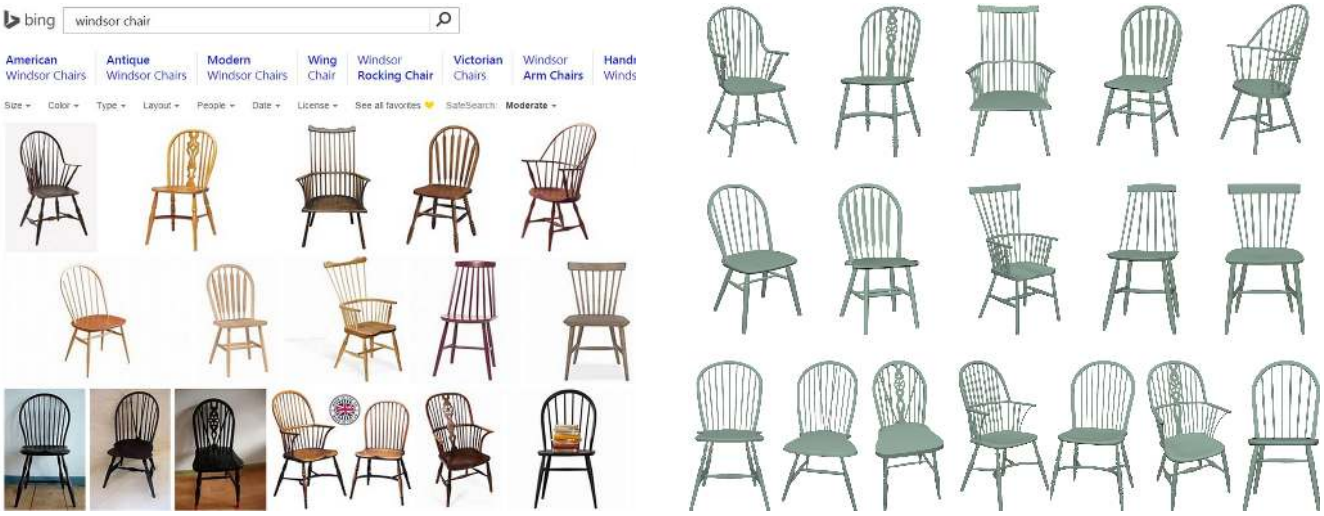
Single-View Reconstruction via Joint Analysis of Image and Shape Collections

Qixing Huang Hai Wang Vladlen Koltun
Toyota Technological Institute at Chicago Intel Labs



Web image search Reconstructed 3D models

Figure 1: *Our approach reconstructs objects depicted in images, even if each object is only shown in a single image. Left: Web image search for "windsor chair", first page results. Right: 3D models automatically generated by our approach for these images.*

Abstract

We present an approach to automatic 3D reconstruction of objects depicted in Web images. The approach reconstructs objects from single views. The key idea is to jointly analyze a collection of images of different objects along with a smaller collection of existing 3D models. The images are analyzed and reconstructed together. Joint analysis regularizes the formulated optimization problems, stabilizes correspondence estimation, and leads to reasonable reproduction of object appearance without traditional multi-view cues.

CR Categories: I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling

Keywords: single-view reconstruction, image-based modeling

1 Introduction

Can we create 3D models of all objects in the world? High-fidelity models can be obtained from range scans and dense multi-view datasets, but acquiring such data for millions of objects demands substantial time and labor. Can large repositories of 3D models be created using existing data that is already present on the Web?

Web images have been used to reconstruct landmark scenes, which are densely sampled by thousands of visitors [Snavely et al. 2010]. But what about the objects that populate our daily lives? Millions are already depicted on the Web. Can we exploit the regularity of object shapes [Kalogerakis et al. 2012] to reconstruct an object even if it appears in just a single image, thereby paving the way to large repositories of 3D models created by mining the Web?

In this paper, we present an approach to creating 3D models of objects depicted in images, even if each object is only shown in a single image. Our approach uses a comparatively small collection of existing 3D models to guide the reconstruction process. A key challenge is that these existing 3D models may only sparsely sample the underlying shape space. The number of high-fidelity shapes in existing 3D model repositories, such as the 3D Warehouse, is much smaller than the number of images on the Web. For many families of objects, high-quality images vastly outnumber high-quality 3D models. Thus simply retrieving the most similar existing 3D model for each input image does not yield satisfactory results: even if such retrieval is performed reliably, the closest pre-existing model is often quite different from the depicted object. We therefore develop an assembly-based approach that reconstructs objects by composing parts from pre-existing shapes.

A key idea in the presented pipeline is to jointly analyze the images and the 3D models. First, camera poses for all images are estimated by optimizing a global objective that measures the consistency of estimated poses across similar images. Then, a global network of dense pixel-level correspondences is established between natural images and rendered images. These correspondences are used to jointly segment the images and the 3D models. The computed segmentations and correspondences are used to construct new models, which are then optimized.

(a) Input images and shapes (b) Camera pose estimation (c) Correspondences (d) Segmentation (e) Reconstruction

Figure 2: *Stages of the presented approach. (a) Input: natural images and pre-existing 3D models. (b) Camera pose estimation for natural images, visualized by showing pre-existing models from the estimated poses. (c) Correspondence structure via a dense network of patches that interconnect natural images and rendered images. (d) Joint image segmentation. (e) 3D reconstruction of each natural image using parts from pre-existing models.*

We evaluate the presented approach on a number of datasets collected from the Web. The accuracy of the approach is validated through extensive experiments. Figure 1 shows results on Windsor chairs.

2 Related Work

Images are easy to acquire and provide direct information on object appearance. However, a single image provides only limited geometric information. To obtain veridical 3D models, additional input is generally required. Such input can take the form of human assistance, multiple images, pre-existing models, simplifying assumptions, or a combination thereof.

An early image-based modeling system that utilized human assistance was developed by Debevec et al. [1996], who focused on architectural modeling. More recently, Xu et al. [2011] presented a user-guided approach that deformed stock 3D models to a given image; human assistance was used to segment the image and establish correspondences between the image and the models. The approaches of Zheng et al. [2012] and Chen et al. [2013] let users interactively fit cuboids and generalized cylinders to objects in images. The system of Kholgade et al. [2014] let the user align a stock 3D model to a photograph, thus enabling advanced image editing. In contrast to all of these systems, our approach is automatic: ambiguities that arise in considering a single image are resolved through joint analysis of an image collection along with pre-existing 3D models, and by enforcing structural relationships such as symmetry and adjacency.

Three-dimensional models can be automatically created from an image collection that densely samples the appearance of a single static object or environment. The geometry of image formation can be used to interconnect the collection with correspondences, which can be used to reconstruct the depicted object or environment [Hartley and Zisserman 2000]. This approach has been impressively applied to reconstruct a variety of public spaces from Web images [Snavely et al. 2010]. Our work also uses Web images, but our images depict many different objects. Each object may have a distinct shape and may only appear in a single image.

Estimation of three-dimensional layout from a single image can be performed using projective geometry techniques [Criminisi et al. 2000] or by data-driven approaches [Fouhey et al. 2013; Su et al. 2014; Eigen et al. 2014]. In contrast to these techniques, our approach reconstructs complete 3D models, including surfaces that are occluded in input images. A key technical difference between our work and these prior approaches is that instead of processing each image in isolation, we analyze a whole collection of images jointly. Our experiments demonstrate that joint reconstruction of the image collection produces much better results than treating each image separately.

A number of recent works consider object reconstruction from image collections [Carreira et al. 2015b; Carreira et al. 2015a; Kar et al. 2015; Averbuch-Elor et al. 2015]. These approaches do not use available 3D models and produce only coarse three-dimensional proxies. Our approach analyzes images and shapes together. Although our shape collections are comparatively small and do not lend themselves to simple retrieval approaches, they provide valuable geometric information. Proper use of this data is at the heart of our approach.

A number of techniques analyze shape collections, producing consistent segmentations and correspondences [Huang et al. 2011; Kim et al. 2012]. In contrast, our work is focused on joint analysis of images and shapes for the purpose of 3D reconstruction. Shen et al. [2012] describe a pipeline that assembles a 3D model to fit a given depth image. In contrast, our pipeline is designed to reconstruct objects depicted in regular images.

3 Overview

Input. The presented pipeline jointly analyzes a collection of images $\mathcal{I} = \{I_1, \cdots, I_{N_\mathcal{I}}\}$ and a collection of shapes $\mathcal{S} = \{S_1, \cdots, S_{N_\mathcal{S}}\}$, all depicting objects from a common category, such as chairs or bicycles. For objects we consider, high-quality images are considerably more numerous than high-quality 3D models. Thus $N_\mathcal{I}$ is much larger than $N_\mathcal{S}$.

As in prior work on 3D reconstruction from image collections [Carreira et al. 2015b], we assume that a bounding box of the object in each image is provided. Given the bounding boxes, each image is automatically cropped and then scaled to 500 pixels in width or height, whichever is larger. Henceforth we assume that each image has a maximal side length of 500 pixels and the depicted object abuts on the image boundary on all sides.

We assume that the input shapes have a consistent scale and orientation, and that the underlying reflectional symmetry plane, if present, is about the x-plane. We also assume that each shape comprises one or more disconnected segments. Publicly available 3D models are often composed of disconnected parts: human modelers typically sculpt distinct components, which correspond to parts of the object, and then simply position them to form the model. These vestigial segmentations are noisy, with some models segmented poorly or not at all. We thus do not assume that the given segmentations can be used as found. We also do not assume that any correspondence structure between the segments is known in advance. Our pipeline uses the given noisy segmentations to initialize automatic joint segmentation of images and shapes.

To propagate information between shapes in $\mathcal{S}$ and images in $\mathcal{I}$, we synthesize a set of rendered images. To create these, we generate 360 camera poses by sampling the upper half of the viewing sphere using farthest point sampling [Gonzalez 1985] and direct-

ing the camera toward the center from each sample. Let $\mathcal{P}$ be the resulting set of 360 poses. We now generate a set $\mathcal{R}$ of rendered images by rendering each shape in $\mathcal{S}$ from each pose in $\mathcal{P}$. Thus $|\mathcal{R}| = |\mathcal{S}||\mathcal{P}|$.

Pipeline. The pipeline begins by estimating the camera pose for each image in $\mathcal{I}$. This is cast as a structured prediction problem. We construct a conditional random field that links natural images to each other and to rendered images, based on image appearance descriptors. Camera poses for natural images are optimized to be consistent with the ground-truth poses of similar rendered images, as well as with optimized poses in similar natural images. This stage is described in Section 4.

Next, we compute dense correspondences that link images in $\mathcal{I}$ to each other and to images in $\mathcal{R}$. To this end, we iterate over patches in all images in $\mathcal{I}$ and identify similar patches in other images based on appearance features. This is used to construct a weighted graph that connects similar patches. A robust clustering algorithm is used to extract a set of patch clusters with reliable correspondences within each cluster. Based on these correspondences, we align all patches within each cluster by optimizing a joint nonrigid alignment objective. This yields highly accurate pixel-level correspondences within patch clusters.

The patch clusters connect natural images and rendered images. Rendered images carry segmentation information from the original shapes: for each pair of pixels in a rendered image, we know whether the pair belongs to the same segment or to different segments on the original shape. These pairwise relationships are propagated to natural images via pixel-level correspondences within patch clusters. Graph clustering is then used to segment both the natural images and the rendered images based on the same evidence. These segmentations of the rendered images are then propagated back to the shapes in $\mathcal{S}$. We thus obtain compatible segmentations and correspondences that connect image segments and shape parts. This stage is described in Section 5.

The computed correspondences are used to associate each image segment with a shape part, and to carry over adjacency and symmetry information from shapes to images. An initial model for each image $I \in \mathcal{I}$ is assembled from shape parts associated with each segment in I. This model is then optimized to projectively align with image contours. The optimization objective is based on projective correspondences, the accuracy of which depends on the accuracy of the camera parameters. Since the initial camera pose estimation was approximate, we include intrinsic and extrinsic camera parameters in the optimization, refining them while optimizing the rigid transforms and nonrigid deformations for all segments. The object's configuration is regularized by symmetry and adjacency relationships inferred from the input shapes and images. The optimization is described in Section 6.

4 Camera Pose Estimation

A natural approach to estimating the pose of an image $I \in \mathcal{I}$ is to retrieve the most similar rendered images and use their known camera poses to estimate the pose of I [Aubry et al. 2014; Lim et al. 2014; Su et al. 2014]. We found that the results produced by this approach are noisy, as illustrated in Figure 3.

Our solution is to optimize a global objective that links similar images across $\mathcal{I}$. In this way, the image collection is used to regularize camera pose estimation for all images. To estimate a camera pose for each image in $\mathcal{I}$, we optimize a conditional random field (CRF). The label space is $\mathcal{P}$: the set of camera poses used for rendering the synthetic images in $\mathcal{R}$. The variable set is $\mathbf{P} = \{P_i\}$, where $P_i \in \mathcal{P}$ is the camera pose associated with image $I_i \in \mathcal{I}$. The CRF objective has the following form:

(a) Separate estimation (b) Similar images (c) Joint estimation

Figure 3: *Camera pose estimation. (a) Erroneous estimate, obtained when considering one natural image separately. (b) Our approach estimates camera poses for similar images jointly. (c) Joint analysis regularizes the estimation and significantly increases reliability.*

$$\underset{\mathbf{P}}{\text{minimize}} \quad \sum_{i=1}^{N_{\mathcal{I}}} \phi_i(P_i) + \sum_{(I_i,I_j)\in\mathcal{N}} \psi_{ij}(P_i, P_j). \qquad (1)$$

The unary potentials $\phi_i(P_i)$ are defined by the distribution of camera poses of images in $\mathcal{R}$ that are similar to I_i. The camera poses are quantized to $\mathcal{P}$. The distribution is aggregated over a set of nearest neighbors in HOG descriptor space [Dalal and Triggs 2005]. We use 4x4 HOG cells.

The connectivity structure $\mathcal{N}$ links each image to its 6 nearest neighbors in HOG space. The pairwise potentials penalize inconsistent estimates for similar images:

$$\psi_{ij}(P_i, P_j) = \mu(P_i, P_j)\alpha(I_i, I_j).$$

Here $\mu(P_i, P_j)$ is a label compatibility term, defined as $\mu(P_i, P_j) = \rho(\angle(P_i, P_j))$, where $\angle(P_i, P_j)$ is the angle between the optical axes of P_i and P_j and ρ is the truncated L^1 penalty. The weight $\alpha(I_i, I_j)$ adjusts the strength of the pairwise term based on the similarity of I_i and I_j and is defined in terms of distance in HOG space.

Objective (1) is optimized using TRW-S [Kolmogorov 2006]. As shown in Section 7, estimating camera poses jointly across the image collection increases accuracy considerably.

5 Segmentation and Correspondence

Given the estimated camera poses for all images, the second stage of the pipeline computes dense pixel-level correspondences between image patches. The correspondences connect the image sets $\mathcal{I}$ and $\mathcal{R}$, ultimately enabling inference about the three-dimensional structure of objects depicted in natural images.

5.1 Pixel-level correspondences

Patch graph construction. We operate on patches of size 96x96, regularly sampled with a 16-pixel stride in all images in $\mathcal{I} \cup \mathcal{R}$. These patches are connected into a graph $\mathcal{G} = (\mathcal{G}_V, \mathcal{G}_E)$. The edges $\mathcal{G}_E$ link patches with similar appearance and context. For each patch u in image I, we compute a multi-scale HOG descriptor centered at u. The descriptor summarizes the appearance of the patch and its context. To link u with other patches, we consider images with viewing direction within 15° of the estimated viewing direction of I. In each image, we identify the patch with the closest descriptor to u. Among these, we retain the k most similar. (We use $k = 12$ in all experiments.) A high-level descriptor like HOG is not in itself sufficiently discriminative for the level of precision we seek. We therefore compute dense correspondences between u

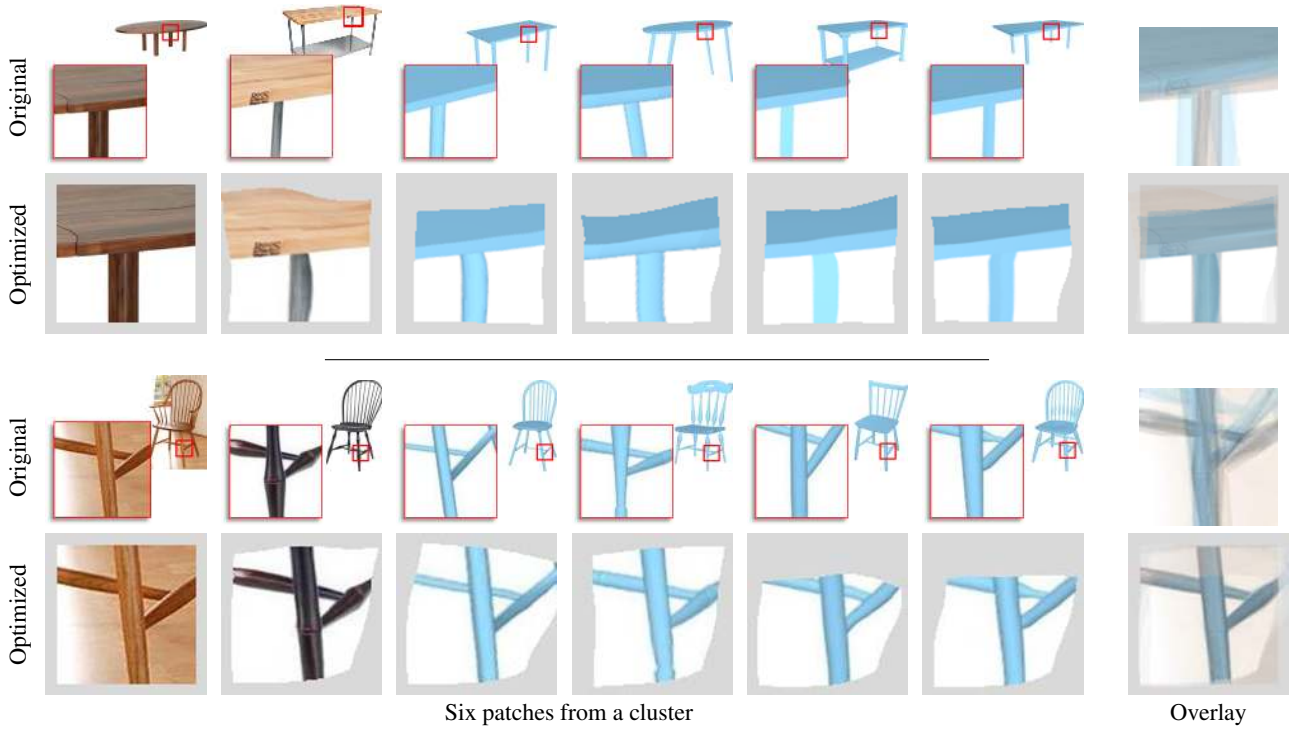


Figure 4: *Nonrigid alignment for dense pixel-level correspondence. Example from the tables dataset on top, example from the Windsor chairs dataset below. For each example, the top row shows six patches from a cluster ω and the bottom row shows the optimized nonrigid alignment of these patches to the common domain $\Sigma(\omega)$. Each row shows two patches from natural images, four patches from rendered images, and the overlay of all six patches. The overlays illustrate that the optimized patches are aligned better than the original patches.*

and each of the k retained candidate patches using SIFT flow [Liu et al. 2011]. The patch u is then connected to the closest $k' = k/2$ patches based on the SIFT flow matching energy, and the weight of the corresponding edge in $\mathcal{G}_E$ is set based on the same matching energy.

The graph $\mathcal{G}$ is used to cluster the patches. We use a variant of spectral clustering. The spectral embedding space is given by the top 10 eigenvectors of the normalized graph Laplacian $\mathcal{L}_\mathcal{G}$. We perform agglomerative clustering in this space. Clusters are merged until each has patches from rendered images of at least 15 models. This yields a set $\Omega = \{\omega\}$ of clusters, such that each $\omega = \{u\}$ collects a set of patches with similar appearance and camera pose. As we shall see, including rendered images of multiple models in each cluster increases robustness to noisy initial segmentations.

Dense nonrigid alignment. We now compute pixel-level correspondences between patches. This is done by jointly optimizing a mapping of all patches in each cluster ω to a common two-dimensional domain $\Sigma(\omega)$. This naturally induces a dense mapping between each pair of patches. Specifically, we optimize a 2D free-form deformation (FFD) $f_i : u_i \to \Sigma(\omega)$ for each patch $u_i \in \omega$ [Sederberg and Parry 1986]. The mapping has the form $f_i(\mathbf{x}) = \sum_l b_l(\mathbf{x}) c_{i,l}$, where b_l and $c_{i,l}$ describe the bilinear basis and the control points, respectively. We use an 8×8 control grid.

To set up the objective function for joint optimization of the deformations $\{f_i\}$, we utilize the point-wise correspondences $\mathcal{K}_{i,j}$ computed by SIFT flow between each pair of patches $u_i, u_j \in \omega$ during patch graph construction. To tolerate outliers in this initial set of correspondences, the objective uses the robust L^p norm, where $p = 0.8$:

$$\underset{\{f_i\}}{\text{minimize}} \quad \sum_{i,j} \sum_{\{\mathbf{p},\mathbf{q}\} \in \mathcal{K}_{i,j}} \| f_i(\mathbf{p}) - f_j(\mathbf{q}) \|^p . \qquad (2)$$

This objective is optimized using the Gauss-Newton method, initialized by the identity maps. To prevent a collapse to the trivial solution in which the control grids shrink to a point, we fix the mapping f_i for one of the images to the identity mapping. The effect of this optimization is illustrated in Figure 4.

5.2 Segmentation

This step jointly segments images in $\mathcal{I}$ and models in $\mathcal{S}$, using the initial segmentations of the models for bootstrapping. Since the initial segmentations are noisy, we use the pixel-level mappings within each patch cluster to aggregate segmentation information from multiple models. In addition, we also aggregate adjacency information that is later used to regularize the reconstruction.

Given patch cluster ω, define a cumulative similarity score $\delta(\mathbf{p}, \mathbf{q}) = \delta_+(\mathbf{p}, \mathbf{q}) - \delta_-(\mathbf{p}, \mathbf{q})$ between each pair of pixels $\mathbf{p}, \mathbf{q} \in \Sigma(\omega)$. Here $\delta_+(u, v)$ ($\delta_-(u, v)$) is the frequency with which $\mathbf{p}$ and $\mathbf{q}$ appear in the same segment (different segments) in rendered images in ω. Despite the noise in the segmentations of the input shapes, we found that the sign of $\delta(\mathbf{p}, \mathbf{q})$ is a good indicator of whether the two pixels belong to the same segment. For each image $I \in \mathcal{I} \cup \mathcal{R}$, we collect pairwise scores from all patches in the image. This yields a weighted graph over the pixels of I, which is segmented using graph clustering, yielding a segmentation of I.

This process segments both the natural images $\mathcal{I}$ and the rendered images $\mathcal{R}$. Segmentations of the rendered images are then used to segment the shapes $\mathcal{S}$ using a variant of the approach described by

Wang et al. [2013], yielding shape parts that are compatible with image segments.

Adjacency information is propagated in a similar fashion. For pairs of pixels in each patch cluster that belong to different segments, we mark pairs of pixels that belong to adjacent segments, following the sign criterion described above. After segmenting each image, two image segments are marked as adjacent if at least half of the pixel pairs that connect these two segments within patch clusters are marked as adjacent.

5.3 Associating image segments and shape parts

We now associate each image segment with a shape part. To this end, for each pair u, u' of image segments in $\mathcal{I} \cup \mathcal{R}$ that overlap in some patch clusters, we compute a weight $\beta(u, u')$ that quantifies their similarity. The weight has the form $\beta(u, u') = \beta_o(u, u')\beta_s(u, u')$. Here $\beta_o(u, u')$ is defined as the percentage of pixels of u and u' that overlap with each other in one or more patch clusters, and $\beta_s(u, u')$ is the shape context score of u and u' [Belongie et al. 2002]. To link an image segment u with a shape part, we identify the highest-scoring rendered segment. At this stage we also detect symmetry between shape parts by solving a maximum matching problem on a graph that connects all pairs of parts within each shape. For a pair of parts, the corresponding edge weight in the graph is set by the distance between one part and a reflected copy of the other.

6 Reconstruction

The final stage of the pipeline creates a 3D model M for each image $I \in \mathcal{I}$. The model is composed of a set of parts $\{v^j\}$. This set is initialized by retrieving the shape part associated with each image segment $u^j \subset I$. Each part carries over its initial pose from its source shape.

The initial models are illustrated in Figure 5. Initial shapes and poses of the collected parts only approximate the object shown in I. We now optimize these shapes and poses to fit contours of the corresponding image segments in I. Since this optimization aims to fit the projections of the shape parts to the image segments, the quantization of the camera pose P, described in Section 4, becomes an impediment to accuracy. We therefore include the extrinsic and intrinsic camera parameters in the optimization and optimize them in tandem with the shapes and poses. Since 3D reconstruction from a single view is ill-posed, we use symmetry and adjacency information to regularize the optimization.

(a) Input (b) Before optimization (c) After optimization

Figure 5: *Effect of optimization. From left to right: input images, initial models (two views), final models after optimization (same views).*

6.1 Objective

The configuration of a part v^j is represented by a rigid pose T^j and a nonrigid deformation C^j. The pose T^j represents a rigid displacement from the initial pose and is initialized to the identity transform. The deformation C^j specifies the configuration of an FFD control grid, which is parameterized relative to its default configuration and is initialized to 0. The optimization jointly refines the camera transform P, the set $\mathbf{T} = \{T^j\}$ of rigid poses, and the set $\mathbf{C} = \{C^j\}$ of control grid configurations. The parameterization of rigid poses and control grid configurations is reduced such that variables are shared between symmetric parts: the control grids for a symmetric pair are identical and the poses are related by a reflection about the symmetry plane.

The objective comprises an image alignment term, a simple regularizer on the rigid pose and nonrigid deformation of each part, and a correspondence term that binds adjoining parts:

$$E(P, \mathbf{T}, \mathbf{C}) = E_{\text{image}}(P, \mathbf{T}, \mathbf{C}) + \lambda E_{\text{reg}}(\mathbf{C}) + \mu E_{\text{corr}}(\mathbf{T}, \mathbf{C}). \quad (3)$$

The image alignment term guides the contours of shape parts, as seen from the camera, to align with the boundaries of image segments. Since shape part contours depend both on the part configurations and the camera parameters, this term involves all sets of variables. Let Λ^j be the set of contour pixels associated with u^j. The image alignment objective is

$$E_{\text{image}}(P, \mathbf{T}, \mathbf{C}) = \sum_j \sum_{\mathbf{p} \in \Lambda^j} \min_{\mathbf{q} \in \partial P(T^j(C^j(v^j)))} \|\mathbf{q} - \mathbf{p}\|^2, \quad (4)$$

where $P(\cdot)$ denotes the application of the camera transform, $C^j(\cdot)$ denotes the application of the nonrigid deformation, and ∂ is the boundary operator.

The regularization term penalizes large rigid transforms and nonrigid deformations:

$$E_{\text{reg}}(\mathbf{C}) = \sum_j \left(\|T^j - I\|_F^2 + \sum_{c \in C^j} c^2 \right), \quad (5)$$

where I is the identity matrix and $\|\cdot\|_F$ is the Frobenius norm. The correspondence term $E_{\text{corr}}(\mathbf{T}, \mathbf{C})$ binds adjoining parts:

$$\sum_{\{i,j\} \in \mathcal{K}} \frac{1}{|\mathcal{J}_{ij}|} \sum_{(\mathbf{x}, \mathbf{y}) \in \mathcal{J}_{ij}} \left\| T^i(C^i(\mathbf{x})) - T^j(C^j(\mathbf{y})) \right\|^2, \quad (6)$$

where $\mathcal{K}$ is the set of adjoining part pairs identified in Section 5.3 and $\mathcal{J}_{ij}$ collects pairs of points that span the minimal translational distance between v^i and v^j [Cameron and Culley 1986].

6.2 Optimization

The presented objective is difficult to optimize. The first difficulty is the coupling of the camera configuration P and the apparent contours $\{\partial P(T^j(C^j(v^j)))\}$ in the image alignment term (4): as the camera configuration changes, the contours change. The second difficulty is that the pairs $\{\mathcal{J}_{ij}\}$ that bind adjoining parts in the correspondence term (6) likewise change during the optimization.

We address these difficulties using alternating optimization. Each of the three parameter blocks—the camera configuration, the rigid poses, and the nonrigid deformations—is optimized in turn, and the apparent contours $\{\partial P(T^j(C^j(v^j)))\}$ and correspondence pairs $\{\mathcal{J}_{ij}\}$ are periodically updated.

When the camera configuration P is optimized, only the image alignment term is active. We compute the apparent contours

$\{\partial P(T^j(C^j(v^j)))\}$ and extract projective correspondences between $\partial P(T^j(C^j(v^j)))$ and Λ^j for each j. The apparent contours and projective correspondences are then held fixed. The camera matrix is optimized to fit the computed correspondences; both extrinsic and intrinsic parameters are optimized [Hartley and Zisserman 2000]. To optimize the part poses $\mathbf{T}$, we alternate between updating the correspondence pairs $\{\mathcal{J}_{ij}\}$ and performing Gauss-Newton steps. The adjoining correspondences are held fixed for the other stages. Finally, we update the nonrigid deformations $\mathbf{C}$: this reduces to a convex program.

This procedure is performed for a number of iterations, which is set to 150 in our implementation. We update the correspondence pairs every 10 iterations. The effect of the optimization is shown in Figure 5.

7 Results

We have collected images and shapes for five sets of objects, summarized in Table 1. Images were downloaded by querying the Bing Search API with the corresponding keywords. The results were pruned using Amazon Mechanical Turk (AMT) to retain relevant images in which the target object is seen clearly and distinctly. AMT was also used to obtain bounding boxes for all objects. Shapes were obtained from the 3D Warehouse and Yobi3D.

	Input			Running time (hours)				
	$N_\mathcal{I}$	$N_\mathcal{S}$	# seg.	pose	seg.	rec.	total	# src.
Windsor chairs	981	103	17.1	2.4	12.1	10.2	24.7	4.8
Office chairs	687	79	12	1.9	14.1	9.1	25	1.8
Tables	584	137	6.2	1.2	6.5	7.4	15.1	2.4
Bicycles	897	131	12.1	1.1	7.9	10.1	20.1	3.1
Guns	542	167	7.2	1.1	5.4	14.1	20.6	2.6

Table 1: *Statistics for each dataset. On the left, size of image collection, size of shape collection, and average number of segments in the original input shapes. In the middle, running time for each stage in the pipeline (camera pose estimation, segmentation and correspondence, reconstruction) and total running time, all for processing the complete datasets. In the rightmost column, the average number of source shapes from which parts were taken to compose each reconstructed model.*

The five image collections were reconstructed using the presented pipeline. Running times for processing the complete datasets are summarized in Table 1. Running times were measured on a workstation with two Intel Xeon E5-2660 processors. Our implementation is in Matlab and was not optimized. Examples of reconstructions are shown in Figures 8 and 9. For each input image, the figures show the computed image segmentation and the reconstructed 3D model. For reference, the figures also show the most similar pre-existing model, retrieved using multi-scale HOG. Reconstructions synthesized by the presented approach reproduce the objects depicted in the images more accurately than pre-existing models. This is validated quantitatively in Section 7.1. Additional results are provided in supplementary material.

7.1 Evaluation

We have conducted a quantitative evaluation of each component of the presented pipeline. The evaluation was performed on three datasets: Windsor chairs, bicycles, and guns. Three human assistants were employed to create detailed ground-truth data.

Pose estimation. We evaluated the accuracy of the pose estimation approach presented in Section 4 on 200 randomly sampled images from each dataset. To create a ground-truth camera pose for each image, three annotators manually selected a similar shape from the shape collection and manually manipulated the camera around the chosen shape to match the view in the image.

We compared the camera poses produced by our approach, which is based on joint estimation over an image collection, to camera poses produced by the approach of Su et al. [2014], which estimates a camera pose for each image separately. Accuracy was measured by angular deviations from ground truth. Figure 6 shows cumulative distributions of angular deviations. The average angular deviation of poses computed by our approach was 8.3°, the average angular deviation of poses produced by the reference approach was 16.4°. The variance among the human annotators was 7.2°.

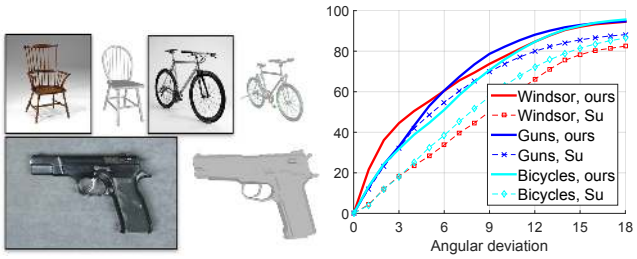


Figure 6: *Evaluation of camera pose estimation. Left: ground-truth poses created by human annotators using reference database shapes. Right: cumulative distributions of angular deviations from ground truth on three datasets. The presented approach significantly outperforms the reference approach.*

Pixel-level correspondences. We now evaluate the accuracy of pixel-level correspondences estimated by the presented pipeline as described in Section 5. To obtain ground-truth correspondences, we randomly sampled 20 pairs of images per dataset. Each pair consists of a natural image and a corresponding rendered image. Human annotators marked sets of corresponding pixels in each pair. Figure 7 shows some of these manually estimated correspondences. Accuracy of pixel-level correspondences is measured by pixel distance from ground truth. We compare the accuracy of the presented approach to the accuracy of correspondences estimated by the approach of Su et al. [2014]. Cumulative error distributions are shown in Figure 7. The average accuracy of our approach is 3.8 pixels, the average accuracy of the reference approach is 6.4 pixels.

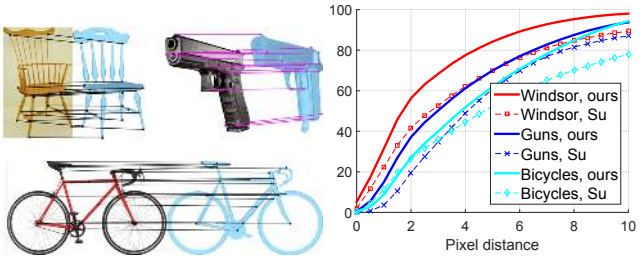


Figure 7: *Evaluation of pixel-level correspondences. Left: ground-truth correspondences marked by human annotators. Right: cumulative distributions of distances from ground truth. The presented approach significantly outperforms the reference approach.*

Segmentation. To evaluate the accuracy of image segmentation performed by the presented approach, we selected 20 images per dataset. Each of these images was manually segmented by human annotators. For reference, the annotators were given images of 10 shapes from each dataset, rendered from multiple views. The shapes were chosen based on the quality of their pre-existing segmentation and rendered such that this segmentation is apparent. Examples of ground-truth segmentations produced by the annotators are shown in Figure 10.

Figure 8: *Results on four datasets. From left to right in each column: Web image, computed segmentation, 3D model reconstructed by our approach (two views, green), and closest pre-existing model, shown for reference (blue).*

Figure 9: *Results on the office chairs dataset, arranged as in Figure 8.*

Image segmentations produced by our complete approach were compared to segmentations produced by the same approach when the input consists of only a single natural image along with the complete shape collection. As a baseline, we also segmented each image using normalized cuts, for which parameters were set to produce a similar number of segments [Shi and Malik 2000]. Accuracy was evaluated using the Rand index [Unnikrishnan et al. 2007]. The results are shown in Figure 10. Segmenting the image collection jointly improves accuracy significantly.

	Chairs	Bikes	Guns
Joint	**5.9**	**8.9**	**12.1**
Individual	13.5	16.1	16.7
Norm. cuts	35.1	37.1	39.1
Human	3.6	4.9	5.6

Figure 10: *Evaluation of image segmentation. Left: images and ground-truth segmentations. Right: Rand index scores for the presented joint analysis approach, an ablated version of the presented approach that only considers a single natural image at a time (along with the complete set of rendered images), normalized cuts, and human consistency. The presented approach significantly outperforms the alternatives.*

Reconstruction accuracy. Finally, we evaluate the accuracy of 3D models produced by the presented approach. For this experiment, we randomly sampled 10% of the pre-existing models in each dataset, removed them from the input model collection, and rendered each model from a random camera pose. Lighting and material parameters were set to approximate conditions observed in natural images. The synthesized images were added to the image collection and the presented approach was used to create 3D reconstructions.

Reconstruction accuracy was measured using the average Euclidean distance of the synthesized model to the ground-truth shape. Distance was computed by distributing 1000 samples uniformly on each surface and evaluating nearest-neighbor distances from the reconstruction to the ground-truth surface. In addition, to evaluate the detailed quality of local surface geometry, we measured angular deviations of the normals of nearest-neighbor pairs.

As a baseline, we measured the accuracy of the most accurate pre-existing model: that is, the model that minimizes the average Euclidean distance to the ground-truth surface. This baseline represents the performance of an oracle that retrieves the best pre-existing model. In addition, we measured the accuracy of depth maps synthesized by the approach of Su et al. [2014]; the output of this approach is a point cloud that models only one aspect of the object, but its accuracy can still be estimated by computing the average Euclidean distance to the ground truth shape.

The results are reported in Figure 11. The average accuracy of reconstructions produced by our approach is $0.031d$, where d is the diameter of the ground-truth model. In comparison, the average accuracy of the best pre-existing model is $0.082d$. The average accuracy of the depth maps produced by the approach of Su et al. is $0.061d$, considerably worse than the accuracy of our reconstructions. Note that the asymmetric distance measure we use does not penalize the approach of Su et al. for covering only the visible front-facing parts of the object, but does evaluate the accuracy of back-facing and occluded surfaces in reconstructions produced by our approach. Despite this stringent protocol, our accuracy is better by a factor of 2. A qualitative comparison is shown in Figure 12.

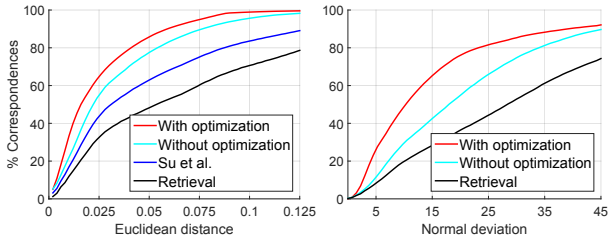


Figure 11: *Evaluation of reconstruction accuracy. Left: cumulative distributions of Euclidean distances from reconstruction to ground-truth shape. Right: cumulative distributions of angular distances of reconstructed normals and corresponding ground-truth normals. Our approach is much more accurate.*

The asymmetric distance measure, which was used to maximize the reported accuracy for the approach of Su et al., is permissive of incomplete reconstructions and does not fully characterize the performance of our pipeline. When symmetric Euclidean distance is

evaluated, averaging over distances from reconstruction to ground-truth as well as from the ground-truth to the reconstruction, the accuracy of our approach remains $0.033d$, while the accuracy of the best pre-existing model drops to $0.145d$.

We have also evaluated the effect of the final optimization stage, described in Section 6, on reconstruction accuracy. To this end, we measured the accuracy of the initial assembled models before the optimization. The results are also reported in Figure 11. The average accuracy at initialization is $0.056d$. While these initial models are already more accurate than the baselines, the optimization stage is important. The impact is particularly significant for local details, as indicated by the angular error distributions.

(a) Input (b) Same viewpoint (c) Different viewpoint

Figure 12: *Qualitative comparison to depth maps produced by the approach of Su et al. [2014]. (a) Input images. (b) Reconstructions synthesized by our approach and depth maps synthesized by the approach of Su et al., shown from the original viewpoint. (c) Same reconstructions and depth maps shown from a different viewpoint.*

7.2 Applications

The motivating application of this work is the creation of very large databases of 3D models, which can be used for computer graphics applications and for training computer vision systems. In this section we present two additional applications that do not entail large-scale reconstruction. The first is demonstrated in Figure 13. Given an image of an object, the presented approach can be used to reconstruct a 3D model of the depicted object. This reconstruction can be textured by projecting the image colors onto the model. To assign colors to occluded surfaces, we use symmetry and simple closest-point color propagation. More sophisticated color mapping approaches can be used [Kholgade et al. 2014]. The textured model can now be rendered from different viewpoints, thus enabling 3D manipulation of objects depicted in photographs.

Another application utilizes three-dimensional understanding of depicted objects to enable advanced image search. Using novel views of an object depicted in an image, synthesized as described above, we can search for images of similar objects seen from different perspectives. This is illustrated in Figure 14. The figure shows images retrieved by distance in HOG space. When synthesized images from novel views are used, the retrieved objects are similar to the object depicted in the original image, despite dissimilarity of image features. Note that both applications use textured models and can be hindered by imprecise alignment of the reconstruction to the original image.

Input Novel views

Figure 13: *Given a single image of an object, our approach can be used to manipulate the depicted object in 3D.*

Original

Synthesized Retrieved images

Figure 14: *Top: search with the original image. Bottom: search with a synthesized image of the same object from a different view. Synthesized images of different aspects can be used to identify additional relevant images.*

8 Discussion

We presented a single-view reconstruction approach that can create 3D models of objects depicted in Web images. The approach has a number of limitations that suggest fruitful directions for future work. First, we rely on the availability of initial segmentations of the input shapes. While our approach is robust to noise in these segmentations, it would be handicapped if the provided shape collection is missing segmentation information altogether: the pipeline would essentially reduce to retrieving a complete shape and fitting it to image contours. While many existing 3D models have useful segmentations, we found that some types of objects, such as bottles and shoes, generally come in one piece. Furthermore, models produced by range scanning usually emerge as fused shapes. Techniques for compatible shape segmentation could be applied in this case [Huang et al. 2011]. Since our approach can utilize a comparatively small shape collection to reconstruct a large image collection, some manual assistance in segmenting the initial shapes could also be employed. This investment would then be leveraged to yield a much larger set of models.

The presented approach will fail to accurately reconstruct the images if appropriate parts are not present in the shape collection or if they are not correctly identified by the computed correspondences. This is illustrated in Figure 15. Furthermore, our current nonrigid deformation formulation cannot match detailed geometry, such as

Figure 15: *Failure cases. From left to right: erroneous pose estimation, incomplete segmentation, inability to retrieve sufficiently similar parts, poor composition.*

ornamentation, if it was not originally present.

Another challenge is scalability. Our current implementation is not optimized and cannot handle Web-scale datasets. We believe that such massive scalability is possible and hope that the presented ideas will inform future work that will yield millions of high-quality 3D models. Our implementation will be made freely available.

References

AUBRY, M., MATURANA, D., EFROS, A., RUSSELL, B., AND SIVIC, J. 2014. Seeing 3D chairs: Exemplar part-based 2D-3D alignment using a large dataset of CAD models. In *CVPR*.

AVERBUCH-ELOR, H., WANG, Y., QIAN, Y., GONG, M., KOPF, J., ZHANG, H., AND COHEN-OR, D. 2015. Distilled collections from textual image queries. *Comput. Graph. Forum 34*.

BELONGIE, S., MALIK, J., AND PUZICHA, J. 2002. Shape matching and object recognition using shape contexts. *PAMI 24*, 4.

CAMERON, S., AND CULLEY, R. 1986. Determining the minimum translational distance between two convex polyhedra. In *ICRA*.

CARREIRA, J., KAR, A., TULSIANI, S., AND MALIK, J. 2015. Virtual view networks for object reconstruction. In *CVPR*.

CARREIRA, J., VICENTE, S., AGAPITO, L., AND BATISTA, J. 2015. Lifting object detection datasets into 3D. *PAMI*. To appear.

CHEN, T., ZHU, Z., SHAMIR, A., HU, S., AND COHEN-OR, D. 2013. 3-Sweep: extracting editable objects from a single photo. *ACM Trans. Graph. 32*, 6.

CRIMINISI, A., REID, I., AND ZISSERMAN, A. 2000. Single view metrology. *IJCV 40*, 2.

DALAL, N., AND TRIGGS, B. 2005. Histograms of oriented gradients for human detection. In *CVPR*.

DEBEVEC, P., TAYLOR, C., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: A hybrid geometry- and image-based approach. In *SIGGRAPH*.

EIGEN, D., PUHRSCH, C., AND FERGUS, R. 2014. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*.

FOUHEY, D., GUPTA, A., AND HEBERT, M. 2013. Data-driven 3D primitives for single image understanding. In *ICCV*.

GONZALEZ, T. F. 1985. Clustering to minimize the maximum intercluster distance. *Theoretical Computer Science 38*.

HARTLEY, R., AND ZISSERMAN, A. 2000. *Multiple View Geometry in Computer Vision*. Cambridge University Press.

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

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

KAR, A., TULSIANI, S., CARREIRA, J., AND MALIK, J. 2015. Category-specific object reconstruction from a single image. In *CVPR*.

KHOLGADE, N., SIMON, T., EFROS, A., AND SHEIKH, Y. 2014. 3D object manipulation in a single photograph using stock 3D models. *ACM Trans. Graph. 33*, 4.

KIM, V., LI, W., MITRA, N., DIVERDI, S., AND FUNKHOUSER, T. 2012. Exploring collections of 3D models using fuzzy correspondences. *ACM Trans. Graph. 31*, 4.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *PAMI 28*, 10.

LIM, J., KHOSLA, A., AND TORRALBA, A. 2014. FPM: fine pose parts-based model with 3D CAD models. In *ECCV*.

LIU, C., YUEN, J., AND TORRALBA, A. 2011. SIFT flow: Dense correspondence across scenes and its applications. *PAMI 33*, 5.

SEDERBERG, T. W., AND PARRY, S. R. 1986. Free-form deformation of solid geometric models. In *SIGGRAPH*.

SHEN, C., FU, H., CHEN, K., AND HU, S. 2012. Structure recovery by part assembly. *ACM Trans. Graph. 31*, 6.

SHI, J., AND MALIK, J. 2000. Normalized cuts and image segmentation. *PAMI 22*, 8.

SNAVELY, N., SIMON, I., GOESELE, M., SZELISKI, R., AND SEITZ, S. 2010. Scene reconstruction and visualization from community photo collections. *Proceedings of the IEEE 98*, 8.

SU, H., HUANG, Q., MITRA, N., LI, Y., AND GUIBAS, L. 2014. Estimating image depth using shape collections. *ACM Trans. Graph. 33*, 4.

UNNIKRISHNAN, R., PANTOFARU, C., AND HEBERT, M. 2007. Toward objective evaluation of image segmentation algorithms. *PAMI 29*, 6.

WANG, Y., GONG, M., WANG, T., COHEN-OR, D., ZHANG, H., AND CHEN, B. 2013. Projective analysis for 3D shape segmentation. *ACM Trans. Graph. 32*, 6.

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.

ZHENG, Y., CHEN, X., CHENG, M., ZHOU, K., HU, S., AND MITRA, N. 2012. Interactive images: cuboid proxies for smart image manipulation. *ACM Trans. Graph. 31*, 4.