objects properties, such as shapes and light reflection in a scene. They are just available for some certain scenes. In subsequent studies, some researchers attempt to add interactions to simplify the reconstruction problem [7] [5]. Many of these methods construct the scene through manually setting vanishing points and geometric invariant for an image. But they also have some limitations that only certain geometry and basic plane can be constructed. In our method, based on the existing model library, we could reconstruct a more reasonable scene from a single image.

Model-driven 3D shape modeling In recent years, data-driven modeling based on the model library has become an emerging modeling way, as the number of 3D models on the Internet is rapidly increasing. Many studies have achieved modeling technologies for single object. In these studies, there are three categories according to various input items. In the first category, people transform models guided by images. In Ref. [3] [2] [36], with the help of partitioned models in library, people use operations like parts retrieval and model deformation to reconstruct a new model with the guidance of an image. Yunhai Wang et al. introduce projective analysis for semantic segmentation and labeling of 3D shapes [31]. Another category of methods starts from sketches.

Some researchers have implemented the sketch-based model retrieval technologies [11] [27]. With the sketches as input, they retrieve similar models in the model library. However, this kind of methods pay more attention on the increase of retrieval precision, but not the modeling itself. Moreover, some people use sketches as the guidance to construct 3D model based on the model library. They provide a new direction for single object modeling [12] [34]. In the third category, people use collected point cloud data [29] or estimate image depth [30] to help complete the reconstruction. However, only single object could be reconstructed with all the above methods. Also, apart from sketch-based retrieval technology, other methods have certain limitations on the model library. That is, when objects in different categories have to be modeled, the library has to be built by categories. However, in the scene reconstruction, as there are several objects in one scene, it's difficult to exploit above methods. For the reason that different objects in a scene are not marked, we don't know how to map them to the model categories.

Model-driven 3D scenes modeling With the development of technology, some people proposed scene modeling approaches based on the model library which are classified into two categories according to different input items. One category of methods is for sketches. Based on the existing model library, sketch retrieval technology is combined with sketch modeling technology to construct a scene in some approaches [20] [35]. Cases such as various objects in one scene can be handled. However, there is still certain requirement on the order and category for input sketches. In the other category of methods, with the help of model library, people use the collected point cloud data to construct a scene [28] [25] [4]. They segment the scene image with point cloud data. Then, each object in image can be located and similar object is easily retrieved in the model library. When every object is fixed, the scene reconstruction is completed accordingly. Although these studies have made some achievements, there are special requirements for the input. Besides, this technology is inconvenient for common people, especially non-professional users would find it difficult to draw the scene sketch and unable to collect point cloud data without professional devices. We note that single scene image is quite easily obtained from the real world, so if the scene reconstruction can be completed from a single image with the help of existing model library, the method would be more valuable and widely available for common users. We start from above motivation to carry out research about reconstruction from a single image.

3 Overview

The input of our algorithm is an image I taken from a single viewpoint and model library S containing many furniture models. In our method, we would reconstruct the scene in a two-stage process involving image analysis and model retrieval. Figure 1 shows the overview of our approach.

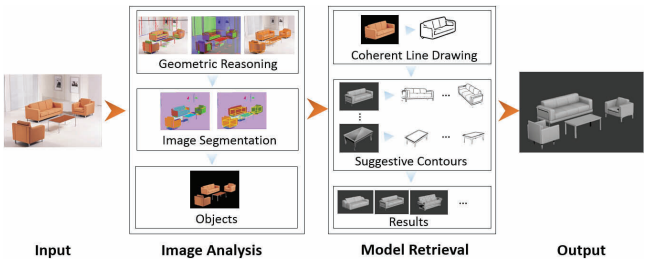

Figure 1: An overview of our approach. With an input indoor scene image, the output 3D scene is reconstructed with a two-stage process: image analysis and model retrieval.

Image analysis In this stage, the main goal is to extract objects from image I for the convenience of following retrieval. The corresponding objects are represented as (O_1,..., O_n). In this paper, geometric reasoning [14] [19] is used combined with image segmentation technology to complete the analysis task. First, by geometric reasoning method, camera parameters as well as cuboids of objects in image can be obtained. Then the scene image is over-segmented using mean-shift image segmentation method [6], in which the over-segmented areas are merged to obtain the target objects from the input image with the guidance of the cuboids we just obtained. Detailed steps are introduced in section 4.

Model retrieval Our main task of this stage is to retrieve similar models based on obtained objects to reconstruct a scene. To handle the work, we first exploit the idea of sketch retrieval [11]. Different line rendering methods are used to extract line drawings for image objects and models in the model library S. Then, to facilitate the model retrieval, line drawings are encoded into a special feature representation. Inspired by the image organization method Patch Net [15], we propose a new feature encoding method. In this paper, we use tokens to represent the local feature in a line drawing and then form the tokens into a star-graph for global feature representation. Next, guided by the viewpoint information from image analysis stage, each object from input image can be retrieved in the model library with graph-matching method. Finally, we place the models into 3D space of the scene to finish the reconstruction. The details is shown in section 5.

4 Image Analysis

A single input image may contain several objects and objects are retrieved one by one in the model retrieval stage. So it requires us to know the information about objects in an image. In this section, we will introduce the details about the image analysis. Limitations are discussed in section 6.2.

4.1 Geometric Reasoning from a Single Image

To obtain the object information in an image, the most direct way is to conduct the image segmentation. However, we find that existing segmentation methods are difficult to handle complex scenes and we want more information about the image such as the viewpoint of the scene, so simple image segmentation method cannot meet our requirement. Taking the above into account, we need more cues for image analysis. We know that there exists information such as geometric relationship between lines, parallax relationship, contours in an image. With above passive cues, we can easily reason the geometric locations of objects in a scene. These studies, geometric reasoning based on the geometrical characteristics in 3D space, have achieved a series of research results [14] [19] [38] [33]. In this way, by geometric reasoning, we could obtain the object information from an image. In our approach, we use these reasoning methods [14] [19] to analyze the image, obtaining the viewpoint of a scene and the cuboids of main objects. Scene viewpoints help select line drawings in the retrieval stage and object cuboids are used to extract objects from image after image segmentation.

For an input image in Figure 2(a), we first extract line segments using Canny edge detector, link edge pixels and fit line segments in Figure 2(b). Then vanishing points could be recovered from these line segments. With line-sweeping algorithm, we divide these line segments into three groups and form one plane from each line segment group which represents an orientation map (as shown in Figure 2(c)) in the scene area. Different map combinations form a number of cuboids which are possible objects in a scene [14] (as shown in Figure 2(d)). Also, Room hypotheses (as shown in Figure 2(e)) are generated from line segments in a way similar to the method described in Lee et al. [19]. Next, we examine exhaustive combinations of one room hypothesis with several cuboids to form

a series of scene configuration hypotheses. But not all scene configurations are reasonable. So we check them to reject invalid configurations by spatial reasoning which define volumetric constraints of spatial exclusion between wall and objects or between objects themselves. By evaluating scene configurations [14], the most reasonable scene configuration hypothesis could be obtained. In our experiment, if automatic analysis result is unreasonable, we interactively locate one satisfied cuboid and then our system iteratively perform spatial reasoning and scene configuration evaluating to locate other cuboids. Finally we get a satisfied scene configuration and main objects cuboids are shown in Figure 2(f).

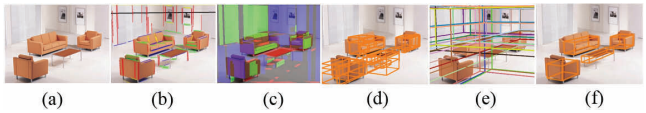

Figure 2: Geometric reasoning process. (a) Input image. (b) Extracted line segments. (c) Three orientation maps in reasoning process. (d) Cube hypotheses. (e) Room hypotheses. (f) Cuboids for main objects in the image.

4.2 Objects Obtaining

We have just obtained objects locations from image in section 4.1. But it is not enough to guide the following retrieval. In the retrieval stage, we need other information such as object contours. We use image segmentation method to over-segment an input image. Although the over-segmented results can hardly show main objects in a scene, with regulation of cuboids in section 4.1, we can obtain each main object information by the method of proportionally combining over-segment areas in a cuboid. Figure 3 shows an overview of process about locating the objects. In our method, we first use mean-shift method [6] to over-segment the image. For the input image in Figure 2(a), we get over-segmented result shown in Figure 3(a). Then these over-segmented areas are combined together by proportionally into the cuboids and the result is shown in Figure 3(b). Through above operations, objects (O_1,..., O_n) are extracted from the image. Figure 3(c) shows the final segmentation results for this input image and four objects are extracted to be candidates for following modeling.

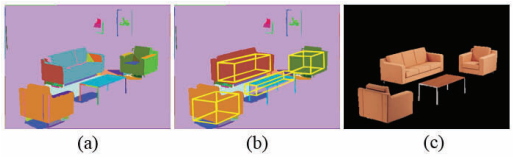

Figure 3: Object obtaining process. (a) Over-segmented result for the input image using mean-shift method. (b) Combination result with the constraint of cuboids. (c) Main objects we extract from the image.

5 Model Retrieval

After obtaining main objects (O_1,..., O_n) from the input image I, we can retrieve similar models in the model library S. For each retrieval operation, one object O_i is handled. In most cases, when people want to retrieve 3D models from 2D objects, they always compare the similarity between them. So in the model retrieval stage, our main problems are: 1) How to represent both the 2D objects and 3D models in a consistent form and conduct feature analysis. 2) What matching mechanism we can use to perform the retrieval efficiently. Our methods for solving above two problems will be introduced in the following.

5.1 Consistent feature representation for 2D objects and 3D models

We use the 2D object information as query to conduct model retrieval. To get a more reasonable result, we need to represent the features of 2D object and 3D model at the same level. Inspired by the sketch retrieval methods [11], we extract line drawings from image objects O_i and render lines for models in model library in order to represent them in the same way of line drawing. In line rendering for objects, we use coherence line drawing [17] (CLD) method for the task. However, since we use different line-rendering methods for 2D objects and 3D models, properties of these two line drawings are apparently different. With this in consideration, we treat the 2D object line drawings by operations of resizing, smoothing and eroding to make them have the same characteristic as 3D model line drawings. Figure 4 shows the main objects obtained from Figure 3, line drawings by coherent line drawing method and final line drawings after our procession. As the results, major line feature information for an object can be shown in the line drawing by this way. Then we use suggestive contours [10] (SC) method to render the models into line drawings. Figure 5 gives the line drawing examples for several models in library. Also, we can see that it represents the model feature details well.

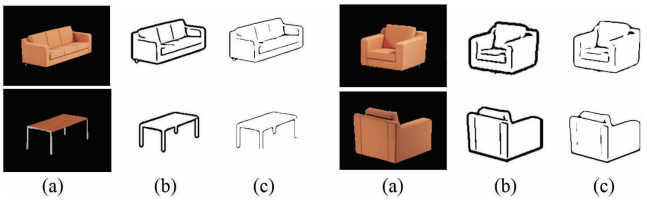

Figure 4: Line drawings extracted from objects in Figure 3. (a) Input objects. (b) Line drawings by CLD method. (c) Line drawings after processing.

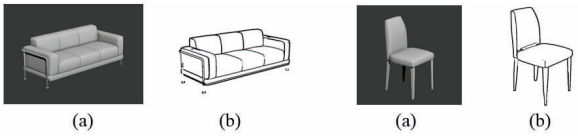

Figure 5: Line drawings of models in library. (a) Input models. (b) Line drawings by SC method.

After the operations above, 2D objects and 3D models are all represented in the same way. Then we extract features from these line drawings. In the process of feature extraction, many kinds of descriptors can be used. Studies on bag-of-word feature [8], SIFT descriptor [24], etc. have made many achievements. However, these descriptors only represent one aspect of the feature, namely the local or global description. The image structures are not considered and hence we may fail to represent relationships of all the features. Recently, a new study named PatchNet [15] is proposed. It represents the image into a graph, with the local and global features linked closely. In our approach, we take inspiration of the PatchNet idea, using various tokens to represent local features in a line drawing, and then organizing the tokens together as a star-graph [22] to show a hierarchical description.

We exploit the idea of sketch tokens [23] to obtain local feature from the line drawings. Our aim is to define a group of tokens to represent different local features in edge structure,including line, T-connection, Y-connection, inflection, parallel, etc. In this paper, model line drawings are rendered under different viewpoints for all 180 models in the model library and these line drawings are taken

as the train data. For an indoor scene, we seldom view the bottom of objects. So we focus on the top views. In our experiment, we take 14 viewpoints in horizontal scale and 6 viewpoints in vertical scale with totally 15120 line drawings. In training process, each line drawing is set in the size of 820*668. According to this size, we sample them by 35*35 patches which have better local feature description from our experiment. The patch sampling details are introduced as follows. First, we collect all black pixels of the line drawing image. Then, for one certain pixel, we delete pixels around it whose Manhattan distance is less than 17. This procedure is performed iteratively until a series of sparse points left. With these sparse points as centers, we sample patches from image and finally obtain many 35*35 size patches. Next, we exploit the Daisy descriptor [32] to calculate each local feature of patches and cluster patches using k-means clustering method. In this paper, we get 150 cluster centers which are known as 150 tokens. Figure 6 shows some tokens of our result.

With these tokens, we can easily represent an input line drawing image by token replacement method by replacing local patches with the most similar tokens. For details, at each time, we calculate Daisy features of all patches and each patch is centered by one black pixel in the line drawing image. Comparing all patches with the 150 tokens, we get the most similar pair and replace the patch with its token pair. Then delete the replaced patch in original line drawing and iterate until almost all patches are replaced. We proceed token replacement for line drawings in Figure 4 and 5, and the result are shown in Figure 7 (Set 1 corresponds to Figure 4(c) and Set 2 corresponds to Figure 5(b)). We can see that the structure of the line drawings do not change too much after the replacement.

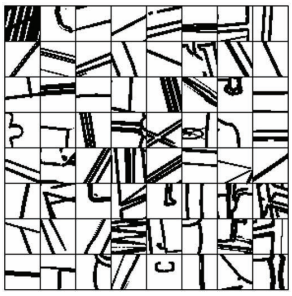

Figure 6: Examples of tokens learned from line drawings. Each token is a clustering center representing local feature.

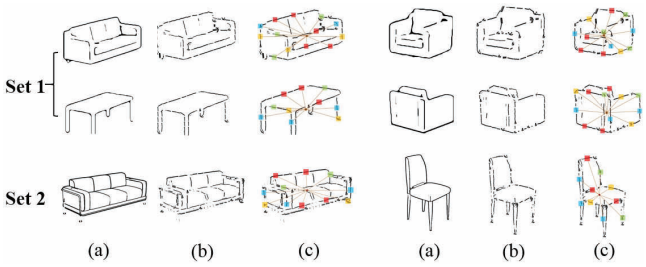

Figure 7: Token replacement and star-graph schematic diagrams for line drawings. (a) Original line drawings of objects (figure 4(c)) and models (figure 5(b)). (b) Token replacement results for figure 7(a). (c) Star-graphs corresponding to each line drawing. In each star-graph, color blocks represent tokens, and lines which link graph center with tokens represent distance vectors. Also, in figure 7, Set 1 is the encoding result for objects, and Set 2 is for models.

However, above steps merely describe the local feature and lack global descriptions. In order to show the line drawings from an integral and structural perspective, we combine the local and global features together to make a representation. The star-graph [22] is used to organize the discrete tokens. With it, the whole line drawing can be represented by a global structure. Details about star-graph are as follows. Firstly, we find the center point of an object, from which we draw a line to each token center. By this means, star-graphs are constructed. The simple schematic diagram is shown in Figure 7(c). We represent each star-graph as G=(V,E), V is the nodes (each token is a node, as color blocks in Figure 7(c)), E is the edges (each edge is a vector from object center to token center, as lines in Figure 7(c)). For visibility of the star-graph, we gives a part of tokens and same token color represent the same token. We can see that the star-graphs have a good structure to organize the local features.

By the above encoding approach, we can prepare all the models off-line in the model library. That is, for each model in the library, we render the model projection under different viewpoints and then encode them with above approach. Although this task takes much time, since it is the off-line work and we don't spend too much time when retrieving. It will improve the efficiency of our retrieval task much.

5.2 Retrieval process

After above feature representation, we begin the retrieval stage. In traditional sketched-based model retrieval or image-based single-object reconstruction [36] [12] [11], there isn't much prior knowledge to use. To get a better retrieval result, researchers always project the model in many different viewpoints by which to compare different projection features. This method complicates retrieval process and also increases calculation. However, in section 4.1, we mentioned that an image includes many cues, based on which we can reason the geometric information of that image to obtain the scene viewpoints. With the known viewpoint, several viewpoints can be selected from the library and calculation can be simplified greatly when comparing the projection features. Besides, for the following retrieval, since we represent the features into star-graphs, retrieval can be processed by graph-matching approach. Next, we introduce the details of these two works.

5.2.1 Viewpoint selection

In 2D image, the camera position relative to a scene has direct impact on the distributions of vanishing points. Reversely, if we know the vanishing points, it's easy to obtain the camera position. According to geometric reasoning result in sections 4.1, we could estimate three vanishing points by extending three line sets in Figure 2(b). 2D cuboids on the image plane can be seen as the perspective projection of cuboids in 3D space. Under these conditions, we could calculate the scene viewpoint with known vanishing points. Details are introduced as follows.

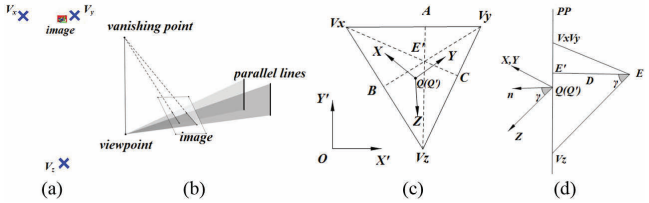

Figure 8: Schematic drawings for how we obtain the scene viewpoint. (a) Three vanishing points of the image. (b) Schematic drawing for parallel lines projection. (c) Front view of the image in three-point perspective. (d) Side view of the image.

We obtained three vanishing points V_x,V_y,V_z from geometric reasoning result, as Figure 8(a) shows. Figure 8(b) is the schematic

drawing for formation of vanishing points. Projection of two parallel lines in 3D space extends with intersection at vanishing point. The line linking the vanishing point and viewpoint is parallel with the original two lines in 3D space. Suppose $QXYZ$ is local object coordinate system and coordinate origin Q has projection point Q' on image projection plane in Figure 8(c). Starting from Q' we draw lines to the three vanishing points V_x,V_y,V_z on image plan, $Q'V_x$,$Q'V_y$,$Q'V_z$ can be seen as projection coordinate axes of $QXYZ$. With the vanishing points V_x,V_y,V_z as vertices, we plot a triangle $\triangle V_xV_yV_z$. For each triangle edge, we construct three hemispheres that intersect at one point, which is known as viewpoint of the scene in 3D space. Figure 8(d) shows the side view of the image plan and viewpoint E has orthographic projection E' on the plan. According to Figure 8(b), we know that the line between viewpoint and one vanishing point is parallel with the corresponding coordinate axis, i.e. in Figure 8(d), $EV_z \| QZ$ (QZ is the z axis of local object coordinate system). Viewpoint E is at the normal of projection plane through the projection point E'. Distance D from viewpoint E to projection plane could be calculated by formula (1).

$$D^2 = |AE'| \times |E'V_z| = |BE'| \times |E'V_y| = |CE'| \times |E'V_x| \tag{1}$$

Next, we move the local object coordinate system $QXYZ$ towards projection plane along the projection direction so that Q and Q' are at the same position. Assuming that Q and Q' are the same point which makes no difference on view angles to the scene, for the reason that changing distance between a scene and projection plane merely changes the size of projection on the plane. In Figure 8(d), $\boldsymbol{n}$ is normal vector of the projection plane and α, β, γ are angles between $\boldsymbol{n}$ and three coordinate axes x,y,z of $QXYZ$ (We only shows γ angle in the figure). Then we could obtain that,

$$\cos\gamma = \frac{D}{|EV_z|} = D/sqrt\left(D^2 + |E'V_z|^2\right) \tag{2}$$

The same procedure may be adapted to obtain the following.

$$\begin{aligned} \cos\alpha &= \tfrac{D}{|EV_x|} = D/sqrt\left(D^2 + |E'V_x|^2\right) \\ \cos\beta &= \tfrac{D}{|EV_y|} = D/sqrt\left(D^2 + |E'V_y|^2\right) \end{aligned} \tag{3}$$

Suppose point E' has the position of $E'(X_{E'},Y_{E'},Z_{E'})$ relative to the coordinate system $QXYZ$ and i_z is the unit vector on $E'V_z$. Then we can obtain from Figure 8(d) that $Z_{E'} = |Q'E' \cdot i_z| \cdot \sin\gamma$, where $Q'E' \cdot i_z$ is the projection vector of $Q'E'$ in i_z direction. Then,

$$Z_{E'} = \left|\mathrm{Q'E'} \cdot i_z\right| \cdot \sin\gamma = \left|\mathrm{Q'E'} \cdot i_z\right| \times \sqrt{1 - cos^2\gamma} \tag{4}$$

The same procedure may be adapted to obtain the following,

$$\begin{aligned} X_{E'} &= |\mathrm{Q'E'} \cdot i_x| \cdot \sin\alpha = |\mathrm{Q'E'} \cdot i_x| \times \sqrt{1 - cos^2\alpha} \\ Y_{E'} &= \left|\mathrm{Q'E'} \cdot i_y\right| \cdot \sin\beta = \left|\mathrm{Q'E'} \cdot i_y\right| \times \sqrt{1 - cos^2\beta} \end{aligned} \tag{5}$$

Hence position of the viewpoint E relative to $QXYZ$ is as follows.

$$E\left(X_E, Y_E, Z_E\right) = E\left(X_{E'} - D \times cos\alpha, Y_{E'} - D \times cos\beta, Z_{E'} - D \times cos\gamma\right) \tag{6}$$

Next, we transform the position into a tuple $\langle\varphi,\theta\rangle$ under the corresponding sphere coordinate system representing deflection angle of the viewpoint to scene models, φ is the horizontal deflection angle and θ is vertical deflection angle.

$$\begin{aligned} \varphi &= \arctan\frac{\sqrt{{X_E}^2 + {Y_E}^2}}{{Z_E}^2} \\ \theta &= \frac{\pi}{2} - \arctan\frac{Y_E}{X_E} \end{aligned} \tag{7}$$

After obtaining the scene viewpoint, all the 3D models should be under similar viewpoint so that they are consistent with the query object. In the following retrieval process, for the permission of error range, we choose projection line drawings within the range of $\langle\varphi+\varepsilon,\theta+\sigma\rangle$, ε, σ is a certain amount of error. That is, in the model library that have been pre-processed, we simply choose projections under feasible viewpoints. We randomly select several models in library and Figure 9 gives model projections under similar viewpoints.

Figure 9: Model projections under similar viewpoints. (a) Object from image. (b) Model projections under similar viewpoints with the object.

5.2.2 graph-matching-based model retrieval

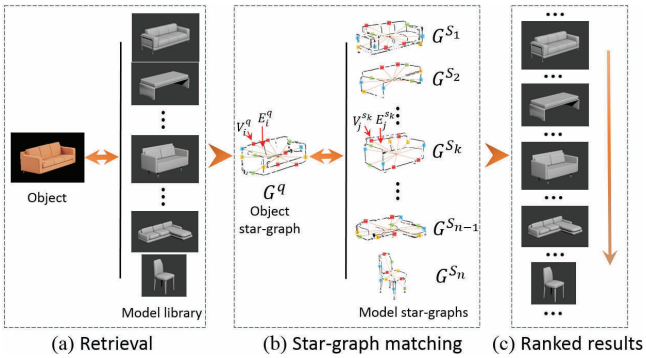

Figure 10: Star-graph-matching based model retrieval. (a) Retrieval. (b) Star-graph matching. (c) Retrieval result ranked by similarity from top to bottom.

After calculating the viewpoint of a scene, we choose model star-graphs from library in a permissible error range. Based on graph matching, model retrieval is conducted shown in Figure 10(a). Firstly, we define the involved variables. Suppose q as object line drawing and S_k as one of n selected line drawings in library. Then in Figure 10(b), the corresponding star-graph is $G^q = (V^q,E^q)$ and $G^{S_k} = (V^{S_k},E^{S_k})$. The similarity between the object q and projection S_k can be calculated with formula (8).

$$P\left(G^q, G^{S_k}\right) = \sum_i \max_j P\left(V_j^{s_k} | V_i^q\right) P\left(E_j^{s_k} | E_i^q\right) \tag{8}$$

Where $P(\cdot,\cdot)$ is a normalized distance value that measuring the similarity between two star-graphs. We can view it as a probability of similarity. ${V_i}^q$ and $V_j^{s_k}$ are nodes in graphs, known as tokens. ${E_i}^q$ and ${E_j}^{s_k}$ are vectors from graph center to token positions. Using them, we calculate the token feature similarity term using the following formula.

$$P\left(V_j^{s_k} | V_i^q\right) = \frac{1}{1 + \exp\left(-\left\|V_i^q - V_j^{s_k}\right\|\right)} \tag{9}$$

Then, the token location similarity term can be obtained by the following formula.

$$P\left(E_j^{s_k} | E_i^q\right) = exp(-\left(E_j^{s_k} - E_i^q\right)^T S_L^{-1}\left(\left(E_j^{s_k} - E_i^q\right)\right) \tag{10}$$

Here, S_L is a constant covariance matric to allow for some deviations in patch locations.

We introduce the details of the star-graph matching in the following. First, for each query token in G^q, we seek for D tokens at the approximate location in G^{S_k}. Here D is much smaller than the number of total tokens. Then, in the D tokens, we select the most similar token with the query token in G^q, using both feature and location similarity term to measure the token similarity probability. Finally, we accumulate similarity probabilities for all query tokens in G^q. The value represents the graph similarity between G^q and G^{S_k}.

For higher retrieval precision, we then select the top M similar graphs from above results to calculate the similarity conversely. That means, for each graph G^{S_k} in the top M, above method is used again to get the $P(G^{S_k},G^q)$ between all M graphs G^{S_k} and the query graph G^q. Then we obtain the final similarity probability P for that query graph in formula (11). The highest score responds to the most similar model in the model library.

$$P = \omega_1 P\left(G^q, G^{S_k}\right) + \omega_2 P\left(G^{S_k}, G^q\right) \tag{11}$$

Where G^q is the query graph and G^{S_k} is a graph of top M in first retrieval. ω_1, ω_2 are the coefficients. We conduct an experiment and assign 0.5 to both coefficients for higher retrieval precision.

After above calculation, we can obtain the ranked similarities between image objects and all models in library shown in Figure 10(c). In this paper, we output the top 5 retrieval results for users to select, as is shown in Figure 11 that yellow frames indicate user's selection. We can see that our result has great similarity with input object. Finally, according to the relative locations of each object in the image, we place user-selected models into the 3D scene. In this way, the scene reconstruction is completed. Figure 12 shows our final 3D indoor scene.

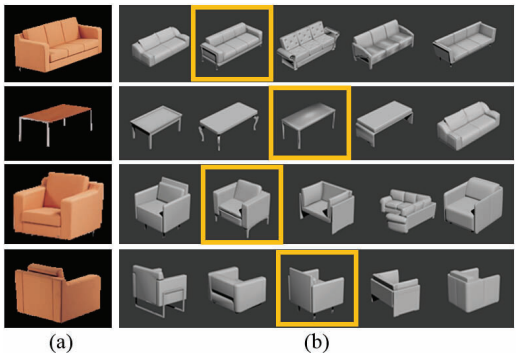

Figure 11: Model retrieval result for the scene objects. (Models in yellow frames indicates user's selection) (a) Input objects. (b) Top 5 retrieval results in our model library.

Figure 12: Scene reconstruction result. (a) Input indoor scene image. (b) Final scene reconstruction result.

6 Experiment and Analysis

In this section, we show our experimental results and analyze them; limitations of our method are discussed as well.

6.1 Result analysis

We implement the algorithm under Windows 8, AMD A8-5500 3.20GHz CPU, 8GB Memory. To verify the effectiveness, we tested with some other indoor scene images. The result is shown in Figure 13. In image analysis stage, geometric reasoning and object extraction consumes about 20s individually. In model retrieval stage, we established the model library with 180 models, among which are 80 sofas, 45 chairs, 20 tables and 35 tea table models (the details are in supplementary material). All the models come from 3D Warehouse. Based on the model library, k-means clustering for 150 tokens takes 10h, and encoding all the line drawings takes about 20h. After feature encoding, for an input scene image, it takes about 50s in reconstruction step(time determined by the image size). Preprocessing is time-consuming however, this work could be done in advance. Thus in model retrieval stage, our method constructs the scene much more quickly.

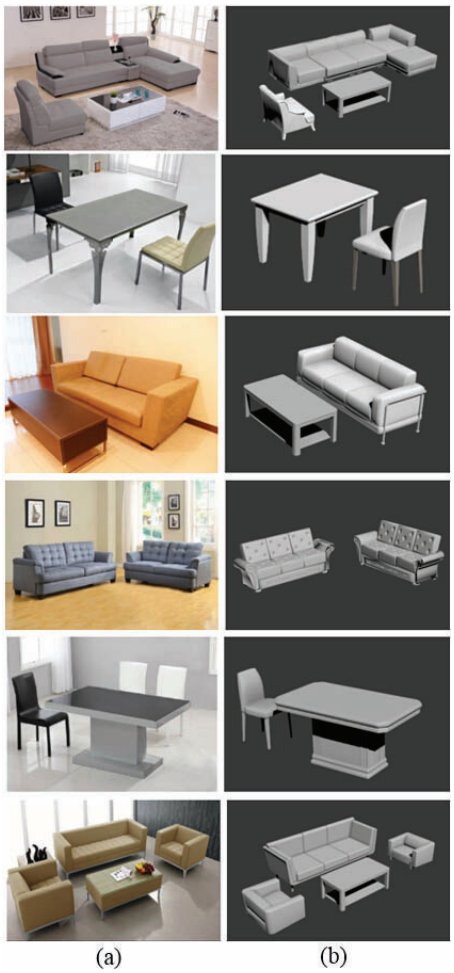

Figure 13: Our reconstruction result. (a) Input indoor scene image. (b) reconstructed 3D scene.

To verify our feature encoding approach, we also conducted common 2D descriptors for comparison. Retrieval results for the extracted objects in Figure 3(c), are provided in Figure 14, using different methods of GALIF [11], Fourier [37], Zernike moments [18] and HOG [9] feature descriptors. Among them, GALIF

is one of the best sketch-based retrieval method in recent years and Fourier, Zernike moments and HOG are common 2D shape feature descriptors that Li et al. [21] mentioned. All retrievals are conducted under the viewpoint obtained in our paper.

It can be seen that our retrieval results have more similarity with the input query object. Besides, for retrieval in mixed model library without category labels, more models from our result are semantically in the same category with the query object. This is mainly because traditional feature encoding methods only start from local or global features. Even though there exist combinatorial optimization of features, it leads to more uncontrollable results and more dissimilarity for the reason that retrieval is not performed on a unified feature. In this paper, we consider the local and global feature in a unified feature system. Models from the same category are more consistent in similarity. Thus the retrieval results are more likely to be in the same category. The feature encoding method we proposed is quite available in mixed model library.

We do not have available Ground Truth for comparison experiment, because we built the library according to our own need. Considering that, we took the idea of Huang et al. [16] who construct Ground Truth by themselves for better comparison. We completed our Ground Truth by an iterative way and details are as follows. In the process, we randomly chose 20 users and asked them to construct their own Ground Truth iteratively one by one. Specially, for each object, the first user select from the model library the top 5 similar models, which are modified by the next user until all users agree with the 5 models. We took the 5 models as Ground Truth for the object. Figure 15 gives Ground Truth for objects in Figure 14. We display the models that belong to the Ground Truth, marking them in color frames. Red frame indicates the model is in the Ground Truth while green frame indicates it beyond the Ground Truth. Our result has more red frames, which means that we have more retrieved models belonging to the Ground Truth. It proves our results are more similar to actual object and better corresponded to characteristics of human vision. Therefore, we get to a conclusion that our approach is more precise and relatively more effective.

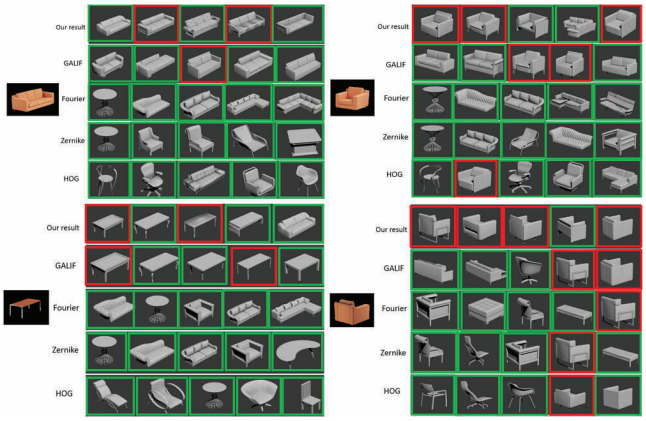

Figure 14: Retrieval results for one scene using different feature descriptors. Red frame indicates the model belongs to the Ground Truth while green frame indicates it beyond the Ground Truth. (Please refer to the supplementary material for clear image)

6.2 Limitations

Firstly, let us discuss about input images. Normally, for an input image, geometric frame structures are needed in image analysis stage. We use edge lines of structures to facilitate calculating the scene vanishing point. Also, image analyzing experiment presents better results for regular geometries than irregular geometries. In object extracting aspect, if there is sharp color contrast between objects

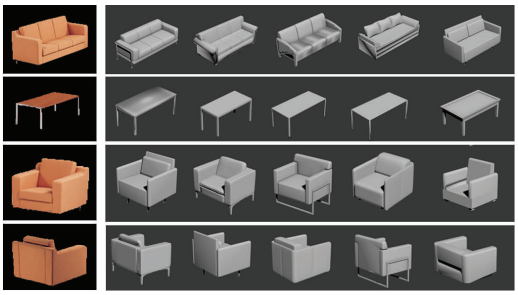

Figure 15: Ground Truth from iterative method.

and background, little interactions are needed in object extracting, as shown in figure 2. For some images however, colors in adjacent areas are quiet similar, so we use obtained cuboids to help segmenting interactions. These interactions towards pre-segmented areas concentrate on borders of cuboids. We simply assign these border areas as part of one object or not in our experiment. In the case of covering, our method tolerates small amount of covering, such as cups on tea table or pillows on sofa. For large covering which change object contours much(e.g. chairs in second and fifth row of figure 13), this method provides a poor result. Generally speaking, our image analysis method provide a reliable performance for most indoor scene images.

Secondly, we describe the line drawings with a hierarchically encoding approach, which is more suitable for retrieval in mixed model library without category labels. It highly improves the retrieval precision. But we just consider the shape and structure of an object in retrieval, not the semantic information of the scene. Hence we may also obtain a similar shape result which is inconsistent with the scene. In the future we would like to learn from the idea [35], conducting scene semantic analysis and doing the joint semantic retrieval.

Thirdly, we do not deform retrieved models. If all the models in the database vary greatly with a query object, it would be difficult to retrieve out a satisfying 3D model. In the future we hope to make the study of deformed method to solve the above problems.

7 Conclusion

In this paper, we present a novel model-driven indoor scenes modeling method based on a single image. With the help of rich model resources on the Internet, we are able to conduct 3D modeling from a single indoor scene image efficiently. The result shows that the method proposed in this paper is simple and efficient. Our method provides a new way for surface modeling technique from a single image.

As for the future work, first of all, we wish to use more reasonable geometric reasoning methods to analysis images better. Besides, we want to use semantic analysis of the scene to achieve joint semantic retrieval. We also consider adding contour deformation in the retrieval process to get more reasonable 3D models.

Acknowledgements

We would like to thank all anonymous reviewers for their constructive comments. This research has been supported by the National Science Foundation of China (61321491, 61100110, 61272219) and the Science and Technology Program of Jiangsu Province (BY2012190, BY2013072-04).

References

[1] 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 *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3762–3769. IEEE, 2014.

[2] S. Chaudhuri, E. Kalogerakis, L. Guibas, and V. Koltun. Probabilistic reasoning for assembly-based 3d modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, page 35. ACM, 2011.

[3] S. Chaudhuri and V. Koltun. Data-driven suggestions for creativity support in 3d modeling. In *ACM Transactions on Graphics (TOG)*, volume 29, page 183. ACM, 2010.

[4] K. Chen, Y.-K. Lai, Y.-X. Wu, R. Martin, and S.-M. Hu. Automatic semantic modeling of indoor scenes from low-quality rgb-d data using contextual information. *ACM Transactions on Graphics (TOG)*, 33(6):208, 2014.

[5] T. Chen, Z. Zhu, A. Shamir, S.-M. Hu, and D. Cohen-Or. 3-sweep: extracting editable objects from a single photo. *ACM Transactions on Graphics (TOG)*, 32(6):195, 2013.

[6] D. Comaniciu and P. Meer. Mean shift: A robust approach toward feature space analysis. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 24(5):603–619, 2002.

[7] A. Criminisi, I. Reid, and A. Zisserman. Single view metrology. *International Journal of Computer Vision*, 40(2):123–148, 2000.

[8] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *Workshop on statistical learning in computer vision, ECCV*, volume 1, pages 1–2, 2004.

[9] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, pages 886–893. IEEE, 2005.

[10] D. DeCarlo, A. Finkelstein, S. Rusinkiewicz, and A. Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (TOG)*, 22(3):848–855, 2003.

[11] M. Eitz, R. Richter, T. Boubekeur, K. Hildebrand, and M. Alexa. Sketch-based shape retrieval. *ACM Trans. Graph.*, 31(4):31, 2012.

[12] L. Fan, R. Wang, L. Xu, J. Deng, and L. Liu. Modeling by drawing with shadow guidance. In *Computer Graphics Forum*, volume 32, pages 157–166. Wiley Online Library, 2013.

[13] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin. Modeling by example. In *ACM Transactions on Graphics (TOG)*, volume 23, pages 652–663. ACM, 2004.

[14] A. Gupta, M. Hebert, T. Kanade, and D. M. Blei. Estimating spatial layout of rooms using volumetric reasoning about objects and surfaces. In *Advances in Neural Information Processing Systems*, pages 1288–1296, 2010.

[15] S.-M. Hu, F.-L. Zhang, M. Wang, R. R. Martin, and J. Wang. Patchnet: a patch-based image representation for interactive library-driven image editing. *ACM Transactions on Graphics (TOG)*, 32(6):196, 2013.

[16] S.-S. Huang, A. Shamir, C.-H. Shen, H. Zhang, A. Sheffer, S.-M. Hu, and D. Cohen-Or. Qualitative organization of collections of shapes via quartet analysis. *ACM Transactions on Graphics (TOG)*, 32(4):71, 2013.

[17] H. Kang, S. Lee, and C. K. Chui. Coherent line drawing. In *Proceedings of the 5th international symposium on Non-photorealistic animation and rendering*, pages 43–50. ACM, 2007.

[18] A. Khotanzad and Y. H. Hong. Invariant image recognition by zernike moments. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(5):489–497, 1990.

[19] D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2136–2143. IEEE, 2009.

[20] J. Lee and T. Funkhouser. Sketch-based search and composition of 3d models. In *Proceedings of the fifth eurographics conference on sketch-based interfaces and modeling*, pages 97–104. Eurographics Association, 2008.

[21] B. Li, Y. Lu, A. Godil, T. Schreck, B. Bustos, A. Ferreira, T. Furuya, M. J. Fonseca, H. Johan, T. Matsuda, et al. A comparison of methods for sketch-based 3d shape retrieval. *Computer Vision and Image Understanding*, 119:57–80, 2014.

[22] Y. Li, Y.-Z. Song, and S. Gong. Sketch recognition by ensemble matching of structured features. 2013.

[23] J. J. Lim, C. L. Zitnick, and P. Dollár. Sketch tokens: A learned mid-level representation for contour and object detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3158–3165. IEEE, 2013.

[24] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[25] L. Nan, K. Xie, and A. Sharf. A search-classify approach for cluttered indoor scene understanding. *ACM Transactions on Graphics (TOG)*, 31(6):137, 2012.

[26] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International journal of computer vision*, 47(1-3):7–42, 2002.

[27] T. Shao, W. Xu, K. Yin, J. Wang, K. Zhou, and B. Guo. Discriminative sketch-based 3d model retrieval via robust shape matching. In *Computer Graphics Forum*, volume 30. Wiley Online Library, 2011.

[28] T. Shao, W. Xu, K. Zhou, J. Wang, D. Li, and B. Guo. An interactive approach to semantic modeling of indoor scenes with an rgbd camera. *ACM Transactions on Graphics (TOG)*, 31(6):136, 2012.

[29] C.-H. Shen, H. Fu, K. Chen, and S.-M. Hu. Structure recovery by part assembly. *ACM Transactions on Graphics (TOG)*, 31(6):180, 2012.

[30] H. Su, Q. Huang, N. J. Mitra, Y. Li, and L. Guibas. Estimating image depth using shape collections. *ACM Transactions on Graphics (TOG)*, 33(4):37, 2014.

[31] Y. Wang, M. Gong, T. Wang, D. Cohen-Or, H. Zhang, and B. Chen. Projective analysis for 3d shape segmentation. *ACM Transactions on Graphics (TOG)*, 32(6):192, 2013.

[32] S. Winder, G. Hua, and M. Brown. Picking the best daisy. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 178–185. IEEE, 2009.

[33] J. Xiao, B. Russell, and A. Torralba. Localizing 3d cuboids in single-view images. In *Advances in Neural Information Processing Systems*, pages 746–754, 2012.

[34] X. Xie, K. Xu, N. J. Mitra, D. Cohen-Or, W. Gong, Q. Su, and B. Chen. Sketch-to-design: Context-based part assembly. In *Computer Graphics Forum*, volume 32, pages 233–245. Wiley Online Library, 2013.

[35] K. Xu, K. Chen, H. Fu, W.-L. Sun, and S.-M. Hu. Sketch2scene: Sketch-based co-retrieval and co-placement of 3d models. *ACM Transactions on Graphics (TOG)*, 32(4):123, 2013.

[36] K. Xu, H. Zheng, H. Zhang, D. Cohen-Or, L. Liu, and Y. Xiong. Photo-inspired model-driven 3d object modeling. In *ACM Transactions on Graphics (TOG)*, volume 30, page 80. ACM, 2011.

[37] D. Zhang, G. Lu, et al. A comparative study on shape retrieval using fourier descriptors with different shape signatures. In *Proc. of international conference on intelligent multimedia and distance education (ICIMADE01)*, pages 1–9, 2001.

[38] Y. Zheng, X. Chen, M.-M. Cheng, K. Zhou, S.-M. Hu, and N. J. Mitra. Interactive images: cuboid proxies for smart image manipulation. *ACM Trans. Graph.*, 31(4):99, 2012.