Structure Recovery by Part Assembly

Chao-Hui Shen[1] Hongbo Fu[2] Kang Chen[1] Shi-Min Hu[1]
[1]TNList, Tsinghua University, Beijing [2]City University of Hong Kong

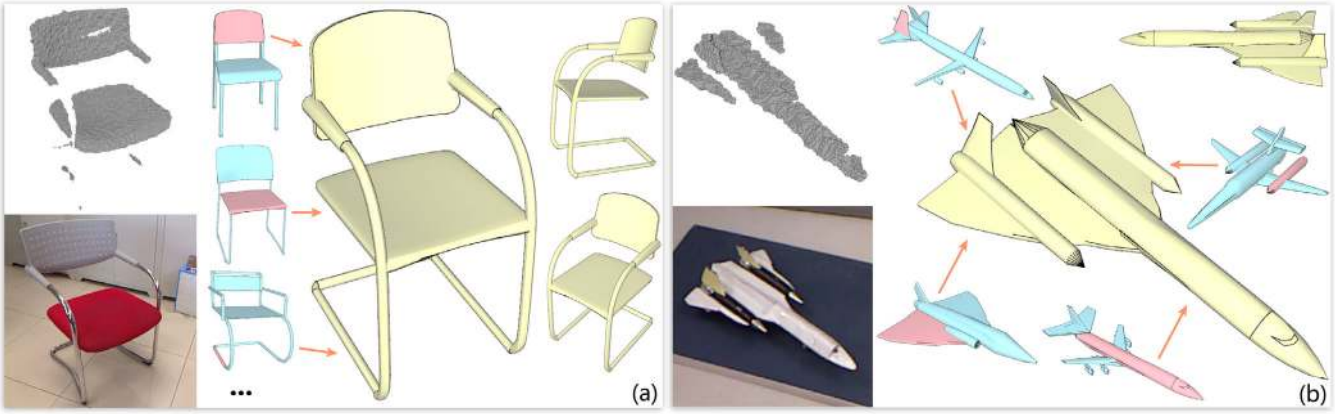

Figure 1: *Given single-view scans by the Kinect system, containing highly noisy and incomplete 3D scans (upper left) and corresponding RGB images (lower left), our approach is able to faithfully recover their underlying structures (yellow) by assembling suitable parts (red) in the repository models (blue).*

Abstract

This paper presents a technique that allows quick conversion of acquired low-quality data from consumer-level scanning devices to high-quality 3D models with labeled semantic parts and meanwhile their assembly reasonably close to the underlying geometry. This is achieved by a novel structure recovery approach that is essentially local to global and bottom up, enabling the creation of new structures by assembling existing labeled parts with respect to the acquired data. We demonstrate that using only a small-scale shape repository, our part assembly approach is able to faithfully recover a variety of high-level structures from only a single-view scan of man-made objects acquired by the Kinect system, containing a highly noisy, incomplete 3D point cloud and a corresponding RGB image.

Links: DL PDF WEB

1 Introduction

3D scanning devices provide a quick way to acquire 3D models of real-world objects or environment, which benefit a variety of applications. However, the acquired models, typically represented as unorganized point clouds, are often corrupted with noise and outlier. Worse, large regions or even entire parts might remain missing (see an example in Figure 1), possibly due to occlusions, grazing angle views, or scanner-unfriendly lighting/materials (e.g., highly reflective materials). These problems further deteriorate for consumer-level scanning devices like the Kinect system of Microsoft, which provide an economical solution to 3D capturing but at the cost of low-quality acquisition of geometry and appearance.

It is challenging to faithfully recover the underlying geometry or structure from such highly incomplete and noisy scan data. Most of the existing works (e.g., [Sharf et al. 2004; Shalom et al. 2010]) focus on geometry completion or reconstruction, and tackle inputs with small deficiencies or simple missing geometry only. Still, it is unclear how to effectively recover the underlying structure even if the geometry gets completed. The template-based approaches [Pauly et al. 2005; Kraevoy and Sheffer 2005] have great potential in completing larger, more complex holes. It is possible to transfer the structural information from the templates to the scan data. However, the existing approaches largely operate in a global-to-local manner, and thus heavily rely on the availability of template models that are globally similar to the underlying object. Although there exist a few online shape repositories like Google 3D Warehouse, the available models are still far from capturing real-world objects exhibiting complex structures, causing the main bottleneck for the existing template-based approaches.

The recent advance in mesh segmentation greatly simplifies the segmentation and labeling of parts in a set of 3D models [Kalogerakis et al. 2010; Huang et al. 2011; Sidi et al. 2011]. The recent works demonstrate how to significantly enlarge the existing database of 3D models via shape synthesis by part composition [Kalogerakis et al. 2012; Jain et al. 2012; Xu et al. 2012]. However, in practice this would result in a 3D model database that grows exponentially, making both the storage and the retrieval challenging to manage. We show that it is unnecessary to explicitly prepare such larger database by part composition and it is possible to retrieve and assemble suitable parts on the fly for structure recovery.

We propose a part assembly approach for structure recovery from a highly incomplete, noisy 3D scan of a man-made object together with the corresponding RGB image acquired by the Kinect system (Figure 1). Our approach is based on the key fact that many classes of man-made objects (e.g., chairs, bicycles etc.) lie in a low-dimensional shape space defined with respect to the relative sizes and positions of shape parts [Ovsjanikov et al. 2011]. This allows us

to quickly filter out most of parts in the database that are irrelevant to the underlying structure, resulting in a small set of top ranked candidates for each part category (Section 4.1). We then compose the structure using a subset of candidate parts for each category, considering both partial matching with respect to the acquired data and the interaction between parts (Section 4.2). Finally, a novel part conjoining approach is proposed to bring the selected parts into a whole (Section 4.3).

We demonstrate that with little user interaction and with a small-scale shape repository, our technique is able to robustly recover the underlying 3D structure, i.e., semantic parts with labels, from a single-view scan of many man-made objects with complex geometry and topology. Although we focus on structure recovery, the geometry of the assembled models is already reasonably close to the underlying geometry. Finally, our bottom-up framework is inherently applicable to multi-view scans, which will be our future work towards robust indoor scene reconstruction.

2 Related Work

There exist many surface completion and reconstruction methods. A full review of them is beyond the scope of our paper. Most of such methods rely on shape continuity inferred from input scans only, and reconstruct or complete the underlying surface largely based on smooth interpolation or extrapolation (see [Shalom et al. 2010; Attene 2010; Lin et al. 2010; Shen et al. 2010] and references therein). Such methods work well for inputs with small deficiencies but have difficulties with large holes where complex geometry is missing. Several approaches [Sharf et al. 2004; Zheng et al. 2010] employ self-similarity priors to tackle input data with rich texture or repetitive elements but still assume that all the necessary content to fill in missing regions could be located somewhere else in the input data. Since the existing methods mostly focus on geometry completion and reconstruction, how to recover underlying high-level structures is still unclear.

Our work is closely related to the existing approaches for example-based completion of arbitrary shapes, whose performance, however, depends on the availability of a single tailor-made template model [Kraevoy and Sheffer 2005] or multiple context models retrieved from a database of 3D models that are *globally* similar to the input model [Pauly et al. 2005]. By assuming shape continuity across the boundaries of missing regions, these approaches first match one or more template models with known regions of the input data and then use the unmatched patches of the template models to fill in missing regions i.e., by performing *shape extrapolation via the template models*. Unlike these approaches, which operate in a global-to-local manner, our approach is essentially local to global and bottom up, thus enabling part assembly on the fly.

Recently Xu et al. [2011] present a photo-inspired 3D modeling approach that deforms individual 3D candidate models to a target image object in a structure preserving manner. Our work takes the same type of models as input, i.e., a database of 3D man-made models that belong to the same class as the target object. However, our approach is largely orthogonal to theirs: their deformation-based approach creates new geometric variations but is intrinsically limited by the available structures in the candidate set, while our part assembly approach automatically produces new structures with respect to the underlying object. Besides, our approach has an extra 3D scan as input, though its quality is rather low.

Our work got inspirations from assembly-based 3D modeling, pioneered by Funkhouser et al. [2004]. For this task, a variety of user interfaces have been proposed for interactive part retrieval, including shape-based search [Funkhouser et al. 2004; Chaudhuri and Koltun 2010] and sketch-based retrieval [Shin and Igarashi 2007; Lee and Funkhouser 2008]. Recent research focuses on data-driven suggestions and aims to support open-ended 3D modeling [Chaudhuri and Koltun 2010; Chaudhuri et al. 2011]. In particular, Chaudhuri et al. use a repository of segmented and labeled shapes, the same as ours. Contemporaneous with our work, several techniques have been proposed to synthesize new shapes by part composition [Kalogerakis et al. 2012; Jain et al. 2012; Xu et al. 2012]. However, all these assembly-based modeling techniques concentrate on creative modeling and interaction while we focus on structure recovery of highly incomplete, noisy 3D scans with the help of acquired images from the same view. Therefore, instead of actively retrieving suitable parts to match user intent, our solution attempts to match individual parts with the scan data.

Our solution bears some resemblance to primitive fitting used in the context of structure recovery [Wu and Kobbelt 2005; Li et al. 2011a], surface reconstruction [Gal et al. 2007a; Schnabel et al. 2009] or 3D collage assembly [Gal et al. 2007b; Theobalt et al. 2007]. However, the input to these techniques is either complete polygonal meshes [Wu and Kobbelt 2005; Gal et al. 2007b] or dense but noisy point clouds with few holes of big size [Gal et al. 2007a; Schnabel et al. 2009; Li et al. 2011a]. In addition, all these approaches use a predefined set of primitive types (e.g., spheres, cylinders) except for [Gal et al. 2007b; Theobalt et al. 2007], which work on elements as general proxies taken from a given database. In contrast, our parts to be assembled have semantic labels and have context information relative to their parent models in the database.

Our work is also related to example-based shape completion for object classes whose shape spaces are better-defined and can even be parameterized, e.g., human faces [Blanz and Vetter 1999; Weise et al. 2011] and human bodies [Anguelov et al. 2005; Tong et al. 2012]. However, man-made objects we are tackling exhibit a more variety of structures and thus their shape space is challenging to model analytically.

Since its first release in 2010, the Kinect system has attracted great interest from the research community and has been used to acquire shape geometry of human faces [Weise et al. 2011], human bodies [Weiss et al. 2011; Shotton et al. 2011], and indoor scenes [Izadi et al. 2011]. The powerful *KinectFusion* system presented by Izadi et al. fuses live depth data from multiple viewpoints into a single global 3D model of a physical scene in real-time. However, their focus is on the reconstruction of low-level surface geometry instead of high-level structures considered in this paper.

3 Overview

With minimal user intervention, our goal is to recover high-level structures, i.e., *semantic parts with labels*, from a single-view scan of a man-made object acquired by the Kinect system, containing an unorganized 3D point cloud (i.e., 3D scan) and a corresponding RGB image under the same viewpoint. Such high-level structures benefit various applications like structure-preserving modeling or editing [Xu et al. 2011; Zheng et al. 2011; Zheng et al. 2012]. Aiming at a rapid 3D modeling tool, our technique also requires the assembly of the recovered parts to be reasonably close to the underlying geometry.

The same as [Ovsjanikov et al. 2011], we focus on classes of man-made objects (e.g., chairs, tables, airplanes, bicycles etc.) whose shape variability is low dimensional and can be expressed in terms of the relative sizes and positions of shape parts. We assume an available repository of polygonal mesh models that have roughly the same functionality as the input object and have been pre-segmented into semantic parts with corresponding labels (e.g., legs, arms etc.). Our key idea is then to recover the geometry

and structure by part assembly with respect to the acquired data. Using a repository of complete models with labeled semantic parts instead of a repository of individual parts is crucial, since the latter largely demands the pre-segmentation of the acquired data into semantic parts, which is rather challenging given the poor input data. Instead, each complete model in our repository is able to provide a global context for its individual parts and to some extent a possible segmentation of the acquired data, thus making the recovered structure valid.

The acquired point cloud and the corresponding image have complementary characteristics [Li et al. 2011b]. Although the point cloud might be highly incomplete and noisy, it is inherently 3D and thus provides more accurate cues for the underlying geometry and structure when such data is available. In contrast, the image is lack of critical depth information but captures the complete object under the current viewpoint. Considering the complementary traits of the acquired image and 3D scan, we design a novel structure recovery algorithm which consists of the following key stages.

Candidate parts selection. A brute-force part assembly approach is to seek for a best-fit composition among all the possible part compositions via exhaustive search, which is, however, computationally prohibitive (exponential complexity in the number of parts). Therefore, how to identify a small set of candidate parts for each part category is crucial for deriving promising structures efficiently (Section 4.1). This is achieved by matching individual parts in the database with the input 3D scan and its corresponding image. To reflect the fact of low-dimensional shape variation, each part is searched only in a window of small size, determined by its relative position with respect to its parent model that is already globally aligned with the input 3D scan. This allows our algorithm to quickly filter out most of the irrelevant parts and results in a small set of top ranked candidates for each part category, roughly positioned at the desired locations.

Structure composition. This step composes a structure by using a subset of candidate parts for each category (Section 4.2). To achieve this, we first search for candidate compositions of parts that are more likely to form the desired structure. We take into account the geometric fidelity of each part and the interaction between parts (proximity, overlap) in the searching process. A score function is devised to measure the quality of individual compositions and identify the optimal structure composition as the one with the highest score.

Part conjoining. This step is to conjoin the loosely placed parts from the previous step to form a well-connected and visually pleasing model (Section 4.3). We use a global optimization process which first identifies the contact relations between parts and then optimizes the sizes and positions of the parts to form a consistent and complete model. It is driven by the proximity between the currently involved parts and some prior knowledge learned from the repository models.

4 Methodology

This section describes the algorithm and implementation details at each key stage. Our algorithm is fully automatic except for the preprocessing step which needs a moderate amount of user assistance in the image domain.

Preprocessing of shape repository. Our algorithm requires a repository of models with the same class as a man-made object to be acquired. All the models in the repository are pre-segmented into semantic parts with labels (see the supplemental materials). This is achieved by the learning-based segmentation and labeling algorithm [Kalogerakis et al. 2010; Chaudhuri et al. 2011]. We also apply symmetry analysis to detect reflective and rotational symmetries between parts in individual models [Mitra et al. 2006]. Note this preprocessing task needs done only once for each object class.

User-assisted preprocessing of acquired data. As we intend to use the acquired RGB image to provide extra cues for the missing regions in the 3D scan, having the object in the image properly segmented is crucial to the structure recovery process. Although there exist automatic methods for foreground extraction (e.g., [Sun et al. 2007]), for simplicity we let the user interactively cut out the object in the image, for which we use the GrabCut algorithm [Rother et al. 2004]. It results in a binary object mask (Figure 4(left)). Since the image and 3D scan are captured from the same viewpoint, we can use this object mask to easily extract the 3D scan pieces corresponding to the object by checking their image projection against the object mask. Here we assume that the object of interest is not occluded by any other objects in the scene. Otherwise, more user intervention is needed.

4.1 Candidate Parts Selection

This step aims to select a small set of candidate parts for each category of parts. The selection is basically achieved by retrieving parts in the database that fit well *some regions* of the object both in the image and the 3D scan. It is rather challenging to segment the acquired data into meaningful parts given the poor quality of the data. A straightforward fitting solution is then to directly search for the best-fit parts in the repository *over the entire domain* of the 3D scan and/or its corresponding image, which is equivalent to having a repository of individual parts instead of complete models as input (Figure 2). However, this is unlikely to produce good results, as it essentially disregards the semantics associated with each part and the interaction between different parts, without which their composition as a whole would not be a semantically meaningful structure.

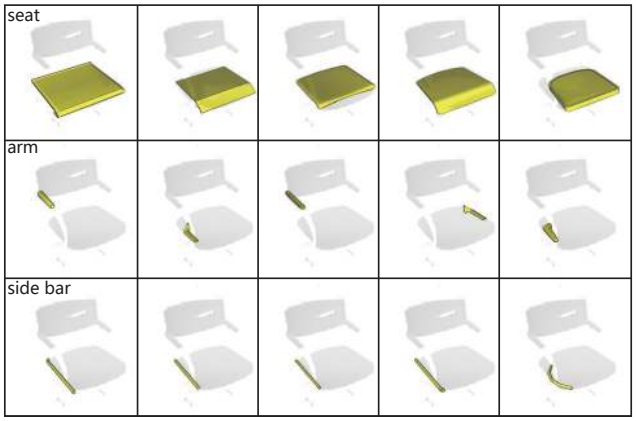

Figure 2: *An example of directly fitting individual parts (using Eq. 3 as the matching metric) to the input without considering their parent models as a global context. Note that poorly placed candidate parts, e.g., most arm candidates and all side bars, would significantly confuse the subsequent composition process.*

Since the objects we are interested in (e.g., chairs, airplanes etc.) lie in a low-dimensional shape space defined with respect to the relative sizes and positions of shape parts, we propose to employ each 3D model in the repository as a global context to constrain the search space of its individual parts for the best fitting to the input. This not only greatly reduces the search space but also implicitly enforces the semantic relation between different parts. This motivated the following two-step approach: matching first each repository model and then its individual parts to the acquired data. Although we are aware of a variety of techniques designed

Figure 3: *Rough alignment of repository models with input scan.*

for either global or partial shape matching (e.g., [Funkhouser et al. 2004; Gal et al. 2007b; Shao et al. 2011; Attene et al. 2011]), the approach briefly described below works well in practice.

As a first step, we align each repository model with the input 3D scan by global matching, though partial matching is also promising. Unlike traditional template-based completion techniques [Pauly et al. 2005], which require accurate matching for shape extrapolation, rough matching is sufficient in our case since it will be refined in the second step of part-based matching and bad candidates will be filtered out in the step of structure composition. This is done by first aligning their upright orientation: all the repository models have their predefined upright orientation [Fu et al. 2008; Laga 2011] and the upright orientation of the input 3D scan is automatically determined by detecting the dominant supporting plane (e.g., the ground plane) in the scene using a RANSAC approach [Schnabel et al. 2007]. Each model is then translated and scaled to fit into the bounding box of the 3D scan. Finally, we determine the remaining degree of freedom, i.e., the orientation of the model around the upright axis by matching the 3D scan with the model under a set of sampled rotations (to minimize the squared distance error), whose rotation axis is determined by the center of the bounding box and the upright orientation. The robustness of the last step is enhanced by *jointly* aligning the repository models with the 3D scan. To achieve this, in the preprocessing step all the repository models are manually aligned to have a consistent frontal orientation. After the orientations around the upright axis are obtained for individual models, we vote for the most frequently used rotation sample, which is then applied to all the models (Figure 3).

Next, we match individual parts of each aligned repository model to the 3D scan and its corresponding image. The use of the acquired image is crucial to address the problem of missing data in the 3D scan. To effectively account for low-dimensional shape variations in terms of the relative positions of shape parts, we search for the best match between each part in the aligned model and the input *only in a small 3D offset window around this part*. Let B denote such window which is centered at the original part centroid, with size $W \times W \times W$. In our experiments, we always set W as 0.1τ, where τ is the diagonal length of the bounding box of the 3D scan. The building block here is a partial shape matching scheme that measures partial shape similarity between a 3D part and both the 3D and 2D region around a given point in the search window, as described below. The design rationale of our similarity is to favor parts that fit the local 3D and 2D region well and meanwhile have high geometric contribution to the input.

We first pre-compute a 3D distance field for the input point cloud by voxelization, denoted as F, and a 2D distance field for the edge map of the image, denoted as G (Figure 4). See the appendix for implementation details. A candidate part centered at a 3D position $\mathbf{p} = [x, y, z]^T$ is also voxelized (by embedding it into a volumetric grid) as V_p, with $V_p(i, j, k) = 1$ on voxels intersecting with it and 0 otherwise. It is then perspectively projected onto the image domain (the focal length of the Kinect system is available) to generate a contour image. Let C_p denote such contour image of the part at position $\mathbf{p}$, with $C_p(i, j) = 1$ on contour and $C_p(i, j) = 0$ otherwise (Figure 4(right)). The *geometric fidelity* score incorporating both the 3D and 2D information is then defined as:

$$S_f(\mathbf{p}) = \frac{1}{2}\left(\frac{\langle V_p, F\rangle}{N_V} + \frac{\langle C_p, G\rangle}{N_C}\right), \quad (1)$$

where N_V is the number of voxels of the candidate part with $V_p(i, j, k) = 1$, N_C is the number of pixels of the contour image with $C_p(i, j) = 1$ and $\langle , \rangle$ is the scalar product. This score measures the ratio of the candidate part covered by the acquired data and thus can penalize parts that protrude too far. Using this score alone might be always in favor of small parts that are matched well locally but have low geometric contribution to the input. However, in practice larger parts are preferred to compose a structure. Therefore, we also consider the extent to which the candidate part contributes to the input, leading to the following *geometric contribution* score:

$$S_c(\mathbf{p}) = \frac{1}{2}\left(\frac{\langle V_p, F\rangle}{N_F} + \frac{\langle C_p, G\rangle}{N_G}\right), \quad (2)$$

where N_F is the number of voxels of the 3D scan with $F(i, j, k) > 0.95$ and N_G is the number of pixels with $G(i, j) > 0.95$. The score S_c measures how the 3D scan and the object contour are covered by the candidate part. The final score that measures the matching quality of a part with both the acquired 3D scan and the image is then defined as:

$$\max_{\mathbf{p}\in B}\{(1-\alpha)\cdot S_f(\mathbf{p}) + \alpha \cdot S_c(\mathbf{p})\}, \quad (3)$$

where B is the 3D search window and α is a weight balancing the influence of geometric fidelity and contribution ($\alpha = 0.7$ in all our examples). Note that accurately evaluating such a similarity score is time consuming, while a coarse score could be enough for this filtering step. We thus only evaluate it at a few sampled positions within the search window. In our experiments, we use $5 \times 5 \times 5$ uniformly sampled positions within the search window in this step, which we find to be a tradeoff between accuracy and efficiency. For the evaluation of Eq. 3, individual parts are translated within the 3D search window around their original position. Rotation and scaling are not applied in our current implementation.

By adopting the above matching strategy, our method quickly compares each part of each model in the repository against the shape of the input object, leading to lists of ranked parts. Suppose that the class of the object we want to model has at most L different categories of parts. For each category of parts in the database, we pick at most top K parts with the highest matching scores as the candidate parts, which will be used in the next step for structure composition. Figure 5 shows several lists of top 5 candidate parts for the target object in Figure 1(a). Note that the acquired image information plays a crucial role in recovering certain parts when

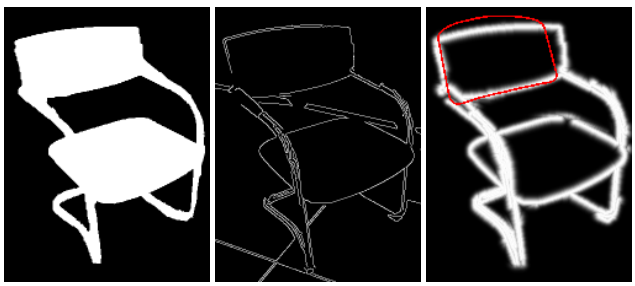

Figure 4: ***Left:*** *Binary object mask.* ***Middle:*** *Edge map.* ***Right:*** *Projected contour C (in red) overlayed on distance field G.*

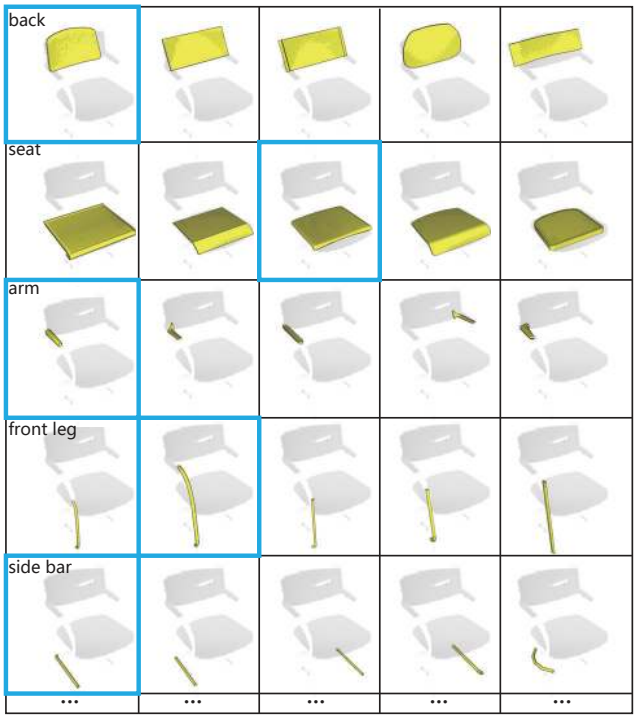

Figure 5: *Lists of top 5 candidate parts for the input in Figure 1(a), in the context of the acquired 3D scan. The parts picked for the final composition are highlighted in blue box.*

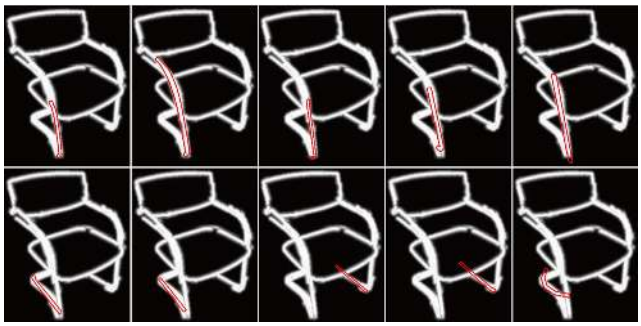

Figure 6: *2D projection of top 5 candidate parts for front leg (top) and side bar (bottom), after position refinement (Section 4.2).*

the corresponding geometry is severely missing, e.g., the front legs and side bars, as shown in Figure 6.

Discussions. Currently the size of search window is fixed (with respect to the 3D scan) and the same for all the categories of parts. However, in practice, different categories of parts might exhibit different degrees of shape variation. It would be interesting to learn the size of search window for each part category from training examples. In addition, we tried to match individual parts at different scales, i.e., to reflect shape variations in terms of relative sizes of shape parts, but found little noticeable improvement in terms of the final structure recovery results in the various examples presented in this paper. This may be due to the fact that a moderate number of repository models can already contain parts with enough scale variation.

4.2 Structure Composition

The previous step quickly filters out most of the irrelevant parts for each category. However, it considers only the matching quality of individual parts to the input and disregards explicit interaction of candidate parts. In this step, we aim to compose the underlying structure by identifying a subset of candidate parts (possibly empty; usually containing one part only) for each category. To achieve this, we first search for promising compositions of candidate parts by mainly examining the interaction between parts and then assess the quality of each composition.

We observed that the parts which can contribute to a promising composition usually satisfy the following geometric constraints:

Geometric fidelity: The parts should match the acquired data reasonably well, i.e. having enough geometric fidelity score.

Proximity: The parts should not be far away from each other. Isolated parts are undesired.

Overlap: The parts should not overlap with each other too much. In other words, the intersection of the parts should be minimized.

We first describe how to quantify the above constraints. The geometric fidelity of a part is defined as $\max_{\mathbf{p}\in B} S_f(\mathbf{p})$ (Eq. 1). The maximization process also refines the position of each part (Figure 5) by exhaustively translating a part to sampled points within the 3D search window and finding the one that maximizes $S_f(\mathbf{p})$. After all the candidate parts are moved to their optimal positions, we measure the proximity between a new part and a set of parts as the nearest distance between it and them. For the overlap constraint, we identify two types of overlap between parts, namely spatial and visual. The spatial one measures the overlap between two candidate parts A and B in 3D, roughly approximated as $\frac{volume(H_A)+volume(H_B)}{volume(H_C)}$, where H_A, H_B and H_C are the convex hulls of A and B, and the union of A and B respectively. The visual overlap is quantified as the ratio of a part covered by other parts after projecting all of them onto the image domain. Note that since we have only a single view as input, these two types of overlap are not necessarily interdependent.

The values of both K and L are typically small enough to allow us to explore large parts of the combinatorial solution space, which we found crucial for deriving interesting structures and variations. We take a *backtracking search* algorithm [Gurari 1999] which incrementally builds composition candidates in a depth first manner and backtracks by removing a part from a partial composition candidate as long as one of the geometric constraints cannot be satisfied. In our implementation, we expand a new part into a partial composition of parts only if all the following conditions are satisfied: the geometric fidelity score is above t_f (typically 0.45); the proximity is below 0.05τ; the visual overlap ratio is below 0.7; and the spatial overlap is below t_s (typically 0.75).

For objects with symmetric parts, we add their symmetric counterpart to a composition candidate during the search process. Our system currently supports reflective and rotational symmetric parts. We use an admittedly less than perfect but simple and efficient strategy to recover the symmetric parts. We assume that the repository model from which the current part of interest comes roughly shares the same reflective plane and rotational axis with the input scan after pose alignment (Section 4.1). Symmetric parts are then recovered by reflecting/rotating the current part along that the roughly aligned reflective plane/rotational axis. The number of rotational parts is automatically determined by enumerating (3-6) and choosing the one with the maximum average geometric fidelity score (Eq. 1). Such simple strategy might lead to some position deviation of symmetric parts but can be resolved in the subsequent part conjoining step.

Once a complete candidate composition $\mathbb{C} = \{P_1, P_2, ..., P_n\}$ is formed, we use a global matching score to evaluate the over-

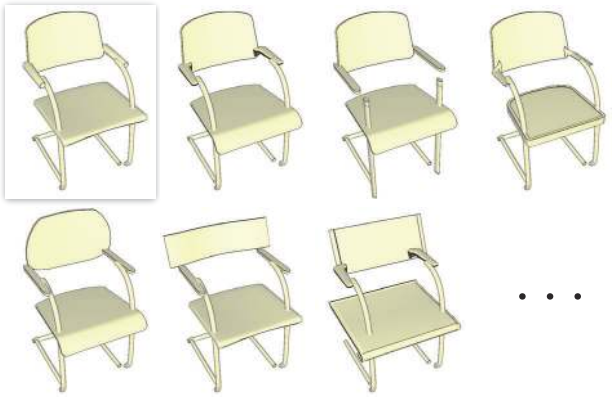

Figure 7: *From left to right, top to bottom: candidate compositions with decreasing global evaluation scores.*

all quality of the composition. The design rationale here is to measure the overall geometric fidelity and contribution of the composed structure, while also favoring compositions with locally fine matched parts. Specifically it is formulated as follows:

$$E(\mathbb{C}) = \frac{1}{n}\sum_i S_f^{P_i} + \beta \cdot S_f^{\mathbb{C}} + \gamma \cdot S_c^{\mathbb{C}}. \quad (4)$$

The first term measures the average geometric fidelity of individual parts, while the second and third terms are the geometric fidelity and contribution scores of the globally composed model from $\mathbb{C}$ (cf. Eq. 1 and Eq. 2). We always use the weights $\beta = 1$ and $\gamma = 1$ in our experiments. We identify the optimal structure composition as the one with the highest score of $E(\mathbb{C})$ (Figure 7(top left)) among all the candidate compositions found (Figure 7). Note that since multiple candidate parts tend to compete with each other in the search of candidate compositions, the optimal composition might not necessarily consist of the highest ranked candidate parts (Figure 5).

4.3 Part Conjoining

Although the interaction between parts is employed in the previous step, the parts are still loosely placed together (Figure 7(top left)). This step aims to conjoin all the parts to form a well-connected and visually-pleasing model. This is achieved by first identifying contact parts as well as contact points and then refining the scales and positions of the individual parts to reflect their underlying mutual relations. For simplicity, we do not involve the input image and 3D scan to guide the conjoining process, since we believe that once a consistent, complete model is constructed, it is straightforward to deform it to fit the acquired data for instance using a variant of the photo-inspired approach by Xu et al. [2011].

We first identify pairs of parts that are likely to contact each other. This is addressed from two aspects. First, we use the proximity between a pair of parts: two parts $P_i, P_j \in \mathbb{C}$ are possible to contact with each other only if they are already close enough. This leads to an indicator function δ_{ij}, with $\delta_{ij} = 1$ if two parts are in close proximity and $\delta_{ij} = 0$ otherwise. Note that simply judging the contact relation from such proximity is not reliable for objects with cluttered components like bicycles. Thus we also employ the prior knowledge from the repository models. Let L_i and L_j be the part labels for P_i and P_j, respectively. It is reasonable to assume that two parts P_i and P_j are more likely to contact each other if the parts with label L_i and those with label L_j in the repository are in frequent contact with each other. To quantify this, we pre-compute the contact relations of different part categories in the repository and count their frequency, leading to a set of contact

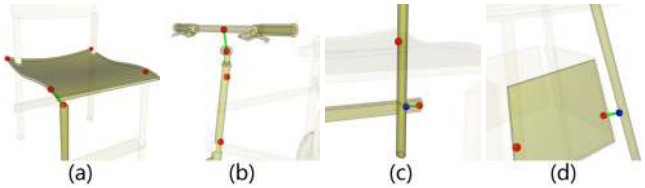

Figure 8: *Examples of primary contact points (in red) and secondary contact points (in blue).*

confidence weights $\omega_{ij} \in [0, 1]$, which will be used shortly in a global optimization (Eq. 5).

Next we establish pairs of contact points between a pair of parts P_i and P_j with $\delta_{ij} = 1$. Again, by inspecting their corresponding repository models, most of such contact points can be found [Jain et al. 2012], which we call as the *primary* contact points (Figure 8(a, b)). We then match pairs of the primary contact points between P_i and P_j using a greedy strategy, in which we pick up pairs of the closest primary contact points sequentially until there exists no pair of primary contact points within a distance threshold t_d (typically chosen as 0.05τ). Afterwards, we allow the remaining unmatched primary points to match with their nearest points on the other part if their distances are below t_d. In an extreme case where even such matching cannot be found, we simply use the pair of the nearest points between the two parts to connect them. We call all the newly generated nearest points on the parts as the *secondary* contact points (Figure 8(c, d)). To enforce symmetry, we symmetrize the contact points from the parts with self-reflective symmetry or among the parts with reflective/rotational symmetry [Mitra et al. 2007]. Let $\{(\mathbf{p}^i_{m_1}, \mathbf{p}^j_{n_1}), (\mathbf{p}^i_{m_2}, \mathbf{p}^j_{n_2}), ...\}$ denote the resulting set of pairs of contact points between P_i and P_j (Figure 9(left)).

Our final goal is to adjust the sizes and positions of the parts to make the identified pairs of contact points meet each other as much as possible. In other words, we need to solve for the scale factor $\mathbf{s}_i = [s^i_1, s^i_2, s^i_3]^T$ (allowing anisotropic scaling) and the translation vector $\mathbf{t}_i = [t^i_1, t^i_2, t^i_3]^T$ for each part P_i along its principle axis. Let $\mathbf{c}_i$ be the center of P_i and $\mathbf{Q}_i = [\mathbf{v}^i_1, \mathbf{v}^i_2, \mathbf{v}^i_3]$ the matrix composed of the principal axes of P_i via PCA. The transformed position of a contact point $\mathbf{p}^i_k$ on part P_i can then be expressed as $T(\mathbf{p}^i_k) = \mathbf{Q}_i\mathbf{\Lambda}_i\mathbf{Q}_i^T(\mathbf{p}^i_k - \mathbf{c}_i) + \mathbf{c}_i + \mathbf{t}_i$, where $\mathbf{\Lambda}_i = diag(s^i_1, s^i_2, s^i_3)$. Taking the pre-computed confidence into account, the goal of bringing the pairs of contact points together leads to the following contact enforcement energy:

$$E_c = \sum_{i,j}\delta_{ij}\omega_{ij}\sum_k \|T(\mathbf{p}^i_{m_k}) - T(\mathbf{p}^j_{n_k})\|^2. \quad (5)$$

We add another two shape preserving terms to avoid drastic changes during adjustment:

$$E_s = \sum_i \|\mathbf{s}_i - \mathbf{o}\|^2 \qquad E_t = \sum_i \|\mathbf{t}_i\|^2, \quad (6)$$

where $\mathbf{o} = [1, 1, 1]^T$. The new scale and translation of the parts after adjustment are then obtained as:

$$\text{argmin}_{\{\mathbf{s}_i, \mathbf{t}_i\}}\omega_c \cdot E_c + \omega_s \cdot E_s + \omega_t \cdot E_t, \quad (7)$$

which leads to a linear least-squares minimization problem and can be efficiently solved. We use the weights $\omega_c = 100$, $\omega_s = 10$, and $\omega_t = 1$ in our experiments. Figure 9(right) shows the conjoined parts after such global optimization, forming a cohesive model. Note that independently, Kalogerakis et al. [2012] describe a similar global optimization but with a different goal (i.e., open-ended shape synthesis) from ours. One of the differences in terms of the implementation is that we use two types of contact points (primary and secondary) while their approach ensures the matching of primary contract points (called as slots) between parts only. We

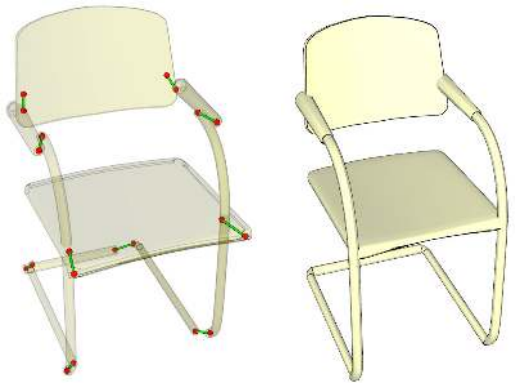

Figure 9: ***Left:*** *Identified contact points of loosely placed parts and their desired connections.* ***Right:*** *Conjoined parts.*

found that having such secondary contact points (Figure 8) is more flexible for our purpose.

5 Results and Discussion

In this section, we show a variety of structure recovery results generated using our part assembly approach. The input objects were all captured using the standard Kinect system of Microsoft, which provides a RGB image and a corresponding depth image both in the resolution of 640×480. Our method was tested on an Intel Core 2 Duo 3GHz computer with 4GB RAM. The candidate parts selection step took around 1 minute in total for a database with 70 models. We chose the number of candidates $K = 5$ in all our examples and the structure composition step took around 2 minutes in total under such configuration. The part conjoining step took less than 1 second. Although our method involves a few parameters, most of them always remain fixed as described in each section. Only the parameters t_s, t_f (Section 4.2) and t_d (Section 4.3) might need certain adjustment. The spatial overlap threshold t_s is decreased to avoid excessive tiny parts for objects like tables. The geometric fidelity threshold t_f is reduced for captured data with poorer quality (e.g. Figure 12(c)). The distance threshold t_d is increased to encourage using primary contact points for objects like bicycles.

We have applied our method to typical daily objects that can be quickly captured using the Kinect system. Similar to [Ovsjanikov et al. 2011], we mainly focus on four categories: chairs, tables, bicycles and airplane models, since it is well known that they usually exhibit large variations in both shape and structure. Therefore, their global structures cannot be well captured without a large database of models. For example, for our chair dataset that contains 70 chair models of different styles (see the supplemental material), it is unlikely that a chair with the same global structure as the input could be found. Figure 10(a) shows a list of chairs from the repository which are structurally the most similar to the input in Figure 1(a)(left), using the measurement in Eq. 4 followed by our manual checking. A similar list of airplanes for the input in Figure 1(a)(right) is shown in Figure 10(b). Although these models bear certain local similarity to the input one, their global structures are still different, reiterating the importance of structure recovery by part assembly. Figure 11 also shows two more apparent examples where new structures are constructed by assembling existing parts from the repository models. For example, although the H-type base of the chair in Figure 11(b) does not exist in the repository, it can be synthesized by our approach with respect to the scan data. Note that our implementation currently focuses only on part-level structures. To recover novel structures at the level of sub-components, a possible solution is to encode existing structures and generate new structures both in a hierarchical manner.

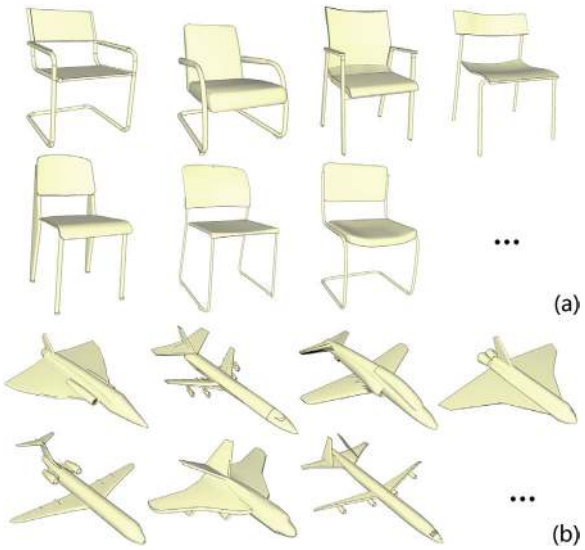

Figure 10: *A list of repository models roughly ordered by their structural similarity to the input ones in Figure 1(a) and Figure 1(b) respectively.*

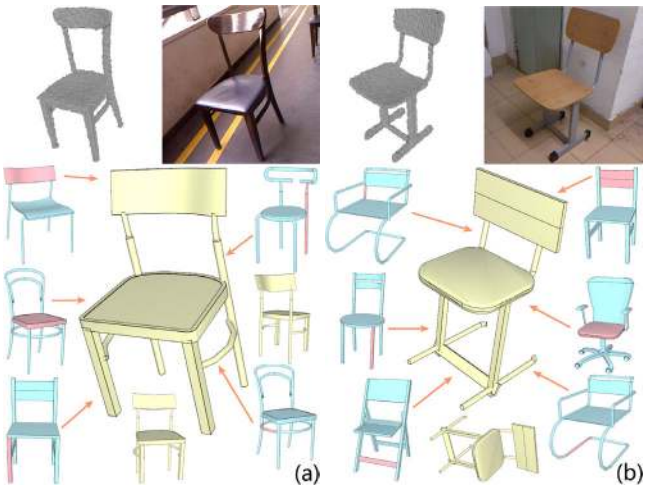

Figure 11: *Examples of constructing new structures (yellow). Parts borrowed from different repository models (blue) are highlighted (red). Note that the H-type base of the chair in (b) does not exist in the repository and is synthesized by our part assembly approach.*

To enable our part assembly approach, each model in the chair dataset is segmented into 5-12 semantically meaningful parts. The parts are also labeled and classified into 11 categories, including seat, back, arm, etc. Figure 1(a) shows a challenging but typical example with large missing data in the 3D scan due to the single-view capturing and the highly reflective material of the legs. Its seat, back and arms are recovered mainly owing to the 3D point cloud information, while the legs and side bars are recovered under the guidance of the image contour. These parts are tightly conjoined together to form a well-structured model. The recovered structure for this example borrows parts from 5 different models. Figure 12(a-e) shows more structure recovery results of chairs, all of which are automatically generated except for a small amount of user interaction for segmenting out the objects in the captured images (taking a couple of minutes). These examples exhibit different structures, which do not exist in the repository (by our

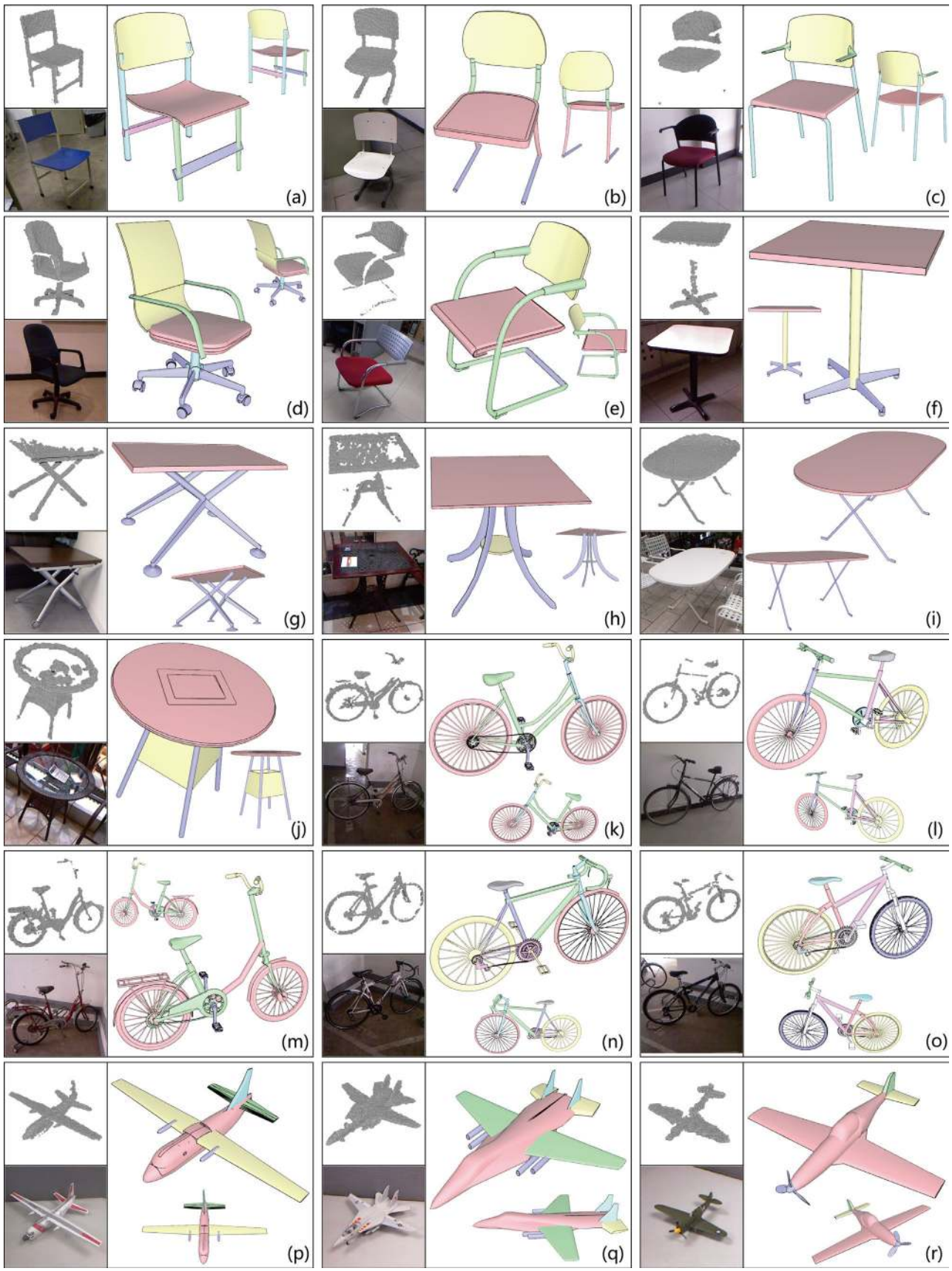

Figure 12: *The result gallery generated using our part assembly approach for structure recovery from a single-view scan of man-made objects acquired by the Kinect system. Parts borrowed from different repository models are shown in different colors.*

manual checking) but can be faithfully recovered by our approach. Figure 12(e) shows the same object as Figure 1 but captured from another view. Our approach is able to generate roughly the same structure, showing that our approach is largely insensitive to the capturing viewpoint. Note that for such objects containing scanner-unfriendly materials, simply capturing the object from more viewpoints would not help too much. Instead, our approach is able to recover entirely missing parts with the help of the exemplar parts and the acquired image information.

To facilitate structure recovery for tables, we prepare a dataset containing 61 tables of different styles (see supplemental), each of which is segmented into labeled parts. There are in total 4 part categories in this dataset. Figure 12(f-j) shows several recovery examples. The scanned table in Figure 12(f) is highly contaminated in the lower half. With the aid of 2D and partial 3D information as well as the dataset, the underlying structure still gets faithfully recovered. Figure 12(g-j) shows other recovered tables, whose structures also differ from the repository models. Figure 12(j) is a typical case where the desktop region is largely missing due to its glassy surface, which can be recovered using our approach by assembling suitable parts from 3 different models. This is also a typical example with symmetric components (i.e., the legs and fences), which are successfully recovered by our technique.

Our repository for bicycles contains only 38 bicycle models of different styles with totally 9 part categories (see supplemental). Figure 12(k-o) gives several examples of captured bicycles with various distributions of missing data. Our part assembly approach successfully brings high-detailed components to the recovered shapes despite the poor quality of both the acquired images and 3D point clouds. Note that most handles of the bicycles are severely contaminated in the 3D scans of these examples (e.g., Figure 12(n)) and are mainly recovered by the acquired image information. In addition, since the repository models happen to contain pedals with various orientations, our method successfully finds the suitable ones to assemble. A more general solution might be to analyze the degrees of freedom of such parts (e.g. rotational parts) in the repository and explicitly exploit such information in the matching process.

Finally, the airplane dataset contains 70 airplanes, with 6 part categories in total (see supplemental). Figure 1(b) and Figure 12(p-r) show the structure recovery results of several airplane models. These airplane models also exhibit quite different structures from the repository models. Figure 1(b) gives another typical example of scanner-unfriendly materials. The engines of that airplane model are entirely missing due to their dark surfaces and are also mainly recovered with the help of the acquired image information. A similar example is the propeller of the airplane model in Figure 12(r), which fails to be captured by the Kinect but is recovered by our approach.

Limitations. Our algorithm is largely based on the assumption that shape variations within a certain class of models can be explained in terms of the relative sizes and positions of the shape parts. Thus our method might fail for objects whose shape variations cannot be characterized by the spatial layout of the parts (e.g., facades with irregular structures).

Second, our focus is on structure recovery instead of fine geometry reconstruction. Therefore, although the resulting models are already geometrically similar to the underlying shape, it still exhibits noticeable shape difference. However, we believe that this problem can already be addressed by existing non-rigid structure-preserving deformation techniques like [Xu et al. 2011].

Third, texture edges in the acquired image might be misunderstood as shape edges especially when the corresponding 3D scan is severely missing, causing undesired candidate parts, though the problem might be resolved by the step of structure composition.

Fourth, due to the low resolution of the input scan data, it might be challenging for our part matching method to discriminate parts that are close to each other. For example, as shown in Figure 13(a), the three slim pillars in the middle of the table are mismatched to a single bigger pillar, while more suitable parts do exist in the repository. Besides, the performance of our technique may degrade for the input with severely missing geometry. Figure 13(b) shows such extreme case, where global alignment in the first step is not reliable, misleading the subsequent steps and causing wrong interpretation of the underlying structure. It can be possibly addressed by adopting partial shape matching or relying more on the image information in the alignment step.

Lastly, although our bottom-up approach allows a much more effective reuse of existing 3D models, like the other example-based approaches, it is still limited by the availability of exemplar parts. Figure 13(c) shows a less satisfactory result where the geometry of the highly curved arms and the cylindrical back is poorly recovered due to the lack of corresponding parts in the repository. Some uncommon structures may not be recovered well either if there is no proper combination of parts in the database. To better illustrate how our method behaves with a decreasing number of repository models, we randomly pick 40, 20, 10, 5 out of the original 70 chair models and use them as the new repository for the input in Figure 12(e). Such process is repeated for 4 times and all the generated results are collected in Figure 14. As expected, the richer the database, the more faithful reconstruction we can achieve. On the contrary, the smaller the database, the lower the chance that a desired result can be synthesized. The randomly generated results also tend to be less stable for a repository of smaller size.

Figure 13: *Less successful examples. (a) Due to limited resolution, our part matching method fails to discriminate the three slim pillars that are close to each other. (b) The input with severely missing geometry can give rise to wrong interpretation of the underlying structure. (c) The geometry is poorly recovered due to the lack of suitable parts in the repository.*

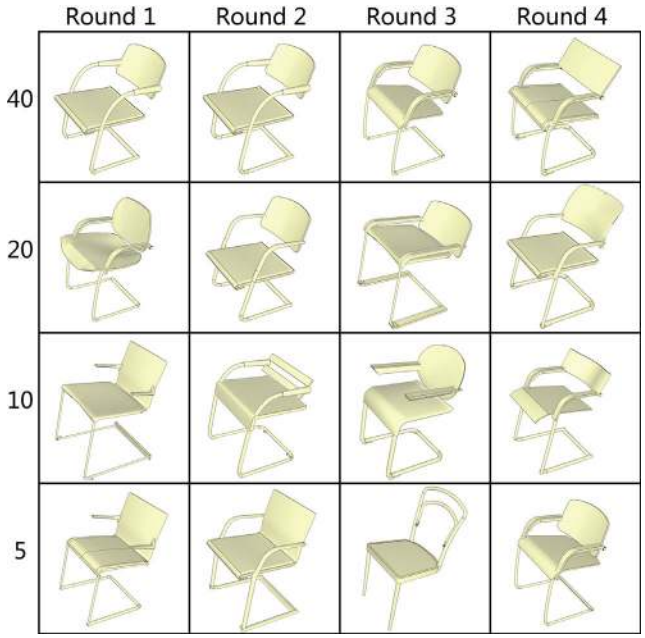

Figure 14: *Results under random subsets of the original repository. The size decreases from 40 to 5 (from top row to bottom row). Such random process is repeated for 4 times (from left column to right column).*

6 Conclusion

We have presented a bottom-up structure recovery approach based on part assembly. Our approach effectively reuses a limited number of existing 3D models to compose new structures automatically adapted to the underlying object. The assembly process is guided by the spatial layout of the parts in the repository models, allowing us to quickly explore large parts of the exponential space formed by part assembly, which correspond to semantically meaningful structures. With only a shape repository of small size, our approach is able to faithfully recover a variety of structures of man-made objects from their single-view scanning acquisition by the Kinect system.

As a future work, we are interested in extending our technique to multi-view inputs, which can not only reduce the amount of user interaction for cutting out the object of interest (via background subtraction) but also alleviate the ambiguous problem. In our current system, the structure composition step is solely dependent on the step of candidate parts selection. It would be interesting to investigate the interdependence of these two steps, which might allow adaptive adjustment of the search window of parts with respect to the existing ones. Our structure composition step relies on geometric constraints only. We are interested in including style or functional constraints to make the final shape a more coherent and realistic model [Xu et al. 2012; Kalogerakis et al. 2012]. Our current algorithm mainly focuses on individual objects. This is only the first step towards our ultimate goal of automatic reconstruction of indoor scenes.

Acknowledgements

We thank the anonymous reviewers for their constructive comments. This work was partially supported by grants from the National Basic Research Project of China (Project No. 2011CB302203), the Natural Science Foundation of China (Project No. 61120106007), the National High Technology Research and Development Program of China (Project No. 2012AA011801), the Research Grants Council of HKSAR (Project No. 113610), and the City University of Hong Kong (Project No. SRG7002776).

References

ANGUELOV, D., SRINIVASAN, P., KOLLER, D., THRUN, S., RODGERS, J., AND DAVIS, J. 2005. SCAPE: shape completion and animation of people. *ACM Trans. Graph. 24*, 3, 408–416.

ATTENE, M., MARINI, S., SPAGNUOLO, M., AND FALCIDIENO, B. 2011. Part-in-whole 3D shape matching and docking. *The Visual Computer 27*, 11, 991–1004.

ATTENE, M. 2010. A lightweight approach to repairing digitized polygon meshes. *The Visual Computer 26*, 11, 1393–1406.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3D faces. In *SIGGRAPH '99*, 187–194.

CANNY, J. 1986. A computational approach to edge detection. *IEEE Trans. Pattern Anal. Mach. Intell. 8*, 679–698.

CHAUDHURI, S., AND KOLTUN, V. 2010. Data-driven suggestions for creativity support in 3D modeling. *ACM Trans. Graph. 29*, 6, 183:1–183:10.

CHAUDHURI, S., KALOGERAKIS, E., GUIBAS, L., AND KOLTUN, V. 2011. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. Graph. 30*, 6, 35:1–35:10.

FU, H., COHEN-OR, D., DROR, G., AND SHEFFER, A. 2008. Upright orientation of man-made objects. *ACM Trans. Graph. 27*, 3, 42:1–42:7.

FUNKHOUSER, T., KAZHDAN, M., SHILANE, P., MIN, P., KIEFER, W., TAL, A., RUSINKIEWICZ, S., AND DOBKIN, D. 2004. Modeling by example. *ACM Trans. Graph. 23*, 3, 652–663.

GAL, R., SHAMIR, A., HASSNER, T., PAULY, M., AND COHEN-OR, D. 2007. Surface reconstruction using local shape priors. In *SGP '07*, 253–262.

GAL, R., SORKINE, O., POPA, T., SHEFFER, A., AND COHEN-OR, D. 2007. 3D collage: expressive non-realistic modeling. In *NPAR '07*, 7–14.

GURARI, E., 1999. Cis 680: Data structures: Chapter 19: Backtracking algorithms.

HUANG, Q., KOLTUN, V., AND GUIBAS, L. 2011. Joint shape segmentation with linear programming. *ACM Trans. Graph. 30*, 6, 125:1–125:12.

IZADI, S., KIM, D., HILLIGES, O., MOLYNEAUX, D., NEWCOMBE, R., KOHLI, P., SHOTTON, J., HODGES, S., FREEMAN, D., DAVISON, A., AND FITZGIBBON, A. 2011. KinectFusion: real-time 3D reconstruction and interaction using a moving depth camera. In *UIST '11*, 559–568.

JAIN, A., THORMÄHLEN, T., RITSCHEL, T., AND SEIDEL, H.-P. 2012. Exploring shape variations by 3d-model decomposition and part-based recombination. *Comp. Graph. Forum 31*, 2.

KALOGERAKIS, E., HERTZMANN, A., AND SINGH, K. 2010. Learning 3D mesh segmentation and labeling. *ACM Trans. Graph. 29*, 4, 102:1–102:12.

KALOGERAKIS, E., CHAUDHURI, S., KOLLER, D., AND KOLTUN, V. 2012. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph. 31*, 4, 55:1–55:11.

KRAEVOY, V., AND SHEFFER, A. 2005. Template-based mesh completion. In *SGP '05*, 13–22.

LAGA, H. 2011. Data-driven approach for automatic orientation of 3D shapes. *The Visual Computer 27*, 11, 977–989.

LEE, J., AND FUNKHOUSER, T. 2008. Sketch-based search and composition of 3D models. In *EUROGRAPHICS Workshop on Sketch-Based Interfaces and Modeling*.

LI, Y., WU, X., CHRYSATHOU, Y., SHARF, A., COHEN-OR, D., AND MITRA, N. J. 2011. GlobFit: consistently fitting primitives by discovering global relations. *ACM Trans. Graph. 30*, 4, 52:1–52:12.

LI, Y., ZHENG, Q., SHARF, A., COHEN-OR, D., CHEN, B., AND MITRA, N. J. 2011. 2D-3D fusion for layer decomposition of urban facades. In *ICCV '11*.

LIN, J., JIN, X., AND WANG, C. C. L. 2010. Fusion of disconnected mesh components with branching shapes. *The Visual Computer 26*, 6-8, 1017–1025.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3D geometry. *ACM Trans. Graph. 25*, 3, 560–568.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2007. Symmetrization. *ACM Trans. Graph. 26*, 3, 63:1–63:8.

OVSJANIKOV, M., LI, W., GUIBAS, L., AND MITRA, N. J. 2011. Exploration of continuous variability in collections of 3D shapes. *ACM Trans. Graph. 30*, 4, 33:1–33:10.

PAULY, M., MITRA, N. J., GIESEN, J., GROSS, M., AND GUIBAS, L. J. 2005. Example-based 3D scan completion. In *SGP '05*.

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

SCHNABEL, R., WAHL, R., AND KLEIN, R. 2007. Efficient RANSAC for point-cloud shape detection. *Computer Graphics Forum 26*, 2, 214–226.

SCHNABEL, R., DEGENER, P., AND KLEIN, R. 2009. Completion and reconstruction with primitive shapes. *Computer Graphics Forum 28*, 2, 503–512.

SHALOM, S., SHAMIR, A., ZHANG, H., AND COHEN-OR, D. 2010. Cone carving for surface reconstruction. *ACM Trans. Graph. 29*, 6, 150:1–150:10.

SHAO, T., XU, W., YIN, K., WANG, J., ZHOU, K., AND GUO, B. 2011. Discriminative sketch-based 3D model retrieval via robust shape matching. *Computer Graphics Forum 30*, 7, 2011–2020.

SHARF, A., ALEXA, M., AND COHEN-OR, D. 2004. Context-based surface completion. *ACM Trans. Graph. 23*, 3, 878–887.

SHEN, C.-H., ZHANG, G.-X., LAI, Y.-K., HU, S.-M., AND MARTIN, R. R. 2010. Harmonic field based volume model construction from triangle soup. *Journal of Computer Science and Technology 25*, 3, 562–571.

SHIN, H., AND IGARASHI, T. 2007. Magic canvas: interactive design of a 3-D scene prototype from freehand sketches. In *GI '07*, 63–70.

SHOTTON, J., FITZGIBBON, A., COOK, M., SHARP, T., FINOCCHIO, M., MOORE, R., KIPMAN, A., AND BLAKE, A. 2011. Real-time human pose recognition in parts from single depth images. In *CVPR '11*, 1297–1304.

SIDI, O., VAN KAICK, O., KLEIMAN, Y., ZHANG, H., AND COHEN-OR, D. 2011. Unsupervised co-segmentation of a set of shapes via descriptor-space spectral clustering. *ACM Trans. Graph. 30*, 6, 126:1–126:10.

SUN, J., KANG, S., XU, Z., TANG, X., AND SHUM, H. 2007. Flash cut: Foreground extraction with flash and no-flash image pairs. In *CVPR'07*, 1–8.

THEOBALT, C., RÖESSL, C., DE AGUIAR, E., AND SEIDEL, H.-P. 2007. Animation collage. In *SCA '07*, 271–280.

TONG, J., ZHOU, J., LIU, L., PAN, Z., AND YAN, H. 2012. Scanning 3D full human bodies using kinects. *IEEE Transactions on Visualization and Computer Graphics 18*, 4, 643–650.

WEISE, T., BOUAZIZ, S., LI, H., AND PAULY, M. 2011. Realtime performance-based facial animation. *ACM Trans. Graph. 30*, 4, 77:1–77:10.

WEISS, A., HIRSHBERG, D., AND BLACK, M. J. 2011. Home 3D body scans from noisy image and range data. In *ICCV '11*.

WU, J., AND KOBBELT, L. 2005. Structure recovery via hybrid variational surface approximation. *Computer Graphics Forum 24*, 3, 277–284.

XU, K., ZHENG, H., ZHANG, H., COHEN-OR, D., LIU, L., AND XIONG, Y. 2011. Photo-inspired model-driven 3D object modeling. *ACM Trans. Graph. 30*, 4, 80:1–80:10.

XU, K., ZHANG, H., COHEN-OR, D., AND CHEN, B. 2012. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Transactions on Graphics 31*, 4, 57:1–57:10.

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

ZHENG, Y., FU, H., COHEN-OR, D., AU, O. K.-C., AND TAI, C.-L. 2011. Component-wise controllers for structure-preserving shape manipulation. *Computer Graphics Forum 30*, 2, 563–572.

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

Appendix: 2D and 3D Distance Field Computation

To pre-compute a 3D distance field of an input point cloud, we first embed it into a volumetric grid within its bounding box. The distance field F is then calculated as $F(i,j,k) = e^{-d^2/2{\sigma_V}^2}$, where d is the distance from voxel (i,j,k) to the nearest point in the 3D point cloud. The voxel size is set to be 0.005τ in each dimension, where τ is the diagonal length of the bounding box. σ_V is always set to be 0.01τ in our experiments.

Similarly, we pre-compute another distance field on the image domain to measure the matching quality of the part with the image. We first detect edges in the input image [Canny 1986], generating an edge map (Figure 4 (middle)). Its 2D distance field G is then computed (Figure 4 (right)), with $G(i,j) = e^{-d_1^2/2{\sigma_I}^2} \cdot e^{-d_2^2/2{\sigma_I}^2}$, where d_1 is the distance from pixel (i,j) to the nearest edge point and d_2 is the distance to the nearest non-zero point of the binary object mask. The second term is used to filter out edges that do not belong to the object in a more continuous (fuzzy) way. We use $\sigma_I = 0.01\mu$ in all the examples of the paper, where μ is the diagonal length of the object in the image.