Chen X, Xu WW, Yeung SK *et al.* View-aware image object compositing and synthesis from multiple sources. JOURNAL OF COMPUTER SCIENCE AND TECHNOLOGY 31(3): 463–478 May 2016. DOI 10.1007/s11390-016-1640-8

View-Aware Image Object Compositing and Synthesis from Multiple Sources

Xiang Chen [1], *Member, ACM*, Wei-Wei Xu [1], *Member, IEEE*, Sai-Kit Yeung [2], *Member, IEEE*, and Kun Zhou [1,*], *Fellow, IEEE*

[1] *State Key Laboratory of Computer Aided Design and Computer Graphics, Zhejiang University*
Hangzhou 310058, China

[2] *Vision, Graphics and Computational Design Group, Singapore University of Technology and Design*
Singapore 487372, Singapore

E-mail: {xchen.cs, weiwei.xu.g}@gmail.com; saikit@sutd.edu.sg; kunzhou@acm.org

Received November 28, 2015; revised March 7, 2016.

Abstract Image compositing is widely used to combine visual elements from separate source images into a single image. Although recent image compositing techniques are capable of achieving smooth blending of the visual elements from different sources, most of them implicitly assume the source images are taken in the same viewpoint. In this paper, we present an approach to compositing novel image objects from multiple source images which have different viewpoints. Our key idea is to construct 3D proxies for meaningful components of the source image objects, and use these 3D component proxies to warp and seamlessly merge components together in the same viewpoint. To realize this idea, we introduce a coordinate-frame based single-view camera calibration algorithm to handle general types of image objects, a structure-aware cuboid optimization algorithm to get the cuboid proxies for image object components with correct structure relationship, and finally a 3D-proxy transformation guided image warping algorithm to stitch object components. We further describe a novel application based on this compositing approach to automatically synthesize a large number of image objects from a set of exemplars. Experimental results show that our compositing approach can be applied to a variety of image objects, such as chairs, cups, lamps, and robots, and the synthesis application can create novel image objects with significant shape and style variations from a small set of exemplars.

Keywords image cloning, 3D proxy, probabilistic modeling, data-driven method

1 Introduction

Image compositing is a useful operation and an important topic in computer graphics. The fundamental goal is to create a new image by putting together imaging contents from different sources. In most of the situations, the processing units are the image objects and the research focus is on how to blend them seamlessly onto a new background image. A well-known example technique is the gradient domain compositing introduced in Poisson Image Editing[1] which attracted significant research attention in the last decade[2–5] and is now the standard for seamless image compositing.

Another popular research direction is image collage in which the focus is on selecting and arranging multiple visual elements from separate source images, and combining them into a single image[6–11]. To achieve visually pleasing results, all the studies above assume the chosen objects have the same or very similar viewpoints to the compositing background.

In this paper, we would like to take one step further into a lower level operation to perform components-based compositing. We introduce view-aware image object compositing, a new approach to compositing novel image objects from multiple-source images which have different viewpoints. The main technical challenge

Regular Paper

Special Section of CVM 2016

This work is partially supported by the National Natural Science Foundation of China under Grant Nos. 61272305, 61303136, 61272392, and 61322204, and the National Program for Special Support of Eminent Professionals of China.

[*] Corresponding Author

is that the collected images lack 3D information required in both the component structure analysis and the stitching of components in different viewpoints into a new image object in a novel viewpoint. The problem becomes particularly severe when the images are from different sources (e.g., downloaded from Internet) and taken in various viewpoints. To this end, we propose a new coordinate-system based single-view camera calibration algorithm that is more suitable for general image objects that have limited geometric clues. With the estimated camera parameters, we then compute the cuboid proxies for image object components with the correct structure relationship via a structure-aware cuboid optimization algorithm. Finally, we stitch the components by a 3D-proxy transformation guided image warping algorithm to obtain the final object composite.

Based on our view-aware image object compositing, we further propose a novel image object synthesis application that can automatically synthesize a large number of image objects from a small given set of exemplar images. Like in 3D shape synthesis[12-13], we generate new image objects by combining components of a set of input images, and emphasize the importance of the correct relationship between components in the synthesis. We propose an analysis-and-synthesis approach to solve the problem. In the analysis stage, we construct a Bayesian graphical model to encode shape styles, camera parameters, structural relationships and complex dependencies among components. In the synthesis stage, we sample the graphical model to obtain the viewpoint and component set information, and merge the components seamlessly to produce new image objects.

In summary, we present a component-level view-ware approach to image object compositing, and make the following technical contributions:

- A structure-aware cuboid optimization algorithm to generate the cuboid proxies for image object components with correct structure relationship;
- A 3D-proxy transformation guided image warping algorithm to stitch object components from different images into a new image object. The correspondence information required in the 2D affine transformation is automatically derived from the 3D proxy transformation;
- A novel image object synthesis application to automatically synthesize a large number of image objects from a set of exemplars;
- A Bayesian graphical model which integrates the structural relationships and viewpoint information. This provides a convenient way to control the viewpoint variations in the synthesis.

We have created a consistently labeled image dataset with a variety of image objects, such as chairs, cups, lamps, toy planes and robots, and tested our composting and synthesis techniques on the dataset. Experimental results show that with a reasonable amount of interactive analysis work of image objects, our approach can composite and synthesize a large amount of man-made image objects with a variety of shapes and appearances (as illustrated in Fig.1).

2 Related Work

Image Compositing and Synthesis. The main goal of both image compositing and synthesis is to create a visually pleasing image from multiple-source im-

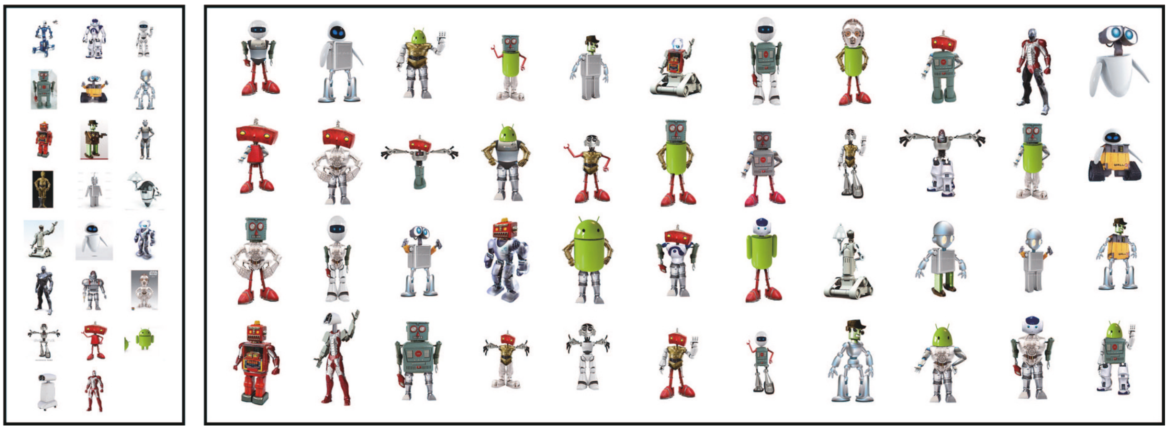


Fig.1. (b) Creative image objects varying in viewpoints, structures and appearances synthesized from (a) a set of exemplars.

ages. For image compositing, the focus is usually on new blending approaches to creating a seamless composition of selected contents. Earlier studies include multi-resolution spline technique[14-15] and compositing operators[16]. Since the introduction of Poisson Image Editing[1], gradient domain compositing[2-5] has become the standard for seamless image compositing in the last decade. More recently, Xue *et al.*[17] improved the visual realism of the composites by adjusting the appearance of the compositing objects.

Image synthesis[18-20] on the other hand usually focuses more on the selection and arrangement of contents. A representative line of work is the image collage in which multiple images are composited together under certain constraints to form a single image. It is pioneered by the interactive digital photomontage[6] and then various methods have been proposed, such as digital tapestry[7], autocollage[8], picture collage[9], Sketch2Photo[10], Photosketcher[11], cross-depiction[21], Arcimboldo-like[22] collage and the most recent circle packing collage[23].

Most image compositing and synthesis algorithms mentioned above implicitly assume the compositing contents have the same viewpoints with the source image, e.g., structured image hybrids[24], and do not handle the camera parameters. In photo clip art[24], the authors attempted to infer the camera pose from object heights. Their method, however, does not handle actual 3D relationships which prohibit operations involving perspective changes such as out-of-plane rotation (see Fig.2 for direct compositing without the consideration of viewpoints). Carroll *et al.*[25] presented a set of user interfaces to add perspective constraints like vanishing points and optimized them for image manipulation. This technology can interactively warp image contents and be applied for scene-level image compositing. Zheng *et al.*[26] explicitly optimized the camera and geometry parameters by representing image objects as 3D cuboid proxies. Our work also employs the 3D proxies representation, but in a more challenging setup that involves spatial structures of multiple components within non-cuboid objects. Recently, Chen *et al.*[27] utilized generalized cylinder to represent components in an image and enable interactive editing on the image. In contrast, our work focuses on compositing and synthesis of image object components from multiple sources in the same category. Miao *et al.*[28] presented a system for generating 3D symmetric freeform shapes from 2D sketches.

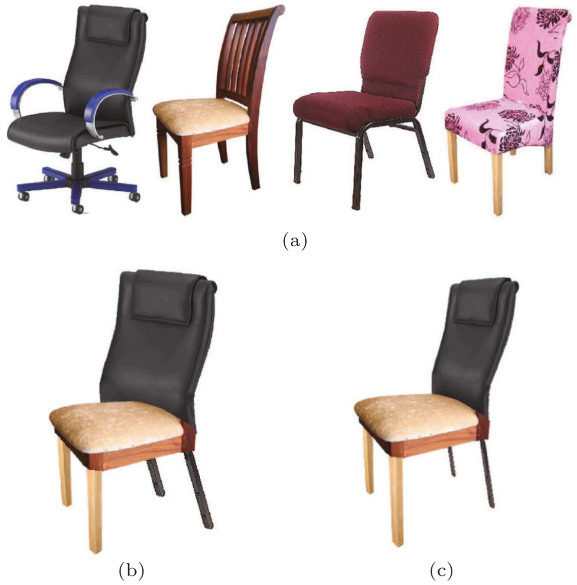


Fig.2. Direct compositing without the consideration of viewpoints could lead to unrealistic results. (a) Source images. (b) Direct compositing. (c) View-aware compositing.

Data-Driven 3D Model Synthesis. One particular application of our compositing technique is image object synthesis from a set of exemplar images, which can also be considered as the 2D counterpart of data-driven 3D modeling pioneered by Funkhouser *et al.*[29] Their modeling by example system allows users to search a database of segmented 3D parts and assemble new shapes interactively with the retrieved components. Follow-up studies take user sketch for components retrieval[30-32] or let users interchange parts from a small set of compatible shapes[33]. More recently, Chaudhuri and KoHun[34] proposed a data-driven approach to suggest suitable components for incomplete shapes and develop a probabilistic representation of shape structure that provides more semantically and stylistically compatible suggestions[35]. Their probabilistic reasoning approach is further extended for synthesizing complete shapes[12]. Our image object synthesis application adopts a similar probabilistic model, but on the relationships of object components in image space.

3 View-Aware Image Object Compositing

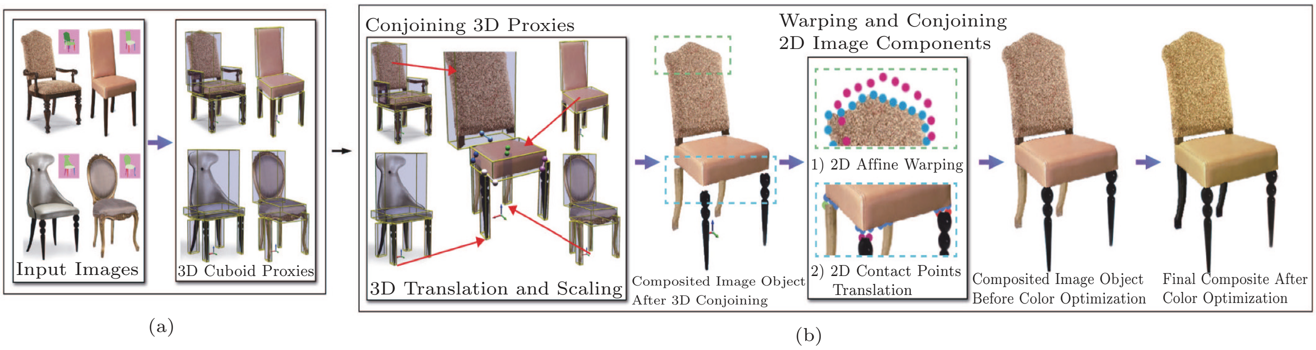
Our image compositing approach takes a repository of images of a particular object as the input, analyzes their structures and extracts the corresponding camera parameters semi-automatically. We fit a set of 3D cuboid proxies according to the underlying structure and build a graph to represent the image object. The image components are stitched together into a complete object with correct perspective under the guidance of 3D proxies.

As shown in Fig.3, our approach consists of two main stages.

3D Cuboid Proxy Construction. We first estimate camera parameters of images by single-view calibration methods and then construct 3D cuboid proxies of each segmented component based on non-local relationships (e.g., reflectional symmetries) to handle perspective effects. Then we propose a graphical representation to store the estimated 3D cuboid proxies and contact points between components. We also store the estimated camera parameters in the graph for each image object.

Proxy-Guided Component Compositing. After constructing the structure-aware 3D proxies for all the components, we can construct new image objects by stitching the components using 3D proxies and contact points information extracted in the first stage. We also optimize colors harmony for the components by leveraging color theme compatibility scores and the color palettes computed from the input images or training dataset.

Before going into the algorithmic details, we would like to describe the graph representation of the image objects. The representation is useful for final compositing and the later image object synthesis application (Section 4).


Input Images
3D Cuboid Proxies
Conjoining 3D Proxies
3D Translation and Scaling
Composited Image Object After 3D Conjoining
Warping and Conjoining 2D Image Components
1) 2D Affine Warping
2) 2D Contact Points Translation
Composited Image Object Before Color Optimization
Final Composite After Color Optimization

(a)

(b)

Fig.3. Overview of our view-aware image object compositing approach. (a) 3D cuboid proxy construction. (b) Proxy-guided component compositing.

3.1 Image Object Representation

Given an image taken by a camera or downloaded from Internet, we segment the object using Lazy Snapping[36] and label each component of the object using LabelMe Toolbox[37]. With the labeled segments, we can then model the structure information of the image object as the connection relationship between the semantic components of the object. It is done by storing the image pixels, silhouette and connections of each component in a graph. For example, a typical chair has components like seat, back, arms and legs. It can be represented by a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where each component C_i is a node in $\mathcal{V}$. Whenever two nodes C_i and C_j are connected in the image, edge e_{ij} is stored in $\mathcal{E}$. Each node C_i is in fact a tuple $C_i = \{X_i, S_i, mc, P_i\}$, where X_i indicates the set of pixels that belongs to the component and S_i is its silhouette in the image. mc indicates the major color extracted from the component pixels by k-means, where $k = 2$. P_i denotes its 3D cuboid proxy which will be described shortly. For those components under occlusions, we adopt the image completion algorithm in [38] to fill textures in the occluded areas. Fig.4 shows the graph representation of a chair. For any two contact components, a set of sampling points is stored at the corresponding edge.

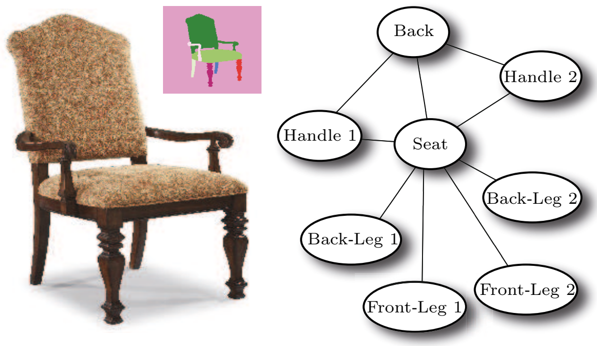


Fig.4. Graph representation of image object.

3.2 3D Cuboid Proxy Construction

While compositing a new image object in a particular viewpoint, our approach frequently leverages components from different viewpoints of the source images to form the new object. To realize this operation, we need the assistance of certain level of 3D information to achieve the effect of viewpoint transformation. Therefore, our approach estimates a 3D cuboid proxy for each component as its rough 3D information, which is a popular choice in recent geometric layout analysis of images[26]. In this subsection, we describe a new single-view camera calibration algorithm and a structure-aware proxy optimization algorithm to form the correct relationships between cuboid proxies.

Coordinate-Frame Based Camera Calibration. Our single-view camera calibration algorithm takes one 2D point and three 2D vectors as the projections of coordinate system origin and its three coordinate axes as the input. Compared with other camera calibration methods for single-view image, our method reduces the requirements of geometry information, e.g., three pairs of vanishing lines points in [39-40], or a number of corner points of geometric primitives in [26, 41-42]. Thus, it is more suitable for general image objects that have limited geometric clues. Note that most of man-made objects are up-right, which indicates a nearly vertical z axis. Meanwhile, parallel lines for one vanishing point can usually be found due to the symmetry in objects. Therefore, there is usually only one axis needed to be tuned subtly (e.g., y axis in Fig.5(a)). In case parallel lines are not present, users can interactively specify the 2D vectors as coordinate axes by trial-and-error in a short time.

The camera projection matrix is defined as: $\boldsymbol{M}_{3\times4} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$, where $\boldsymbol{K}$ is the camera intrinsic matrix (we assume the principle point is at the image center and focus f is the only unknown variable). $\boldsymbol{R} \in SO(3)$ and $\boldsymbol{t} \in \mathbb{R}^3$ represent the camera orientation and position respectively. The seven unknown parameters in $\boldsymbol{K}$, $\boldsymbol{R}$, $\boldsymbol{t}$ are solved using non-linear optimization (see Appendix). The orientation matrix is initialized by interactively aligning a front-view box to the image object. Our algorithm then computes the initial orientation matrix using its corner point projection information.

Structure-Aware 3D Proxy Fitting. With the estimated camera projection matrix $\boldsymbol{M}$, we can initialize axis-aligned cuboids using line grouping as in [26], where the silhouette of each component is approximated by a hexagon and the six hexagon edges are grouped according to vanishing point to form the axis-aligned cuboids. However, these independently estimated axis-aligned cuboids disperse in the 3D space without any structures. We thus optimize them to rebuild the structural relationships among those components (e.g., contact, symmetry). The energy function is as follows:

$$E(P_1, P_2, ..., P_N) = \omega_{\rm f}E_{\rm fitting} + \omega_{\rm u}E_{\rm unary} + \omega_{\rm p}E_{\rm pair}.$$

The first term $E_{\rm fitting}$ penalizes the deviation from initial cuboids, which is computed by accumulating the 2D distances between the projected corner-points of the initial and the optimized proxies:

$$E_{\rm fitting} = \sum_{i}^{N}\sum_{k}\left\|\boldsymbol{M}v^k - \boldsymbol{M}\overline{v}^k\right\|^2,$$

where N is the number of components, v^k and $\overline{v}^k$ are the corner points of the optimized cuboids P_i and the initial cuboids $\bar{P}_i$ respectively. Normalized homogeneous coordinates are used in the computation.

The unary term $E_{\rm unary}$ penalizes the deviation from the structural constraints defined on a single proxy. We mainly design two types of structural constraints $\{GlobReflection, OnGround\}$ for this term to maintain the correct relationship between components and the calibrated 3D coordinate system. *GlobReflection*

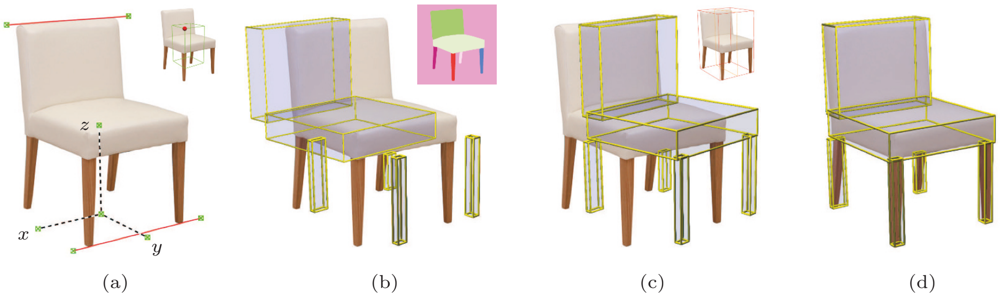


Fig.5. Based on (a) the coordinate-frame based camera calibration and (b) the initial object-on-ground fitting proxies, a structural optimization is carried out to build (d) the semantic structures of the chair, e.g., chair seat is on top of chair legs. We also show (c) the same 3D proxies in a rotated view.

indicates that the cuboid should be of reflectional symmetry with respect to a global coordinate plane. For instance, the chair back should be symmetric with respect to the yz plane as shown in Fig.5(a). *OnGround* indicates the cuboid should be on the ground plane, which is usually the xy plane in our coordinate system setup. We define the term as:

$$E_{\text{unary}} = \sum_{i \in \mathcal{R}_u} dist(c_i, yz)^2 + \sum_{i \in \mathcal{G}_u} \sum_{k=1}^{4} dist(v_i^k, xy)^2,$$

where $\mathcal{R}_u$ is the set of cuboids with *GlobReflection* constraints and $\mathcal{G}_u$ is the set of cuboids with *OnGround* constraints. *dist* is a function to compute point-to-plane distance, c_i is the center of cuboid proxy P_i and v_i^k is the corner point of the cuboid plane with minimum z value.

The pairwise term E_{pair} penalizes the deviation from the structural constraints between two cuboids. Three types of constraints $\{Symmetry, On, Side\}$ are imposed to a pair of cuboids in the optimization, where *Symmetry* requires the cuboids to be of reflectional symmetry with respect to a coordinate plane, while *On* and *Side* represent one cuboid should be on and beside another cuboid respectively. For example, the chair base should rest on the chair legs. The term is computed as:

$$\begin{aligned} E_{\text{pair}} = & \sum_{(i,j) \in \mathcal{R}_p} \|rf(c_i, p) - c_j\|^2 + \\ & \sum_{(i,j) \in \mathcal{O}_p} dist(bc_i, tp_j)^2 + \\ & \sum_{(i,j) \in \mathcal{S}_p} dist(sc_i, sp_j)^2, \end{aligned} \quad (1)$$

where $\mathcal{R}_p$ is the set of pairwise reflection *Symmetry* constraints, $\mathcal{O}_p$ the set of *On* constraints, and $\mathcal{S}_p$ the set of *Side* constraints. rf is a function to reflect a point with respect to plane p. The first term in (1) maintains the reflection symmetry constraint by requiring the reflected cuboid center to coincide with the symmetric cuboid center. The last two terms actually penalize the distance between the plane center of one cuboid and the top or side plane of the other cuboid to maintain the *On* or *Side* constraints. bc_i indicates the bottom plane center of P_i and tp_j the top plane of P_j. Similarly, sc_i indicates the side plane center of P_i and sp_j the side plane of P_j. Note that for axis aligned cuboids, the point to coordinate plane distance is just one coordinate component, which simplifies the equations a lot.

As the cuboids remain axis-aligned in the fitting, we only need to optimize six parameters, the scales and center position, for each cuboid. The vertex position involved in the energy function, such as corner points and plane center, can be easily derived through these six parameters. We minimize the total energy using Levenberg-Marquardt method. In practice, we set $\omega_{\text{f}} = 0.03$, $\omega_{\text{u}} = 10$ and $\omega_{\text{p}} = 3$.

Note that the structural constraints are only specified once for each image object category according to the component type information. For example, a typical pairwise constraint is defined as $(On, Seat, Frontleg)$ for the chair category, and the fitting procedure can automatically impose this constraint between seat and front-leg components when handling a new chair image object.

3.3 Proxy-Guided Component Compositing

After we estimate the camera parameters and fit the 3D cuboid proxies, we can conjoin the selected components together into a single object. We will first translate and scale each 3D proxy in 3D space. Then we will perform 2D image warping based on the transformed proxies.

Component Conjoining. We use the concept "slot" ([12, 43]) to conjoin the image components (Fig.6(a)). After the structure-aware 3D proxies construction, slot-pairs between 3D proxies of connected components are established. Each slot belongs to a host component. The slot defines: 1) to which component (label) the host component can connect; 2) the contact points where two components attach; 3) the size of the component connecting to the host.

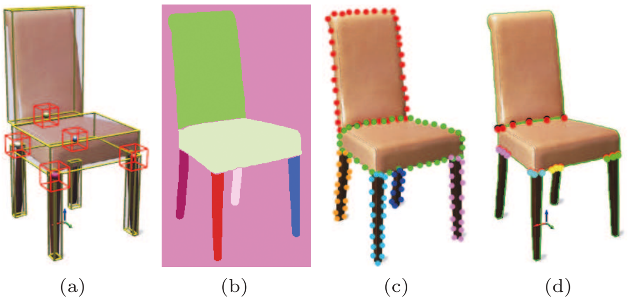


Fig.6. Key tokens for the compositing process. (a) 3D proxies (yellow cuboid) and slots (red cuboid). Each slot contains one 3D contact point. (b) 2D segmentation for the image components. (c) 2D reference points along the segmentation boundary for image warping. (d) 2D contact points for conjoining the image components.

To conjoin the proxies, we need 3D contact points. Each slot will have one 3D contact point (Fig.6(a)). We adopt the following strategies for 3D contact point computation: if two proxies have intersection, the contact point is the center of the intersection part; otherwise, we find the face on a smaller proxy that is closest to the bigger one, and use the center of this face as the contact point. Each contact point is separately stored into corresponding slots of the two connected 3D proxies.

The last step is to conjoin the image components by 2D warping. The warping function for each component is represented by a 2D affine transformation that is estimated from a pair of 2D point sets. The reference points are obtained from the segmentation of the 2D components, as shown in Figs.6(b) and 6(c). Finally we apply the warping function to the 2D contact points for each slot. Notice that each slot can have multiple 2D contact points in contrast to a single 3D contact point. Fig.6(d) shows the contact points in their respective slots.

The detailed processes are as follows.

1) *Conjoining 3D Proxies.* We conjoin 3D proxies by optimizing the position and size of each suggested proxy P_i constrained by its contacting proxies. Let c_i, l_i and p_i^k be the center, the size and the contact point of slot k of P_i respectively. We want to transform P_i and hence its contact point p_i^k by a 3D rigid transformation, i.e., $T_i \times p_i^k = \Lambda_i \times (p_i^k - c_i) + c_i + t_i$, where $\Lambda_i = \mathrm{diag}(s_i)$, s_i and t_i are the scale and the translation components respectively. In addition, the transformation applied on P_i is constrained by the size and position of its connecting proxies. Let ll_i^k be the size of the proxy connecting to slot k of P_i in the original image. The contact energy term can hence be defined as:

$$\begin{aligned} E_{\mathrm{c}} = \sum_{(i,j)\in\mathcal{M}} \big(&\|T_i \times p_i^{m_i} - T_j \times p_j^{m_j}\|^2 + \\ &\|\Lambda_i \times l_i - \Lambda_j \times ll_j^{m_j}\|^2 + \\ &\|\Lambda_i \times ll_i^{m_i} - \Lambda_j \times l_j\|^2\big), \end{aligned}$$

where $\mathcal{M}$ is the set of proxy-pairs with matched slots, and m_i and m_j are the indices of the matched slots in their host proxies respectively. This term brings together pairs of contact points and makes connected proxies have compatible sizes. We also add two shape preserving terms, the scaling energy term E_{s} and the translation energy term E_{t}, to avoid large proxy deformation as in [43]:

$$E_{\mathrm{s}} = \sum_i \|\boldsymbol{s}_i - [1,1,1]^{\mathrm{T}}\|^2, \quad E_{\mathrm{t}} = \sum_i \|t_i\|^2.$$

Our goal is to find the optimal transformation $\boldsymbol{T}_i^*$ for each proxy P_i by minimizing the follow function:

$$\boldsymbol{T}_i^* = \arg\min_{T_i} \omega_{\mathrm{c}} E_{\mathrm{c}} + \omega_{\mathrm{s}} E_{\mathrm{s}} + \omega_{\mathrm{t}} E_{\mathrm{t}}. \quad (2)$$

In practice, we set $\omega_{\mathrm{c}} = 1$, $\omega_{\mathrm{s}} = 0.5$ and $\omega_{\mathrm{t}} = 0.1$. Fig.3 shows the results of the 3D proxy conjoining optimization. Note that symmetries are automatically ensured by our slot definition.

2) *Warping and Conjoining 2D Image Components.* After we conjoin the 3D proxies, we can warp and conjoin the underneath image components to obtain the complete image object. When we construct the 3D proxies from the images, we compute a set of n_i 2D reference points $\{\hat{a}_{i,r} \mid r = 1, 2, \ldots, n_i\}$ by uniformly sampling along the segmented image boundary of each proxy P_i. We also project $\{\hat{a}_{i,r}\}$ to visible faces of the proxies to get the 3D reference points and discard any points outside the 2D projection boundary of the proxy. We set $n_i = 200$ for all our experiments.

In order to obtain an image warping that faithfully resembles the visual effect from the 3D proxies transformation, we need a set of 2D target points $\{\hat{b}_{i,r}\}$ that comprise the information from (2). It is simply done by transforming the 3D reference points using the transformation from (2) to obtain the 3D target points and then re-projecting them back to the 2D space with the camera setting of the image containing the base component, where the base component is usually the component of the largest size of an image object category (e.g., the seat component of chair image objects).

Given a limited view change of the 3D proxies, we adopt a 2D affine transformation ($\boldsymbol{A}_i$) for the image warping. The optimal affine transformation matrix $\boldsymbol{A}_i^*$ is solved by minimizing the distance between the warped reference points and the target points:

$$\boldsymbol{A}_i^* = \arg\min_{\boldsymbol{A}_i} \sum_r \|\boldsymbol{A}_i \times \hat{a}_{i,r} - \hat{b}_{i,r}\|^2.$$

$\boldsymbol{A}_i^*$ is essentially the 2D counterpart of $\boldsymbol{T}_i^*$. The warped image components are ready for final conjoining. Fig.3 shows the image warping results.

To conjoin the warped 2D image components, for each slot k in each component i, we define a set of n_i^k 2D contact points $\{\hat{p}_{i,r}^k \mid r = 1, 2, \ldots, n_i^k\}$ and warp them by $\boldsymbol{A}_i^*$. Then we adopt breadth-first search (BFS) procedure to translate the image components. The largest component is pushed into a queue as base component. Each time when a component is popped from the queue, the un-accessed components connected to it are translated by the 2D contact points in slots, and are pushed into the queue accordingly. The procedure ends when all components are already accessed. In each translation step, say, component i is translated towards component j via their matched slots m_i and m_j, the center of $\{\hat{p}^{m_i}_{i,r}\}$ and the center of $\{\hat{p}^{m_j}_{j,r}\}$ are matched first, and then points $\{\hat{p}^{m_i}_{i,r}\}$ that lay outside component j's segmentation boundary are iteratively translated to their closest points on the boundary. Although there may be cycles in component connection graph, we find that the above greedy strategy works well in practice. Fig.3 shows results of 2D conjoining.

Color Optimization. The image components for compositing are chosen without the consideration of color. Though it is subjective to some extent, very likely the composited objects consist of "unmatchable" or "uncomfortable" colors. We do color optimization by leveraging both the color compatibility model[44] (as prior) and the data-driven palettes (as examples). In a preprocessing stage, we first extract a 5-color theme using k-means from each image in a dataset, and then we do k-means again on colors from all these 5-color themes to generate a 40-color palette. In the composited image, the major color of the largest component is selected, and its hue is assigned to the data-driven palette with variation σ to generate a new palette (see Fig.7). We adopt a similar color optimization procedure as in [45] to choose a set of colors with maximum compatibility score from the new palette. Each image component is then assigned with its optimal color by color transfer method.

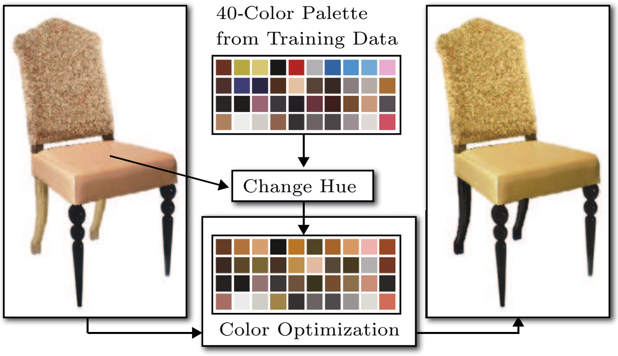


Fig.7. We generate a new color palette by assigning the major hue from the seat to the data-driven palette. We then optimize the color using the similar optimization procedure as in [45] to obtain our final result.

4 Image Object Synthesis

Our view-aware image object compositing technique creates new research possibilities. One particular application we propose here is image object synthesis (see Fig.8). Structure-ware 3D shape synthesis recently attracts much research attention in the computer graphics community. Similarly, we would like to generate new image objects by combining components from an input images dataset. Compared with 3D shape synthesis, image object synthesis has a unique research value for several reasons. First, unlike 3D shapes which often require expensive scanning devices to acquire or sophisticated techniques to model, image dataset is much easier to acquire with digital cameras nowadays. There are also huge resources of images from Internet, which can be quickly accessed via search engines. This implies

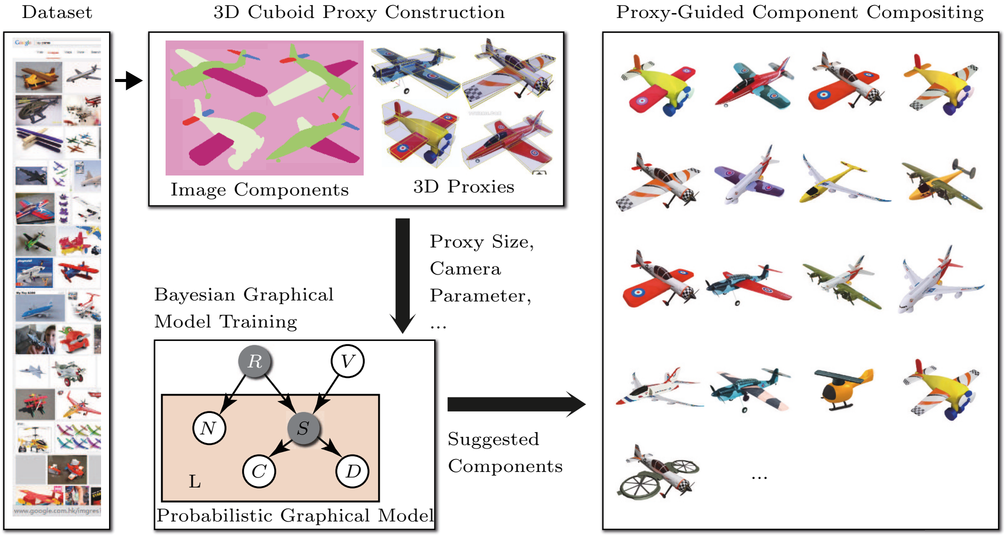


Fig.8. Overview of our image object synthesis approach. L: component category.

that we can get a wider variety of real-world objects with images. Second, objects synthesized from images contain abundant color or appearance information inherited from the source images, which is deficient in existing 3D shape synthesis. Such information could be very important to creative object design and may greatly affect the perception of designers. Finally, the synthesized image objects can inspire shape design or be regarded as the guidance of 3D modeling, and with the recent photo-based 3D modeling technique[46], one can generate 3D shapes conforming to the synthesized image objects.

4.1 Bayesian Graphical Model Training

Given a small image dataset of a specific type of object, the goal of Bayesian graphical model training is to build its probabilistic generative model so that we can sample this model to create new image objects. Similar to [12], hidden variables are used to represent the overall object structure and component styles, while observed variables are used to represent the geometric descriptors for each component and their adjacency. The key difference is that we introduce an observed variable to account for the viewpoint information from different images. The overall structure of the graphical model is shown in Fig.8. We start by introducing notations for the set of random variables used in the graphical model and briefly describe the training algorithm.

Notation. The random variables used in the graphical model are listed in Table 1. The observed variable V represents the viewpoint parameters. To ease the learning algorithm, the viewpoint parameters, i.e., the estimated seven parameters in single view camera calibration computed in 3D proxy construction, are first clustered using mean shift algorithm by setting the radius to be 0.2. V takes integer value to indicate the cluster index. The geometry feature vector $\boldsymbol{C}_l$ of a component category l includes the size of its 3D proxy and one feature vector to represent the point distribution model (PDM) of its 2D silhouettes[47], where 4~6 key points are consistently labeled on the component silhouettes. Mathematically, in PDM, a new 2D polygon shape $\boldsymbol{Y}$ is defined as $\boldsymbol{Y} = \bar{\boldsymbol{Y}} + \boldsymbol{A}\boldsymbol{b}$, where $\bar{\boldsymbol{Y}}$ is the mean shape and $\boldsymbol{b}$ is the vector of weights for each principal component in basis $\boldsymbol{A}$ PCA obtained. To account for viewpoint change, we compute the principal components for each viewpoint cluster and arrange them in an order consistent to the cluster index. $\boldsymbol{b}$ is organized accordingly to create a feature vector with same dimension.

Table 1. Notation of Random Variables Used in the Bayesian Graphical Model

Notation	Domain	Interpretation
R	$R \in \mathbb{Z}^+$	Structure/shape style, latent variable
V	$V \in \mathbb{Z}^+$	View point, observed variable
$S = \{S_l\}$	$S_l \in \mathbb{N}^0$	Component style, 0 means no component in the category l, latent variable
$N = \{N_l\}$	$N_l \in \mathbb{N}^0$	Number of components from category l, observed variable
$C = \{\boldsymbol{C}_l\}$	$\boldsymbol{C}_l \in \{\mathbb{R}^{dim_l}\}$	Continuous geometry feature vector of the component, observed variable
$D = \{\boldsymbol{D}_l\}$	$\boldsymbol{D}_l \in \{\mathbb{Z}^L\}$	Discrete vectors, encode the number of components from each category connected to components from category l, observed variable

The latent variables R and S are learned from the training data, while the others are observed directly. The joint probability distribution is factorized as conditional probability distributions (CPDs) product:

$$P(X) = P(R)P(V)\prod_{l\in\mathcal{L}}\big(P(N_l|R)P(S_l|R,V)$$
$$P(\boldsymbol{C}_l|S_l)P(\boldsymbol{D}_l|S_l)\big).$$

Note that the lateral edges used in the model of Kalogerakis *et al.*[12] are not involved here. We do this simplification since it makes the factorization more compact and increases the variations of synthesis results.

Training. After the preprocessing stage, a set of feature vectors $\mathcal{O} = \{\boldsymbol{O}_1, \boldsymbol{O}_2, \dots, \boldsymbol{O}_K\}$, where $\boldsymbol{O}_K = \{V_K, N_K, C_K, D_K\}$, is extracted from K segmented images as the training data. To learn the graph structure (domain sizes of latent variables) and all the CPDs parameters of the model, we maximize the following maximum likelihood function:

$$J = \ln P(O|G) \simeq \ln P(O|G, \tilde{\theta}_G) - \frac{1}{2}m_\theta \ln K,$$

where the Bayesian Information Criterion score[48] is used to select the graph structure G (domain sizes) that best describes the training data. Here, $\tilde{\theta}_G$ is the maximum a posteriori (MAP) estimation of parameters for a given G, m_θ is the number of independent parameters in the network, and K is the data size. For a particular structure of G, we use expectation-maximization (EM) algorithm to estimate $\tilde{\theta}_G$ with the following MAP function:

$$\tilde{\theta}_G = \arg\max_\theta P(O|G,\theta)P(\theta|G),$$

where $P(\theta|G)$ is the prior distribution of parameter θ of graphical model.

We perform greedy search to find the structure G with maximum J value by increasing the domain sizes, i.e., the number of discrete values for latent variables R and S. For more details on the EM algorithm, please refer to Appendix.

4.2 Synthesis

Our image object synthesis is divided into three steps. First, we will determine the set of components to be used for synthesis. Second, we will conjoin the suggested components into a single object. Third, we will optimize the color of the synthesized objects. The latter two steps are achieved by using our image object compositing method proposed in Section 3.

Component Set Synthesizing. Mathematically, different sets of components can be seen as distinct samples of the probabilistic model. As the lateral edges are not involved, we simply adopt a depth-first search (DFS) procedure to explore the shape space of image objects. Starting from root variable R, each random variable in the searching path is partially assigned with its possible values accordingly. Similar to the deterministic method of Kalogerakis *et al.*[12], we prune the searching branches with assignments probability lower than a threshold (10^{-10} in our implementation). To ensure the searching feasibility, the continuous variables $\boldsymbol{C}_l$ are only assigned with values corresponding to existing components from training data. In valid samples found by the searching procedure, the assignments to variables $\boldsymbol{C}_l$ determine the set of components used for synthesis.

5 Experimental Results

We tested the described compositing technique and synthesis application on five different types of image objects: chair, cup, lamp, robot and toy plane. We collected around 150 images from Internet, and each of them was segmented and analyzed as described in Subsection 3.1 for compositing and synthesizing new image objects. The details of the datasets are given in Table 2. In total, we synthesized more than 1 000 new image objects and the majority results are shown in Figs.9∼12 (see Appendix for more results in high-resolution). Fig.13 shows two histograms regarding some statistics of the results. The first one is the number of components used per synthesized image object and the second one is the number of source images which contribute components per synthesized image object. From the histograms, we can see that most of the synthesized shapes contain 3∼6 components and most of the shapes are synthesized from 2∼4 source images.

Table 2. Datasets Used in Image Object Synthesis Experiments

	Chair	Cup	Lamp	Robot	Toy Plane
Number of training data	42	22	30	23	15
Number of categories	6	3	4	5	4
Number of components	243	44	90	130	63

In Fig.2, we compared the compositing results generated without and with consideration of viewpoint for chair models. The composited chair consists of six components coming from four source images (Fig.2(a)). Directly stitching these components together without our

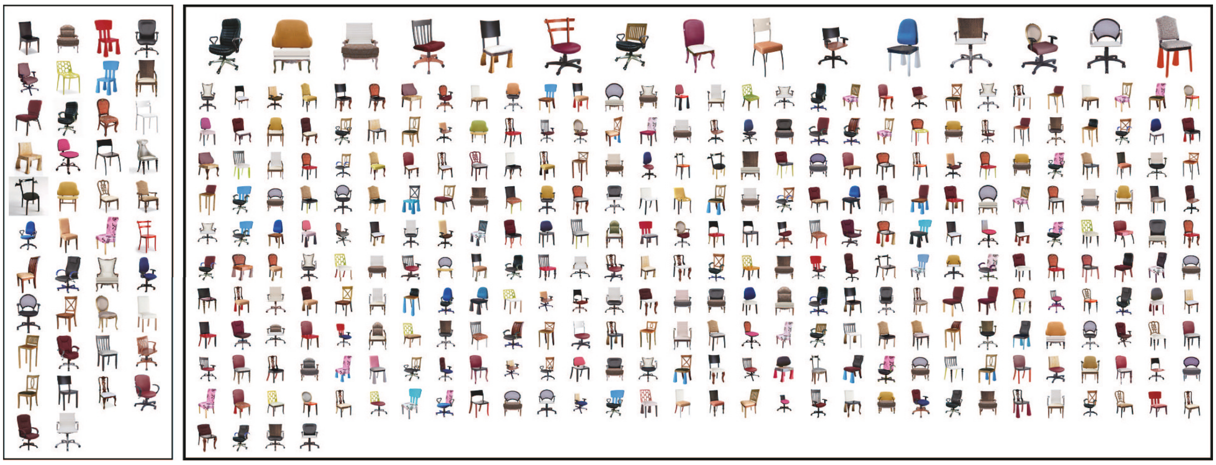

(a) (b)

Fig.9. Chair. (a) 42 input chair images. (b) 259 synthesized chairs.

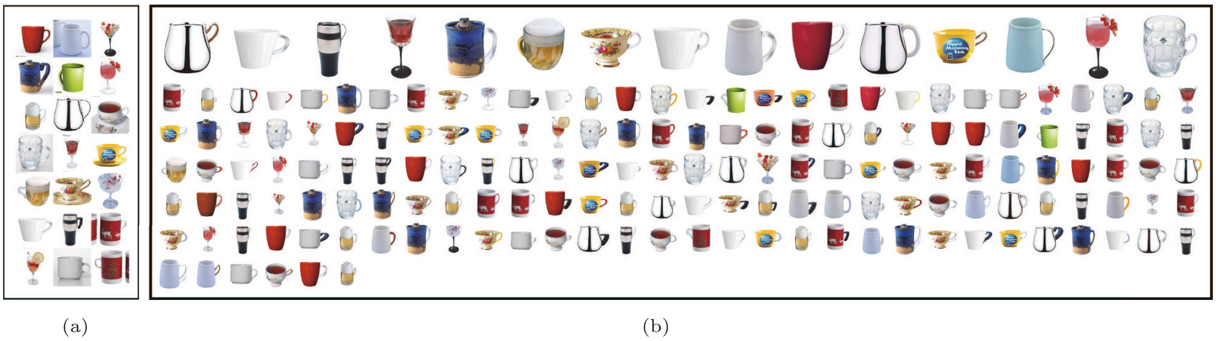


Fig.10. Cup. (a) 21 input cup images. (b) 171 synthesized cups.

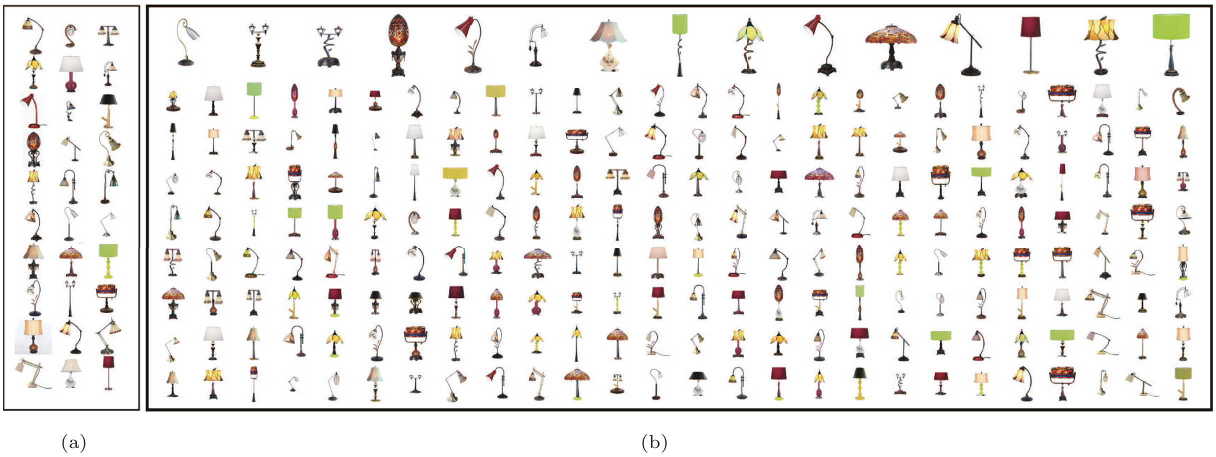


Fig.11. Lamp. (a) 30 input lamp images. (b) 147 synthesized lamps.

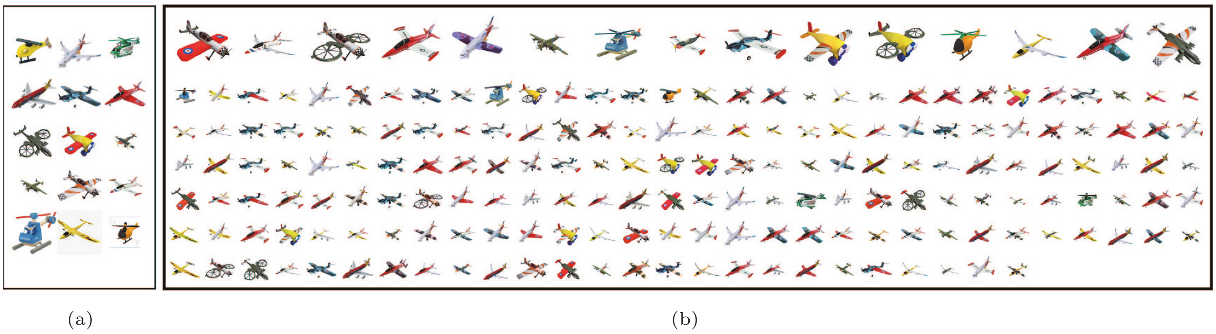
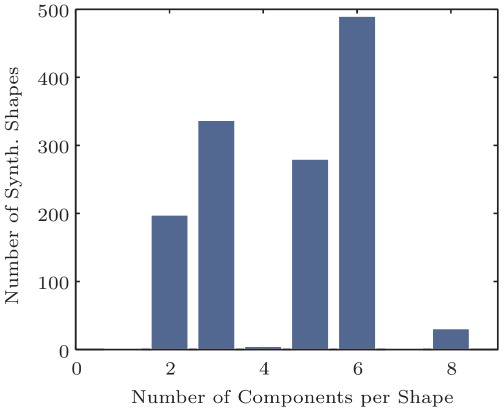


Fig.12. Toy Plane. (a) 15 input toy plane images. (b) 190 synthesized toy planes.

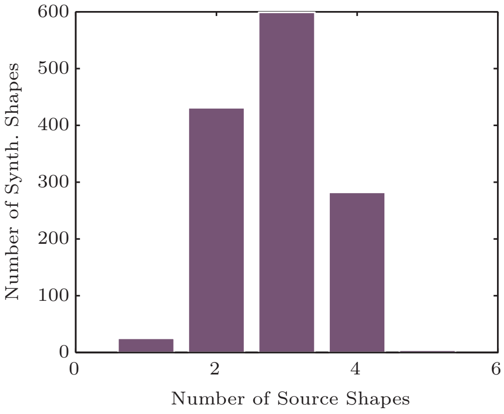
proxy guided warping produces an unsatisfactory result (Fig.2(b)) which appears to be distorted and does not look like a real chair. Our view-ware compositing algorithm generates a more realistic result. Fig.14 shows another comparison for toy planes.

User Study. To evaluate whether our synthesized image objects are both plausible and novel, we recruited 68 subjects who are mostly students majored in computer science and digital media. These subjects were divided into two groups and assigned one of the following two tasks respectively.

*T*1: *Design Preference Test*. The subjects are shown with one training image and one synthesized image side by side in random order, and the source information, i.e., training or synthesized, of these two images is blind to them to avoid possible bias. The subjects are required to answer which image in each pair is more preferable as a design reference. The test set consists of 50 randomly sampled pairs of such images, 10 for each object category.

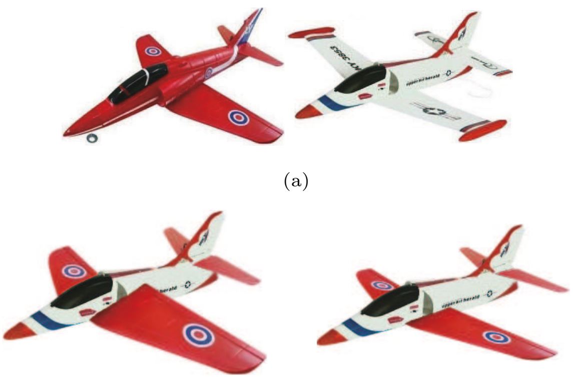


(a)



(b)

Fig.13. (a) Number of components used per synthesized (synth.) image object. (b) Number of source images per synthesized object.



(a)

(b) (c)

Fig.14. Compositings of toy plane images with/without consideration of viewpoints. (a) Source images. (b) Direct compositing. (c) View-aware compositing.

T2: *Creativity Test.* Compared with the reference set of image objects shown on the left window of the user study UI, the subjects are required to determine whether the displayed object on the right window of the user study UI is new from the set. There will be 210 objects in total.

The statistics of design preference test T1 tell us the subjects choose 48%(694) synthesized objects against 52%(756) training images (no significant difference), which reveals that our synthesis results are preferred at a comparable probability to the real-world images.

The test set used in creativity test T2 includes 50 chairs, 30 cups, 43 lamps, 47 robots and 40 toy planes uniformly chosen from the synthesis results. All the images from the training dataset are used as the reference set of images, and the subjects need to judge whether a synthesized object is a new object, not present in the training set. The statistics of true and false choices for each object category are shown in Fig.15. The highest and the lowest positive rates are robot and cup with about 90% and 79% choices respectively. Thus, a significant amount of the synthesized image objects are novel to participants, and should be valuable to inspire creative design.

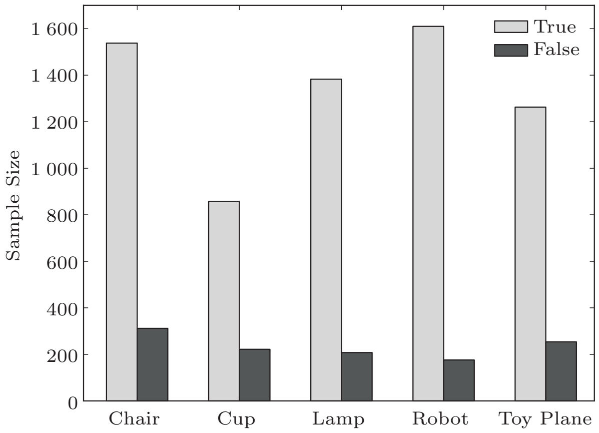


Fig.15. Statistics of true (new object) and false choices for each category in the creativity test.

Running Time. We implemented our method on a PC with Intel® Core$^{\mathrm{TM}}$2 Quad CPU Q9400 and Win7 operating system. The average computation time for component stitching and color optimization is about four seconds and one second respectively. For the image object synthesis, the Bayesian graphical model training took about 20 minutes for chair, five minutes for cup, 12 minutes for lamp, 15 minutes for robot and 3 minutes for toy plane, and enumerating the component set for synthesizing took from 20 seconds to one minute.

User Interaction. Our approach needs some user interactions to assist the construction of source image objects. In general, the user interaction required in the construction of image object representation is around six minutes on average. Specifically, image object component segmentation and annotation are done in about four minutes for most of the images used in our experiments, and it usually takes the user 1∼2 minutes to tune the axis projection on an image in camera calibration. The axis tuning time may slightly increase for non-upright object since its upright z axis needs to be carefully tuned to match its image.

Limitations. There are still difficult situations which are not well handled in our current approach, and we summarize them as follows: 1) the affine transformation we adopt for image component warping can only handle limited viewpoint changes; 2) we cannot use image objects with severe occlusions as source, since it may simply lead to unpleasant proxy fitting or image completion results; 3) we do not handle strong illumination effects or complex shading in our current implementation.

6 Conclusions and Future Work

We introduced a view-aware approach to compositing novel image objects from multiple-source images that have different viewpoints. Compared with previous image compositing techniques that work at the image object level, our approach operates on the lower level of object components. It warps selected components using the fitted 3D proxies according to the estimated camera parameters, and stitches the warped components to generate a new image object. We believe this component-level view-aware compositing operation nicely complements existing image compositing tools.

Based on this compositing approach, we further developed an analysis-and-synthesis technique to automatically create image objects from a set of exemplars. It works by seamlessly combining components of input image objects with respect to the structural constraints described by a probabilistic graphical model trained from the input dataset. We regard our work as the first image object synthesis method that operates on the component level.

Component-based image object compositing and synthesis not only help enrich existing image contents but also provide innovative means for creative object design and even 3D modeling. However, our work is only an initial attempt to solve this problem and there exist many interesting directions for future work. The current approach relies on user interaction for semantic analysis of image objects. To reduce the analysis workload, we plan to explore transductive learning algorithms to automatically segment the image objects into components using our labeled dataset. A possible solution is to adopt diffusion algorithms to transfer the labeling results. Furthermore, we are interested in exploring the compatibility of components from different objects to increase the variation of the synthesized objects, which is important to creative design.

References

[1] Perez P, Gangnet M, Blake A. Poisson image editing. *ACM Transactions on Graphics*, 2003, 22(3): 313-318.

[2] Jia J, Sun J, Tang C K, Shum H Y. Drag-and-drop pasting. *ACM Transactions on Graphics*, 2006, 25(3): 631-637.

[3] Farbman Z, Hoffer G, Lipman Y, Cohen-Or D, Lischinski D. Coordinates for instant image cloning. *ACM Transactions on Graphics*, 2009, 28(3): Article No. 67.

[4] Tao M W, Johnson M K, Paris S. Error-tolerant image compositing. In *Proc. the 11th European Conference on Computer Vision*, Sept. 2010, pp.31-44.

[5] Sunkavalli K, Johnson M K, Matusik W, Pfister H. Multiscale image harmonization. *ACM Transactions on Graphics*, 2010, 29(4): Article No. 125.

[6] Agarwala A, Dontcheva M, Agrawala M, Drucker S, Colburn A, Curless B, Salesin D, Cohen M. Interactive digital photomontage. *ACM Transactions on Graphics*, 2004, 23(3): 294-302.

[7] Rother C, Kumar S, Kolmogorov V, Blake A. Digital tapestry [automatic image synthesis]. In *Proc. IEEE CVPR*, June 2005, pp.589-596.

[8] Rother C, Bordeaux L, Hamadi Y, Blake A. AutoCollage. *ACM Transactions on Graphics*, 2006, 25(3): 847-852.

[9] Wang J, Quan L, Sun J, Tang X, Shum H Y. Picture collage. In *Proc. IEEE CVPR*, June 2006, pp.347-354.

[10] Chen T, Cheng M M, Tan P, Shamir A, Hu S M. Sketch2Photo: Internet image montage. *ACM Transactions on Graphics*, 2009, 28(5): 124:1-124:10.

[11] Eitz M, Richter R, Hildebrand K, Boubekeur T, Alexa M. Photosketcher: Interactive sketch-based image synthesis. *IEEE Computer Graphics and Applications*, 2011, 31(6): 56-66.

[12] Kalogerakis E, Chaudhuri S, Koller D, Koltun V. A probabilistic model for component-based shape synthesis. *ACM Trans. Graph.*, 2012, 31(4): 55:1-55:11.

[13] Xu K, Zhang H, Cohen-Or D, Chen B. Fit and diverse: Set evolution for inspiring 3D shape galleries. *ACM Trans. Graph.*, 2012, 31(4): 57:1-57:10.

[14] Burt P J, Adelson E H. A multiresolution spline with application to image mosaics. *ACM Trans. Graph.*, 1983, 2(4): 217-236.

[15] Ogden J M, Adelson E H, Bergen J R, Burt P J. Pyramid-based computer graphics. *RCA Engineer*, 1985, 30(5): 4-15.

[16] Porter T, Duff T. Compositing digital images. *ACM SIGGRAPH Comput. Graph.*, 1984, 18(3): 253-259.

[17] Xue S, Agarwala A, Dorsey J, Rushmeier H. Understanding and improving the realism of image composites. *ACM Transactions on Graphics*, 2012, 31(4): Article No. 84.

[18] Diakopoulos N, Essa I, Jain R. Content based image synthesis. In *Proc. the 3rd CIVR*, July 2004, pp.299-307.

[19] Johnson M, Brostow G J, Shotton J *et al*. Semantic photo synthesis. *Computer Graphics Forum*, 2006, 25(3): 407-413.

[20] Lalonde J F, Hoiem D, Efros A A, Rother C, Winn J, Criminisi A. Photo clip art. *ACM Transactions on Graphics*, 2007, 26(3): Article No. 3.

[21] Hall P, Cai H, Wu Q, Corradi T. Cross-depiction problem: Recognition and synthesis of photographs and artwork. *Computational Visual Media*, 2015, 1(2): 91-103.

[22] Huang H, Zhang L, Zhang H C. Arcimboldo-like collage using internet images. *ACM Transactions on Graphics*, 2011, 30(6): Article No. 155.

[23] Yu Z, Lu L, Guo Y, Fan R, Liu M, Wang W. Content-aware photo collage using circle packing. *IEEE Transactions on Visualization and Computer Graphics*, 2014, 20(2): 182-195.

[24] Risser E, Han C, Dahyot R, Grinspun E. Synthesizing structured image hybrids. *ACM Transactions on Graphics*, 2010, 29(4): Article No. 85.

[25] Carroll R, Agarwala A, Agrawala M. Image warps for artistic perspective manipulation. *ACM Transactions on Graphics*, 2010, 29(4): Article No. 127.

[26] Zheng Y, Chen X, Cheng M M, Zhou K, Hu S M, Mitra N J. Interactive images: Cuboid proxies for smart image manipulation. *ACM Trans. Graph.*, 2012, 31(4): 99:1-99:11.

[27] Chen T, Zhu Z, Shamir A, Hu S M, Cohen-Or D. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics*, 2013, 32(6): Article No. 195.

[28] Miao Y, Hu F, Zhang X, Chen J, Pajarola R. SymmSketch: Creating symmetric 3D free-form shapes from 2D sketches. *Computational Visual Media*, 2015, 1(1): 3-16.

[29] Funkhouser T, Kazhdan M, Shilane P, Min P, Kiefer W, Tal A, Rusinkiewicz S, Dobkin D. Modeling by example. *ACM Trans. Graph.*, 2004, 23(3): 652-663.

[30] Shin H, Igarashi T. Magic canvas: Interactive design of a 3-D scene prototype from freehand sketches. In *Proc. Graphics Interface*, May 2007, pp.63-70.

[31] Lee J, Funkhouser T. Sketch-based search and composition of 3D models. In *Proc. the 5th SBM*, June 2008, pp.97-104.

[32] Xu K, Chen K, Fu H, Sun W L, Hu S M. Sketch2Scene: Sketch-based co-retrieval and co-placement of 3D models. *ACM Transactions on Graphics*, 2013, 32(4): Article No. 123.

[33] Kreavoy V, Julius D, Sheffer A. Model composition from interchangeable components. In *Proc. the 15th PG*, Oct. 2007, pp.129-138.

[34] Chaudhuri S, Koltun V. Data-driven suggestions for creativity support in 3D modeling. *ACM Trans. Graph.*, 2010, 29(6): 183:1-183:10.

[35] Chaudhuri S, Kalogerakis E, Guibas L, Koltun V. Probabilistic reasoning for assembly-based 3D modeling. *ACM Trans. Graph.*, 2011, 30(4): 35:1-35:10.

[36] Li Y, Sun J, Tang C K, Shum H Y. Lazy snapping. *ACM Transactions on Graphics*, 2004, 23(3): 303-308.

[37] Russell B C, Torralba A, Murphy K P, Freeman W T. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision*, 2008, 77(1/2/3): 157-173.

[38] Barnes C, Shechtman E, Finkelstein A, Goldman D B. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics*, 2009, 28(3): 24:1-24:11.

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

[40] Sinha S N, Steedly D, Szeliski R, Agrawala M, Pollefeys M. Interactive 3D architectural modeling from unordered photo collections. *ACM Transactions on Graphics*, 2008, 27(5): 159:1-159:10.

[41] Wilczkowiak M, Sturm P, Boyer E. Using geometric constraints through parallelepipeds for calibration and 3D modeling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2005, 27(2): 194-207.

[42] Jiang N, Tan P, Cheong L F. Symmetric architecture modeling with a single image. *ACM Transactions on Graphics*, 2009, 28(5): 113:1-113:8.

[43] Shen C H, Fu H, Chen K, Hu S M. Structure recovery by part assembly. *ACM Transactions on Graphics*, 2012, 31(6): Article No. 180.

[44] O'Donovan P, Agarwala A, Hertzmann A. Color compatibility from large datasets. *ACM Transactions on Graphics*, 2011, 30(4): Article No. 63.

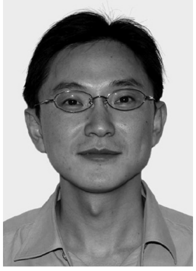
[45] Yu L F, Yeung S K, Terzopoulos D, Chan T F. DressUp!: Outfit synthesis through automatic optimization. *ACM Transactions on Graphics*, 2012, 31(6): 134:1-134:14.

[46] Xu K, Zheng H, Zhang H, Cohen-Or D, Liu L, Xiong Y. Photo-inspired model-driven 3D object modeling. *ACM Trans. Graph.*, 2011, 30(4): 80:1-80:10.

[47] Cootes T F, Taylor C J, Cooper D H, Graham J *et al*. Active shape models — Their training and application. *Computer Vision and Image Understanding*, 1995, 61(1): 38-59.

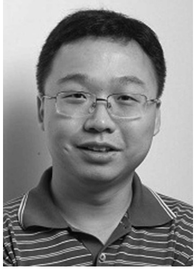
[48] Schwarz G. Estimating the dimension of a model. *The Annals of Statistics*, 1978, 6(2): 461-464.

Xiang Chen is an assistant professor in the State Key Laboratory of Computer Aided Design and Computer Graphics (CAD&CG), Zhejiang University, Hangzhou. He received his Ph.D. degree in computer science from Zhejiang University, Hangzhou, in 2012. His current research interests mainly include fabrication-aware design, image analysis/editing, shape modeling/retrieval and computer-aided design.



Wei-Wei Xu is a researcher at the Department of Computer Science, Zhejiang University, Hangzhou. Before that, Dr. Xu was a Qianjiang professor at Hangzhou Normal University from Sept. 2012 to Mar. 2016 and a researcher in Internet Graphics Group, Microsoft Research Asia, Beijing, from Oct. 2005 to June 2012. He was a postdoctoral researcher at Ritsumeikan University in Japan from 2004 to 2005. He received his Ph.D. degree in computer graphics from Zhejiang University, Hangzhou, in 2003, and B.S. degree and Master's degree in computer science from Hohai University, Nanjing, in 1996 and 1999 respectively. He has published more than 40 papers on international conference and journals, including more than 10 papers on ACM Transactions on Graphics. His main research interests include digital geometry processing and computer simulation techniques. He is now focusing on how to enhance the geometry design algorithm through the integration of physical properties.

Sai-Kit Yeung is an assistant professor at the Singapore University of Technology and Design (SUTD), where he leads the Vision, Graphics and Computational Design (VGD) Group. He was also a visiting assistant professor at MIT and Stanford University. Before joining SUTD, he was a postdoctoral scholar in the Department of Mathematics, University of California, Los Angeles (UCLA). He was also a visiting student at the Image Processing Research Group at UCLA in 2008 and at the Image Sciences Institute, University Medical Center Utrecht, the Netherlands, in 2007. He received his Ph.D. degree in electronic and computer engineering from the Hong Kong University of Science and Technology (HKUST) in 2009. He also received his B.E. degree (First Class Honors) in computer engineering in 2003 and M.Phil. degree in bioengineering in 2005, both from HKUST. Dr. Yeung's research expertise is in the areas of computer vision and computer graphics. His current research focus is on scene acquisition, scene understanding, functional realistic scene re-modeling, and computational fabrication. He is a member of IEEE and the IEEE Computer Society.



Kun Zhou is a Cheung Kong Professor in the Computer Science Department of Zhejiang University, Hangzhou, and the director of the State Key Laboratory of CAD&CG. Prior to joining Zhejiang University in 2008, Dr. Zhou was a leader researcher of the Internet Graphics Group at Microsoft Research Asia, Beijing. He received his B.S. degree and Ph.D. degree in computer science from Zhejiang University in 1997 and 2002, respectively. His research interests are in visual computing, parallel computing, human computer interaction, and virtual reality. He currently serves on the editorial/advisory boards of ACM Transactions on Graphics and IEEE Spectrum. He is a fellow of IEEE.

Appendix

A.1 Derivations for Coordinate-Frame Based Camera Calibration

The camera projection matrix $\boldsymbol{M}$ is usually defined as the product of three matrices: $\boldsymbol{M}_{3\times4} = \boldsymbol{K}[\boldsymbol{R}|\boldsymbol{t}]$. We start by introducing how we parameterize each matrix:

$$\boldsymbol{K} = \begin{pmatrix} f & & u \\ & f & v \\ & & 1 \end{pmatrix}.$$

We set $\{u, v\}$ to be image center, and thus, only focus parameter f left. $\boldsymbol{R}$ is an orthogonal matrix, which is parameterized Euler angles with ZYX rotation order. $\boldsymbol{t}$ is the translation freedom of the camera in the world coordinate system, where $\boldsymbol{t} = \{\boldsymbol{t}_x, \boldsymbol{t}_y, \boldsymbol{t}_z\}$. Therefore, there are totally 7 DOFs in our camera matrix setup.

The projection of the coordinate system, such as origin and coordinate axes, is represented by homogeneous coordinates. Let us denote the projection of the 3D coordinate system origin by $\boldsymbol{P}_o$, and the image position of the 3D position (0, 0, 1) by $\boldsymbol{P}_{up}$, where their entries are:

$$\boldsymbol{P}_o = \begin{pmatrix} \boldsymbol{o}_1 \\ \boldsymbol{o}_2 \\ 1 \end{pmatrix}, \quad \boldsymbol{P}_{up} = \begin{pmatrix} \boldsymbol{z}_1 \\ \boldsymbol{z}_2 \\ 1 \end{pmatrix}.$$

Further, let us denote the projection of 3D coordinate axes $\{x, y, z\}$ by $\{\boldsymbol{l}_x, \boldsymbol{l}_y, \boldsymbol{l}_z\}$ respectively. Their entries are defined as follows:

$$\boldsymbol{l}_x = \begin{pmatrix} l_{x1} \\ l_{x2} \\ 1 \end{pmatrix}, \quad \boldsymbol{l}_y = \begin{pmatrix} l_{y1} \\ l_{y2} \\ 1 \end{pmatrix}, \quad \boldsymbol{l}_z = \begin{pmatrix} l_{z1} \\ l_{z2} \\ 1 \end{pmatrix}.$$

According to projective geometry, we can set up the following equations:

$$\boldsymbol{M}\begin{pmatrix}1\\0\\0\\0\end{pmatrix}\cdot \boldsymbol{l}_x=0,\quad \boldsymbol{M}\begin{pmatrix}0\\1\\0\\0\end{pmatrix}\cdot \boldsymbol{l}_y=0,$$

$$\boldsymbol{M}\begin{pmatrix}0\\0\\1\\0\end{pmatrix}\cdot \boldsymbol{l}_z=0,\quad \boldsymbol{M}\begin{pmatrix}0\\0\\0\\1\end{pmatrix}\simeq \boldsymbol{P}_o,$$

$$\boldsymbol{M}\begin{pmatrix}0\\0\\1\\1\end{pmatrix}\simeq \boldsymbol{P}_{up},$$

where $\simeq$ means equality up to a scale. They are developed into the following seven equations:

$$\begin{aligned}&(fc_1c_2-us_2)l_{x1}+(fc_2s_1-vs_2)l_{x2}-s_2=0,\\&(fc_1s_2s_3-fc_3s_1+uc_2s_3)l_{y1}+\\&(fc_1c_3+fs_1s_2s_3+vc_2s_3)l_{y2}+c_2s_3=0,\\&(fs_1s_3+fc_1c_3s_2+uc_2c_3)l_{z1}+\\&(fc_3s_1s_2-fc_1s_3+vc_2c_3)l_{z2}+c_2c_3=0,\\&ft_1+ut_3-o_1t_3=0,\\&ft_2+vt_3-o_2t_3=0,\\&fs_1s_3+fc_1c_3s_2+uc_2c_3+ft_1+ut_3-\\&z_1c_2c_3-z_1t_3=0,\\&fc_3s_1s_2-fc_1s_3+vc_2c_3+ft_2+vt_3-\\&z_2c_2c_3-z_2t_3=0.\end{aligned}$$

A.2 Derivations for EM algorithm

Parameter $\tilde{\theta}_G$ is optimized by the EM algorithm. In the M-step, the CPT parameters of discrete random variables, i.e., $R,V,S_l,\boldsymbol{D}_l$, and the conditional linear Gaussian parameters of continuous random variables, i.e., $\boldsymbol{C}_l$, are estimated by the same formulas as derived in [12].

In the E-step, we need to compute the conditional probability of latent variables R and S_l to observed data O_k using the following formulas:

$$\begin{aligned}P(\boldsymbol{O}_k)&=\sum_{R,S_l}P(R,S_l,\boldsymbol{O}_k),\\P(R,S_l|\boldsymbol{O}_k)&=\frac{P(R,S_l,\boldsymbol{O}_k)}{P(\boldsymbol{O}_k)},\end{aligned}$$

where l is category label. Therefore, the key is to compute the joint probability $P(R,S_l,O_k)$:

$$\begin{aligned}&P(R,S_l,O_k)\\=&\sum_{S\backslash\{S_l\}}P(R)P(V)\prod_{l'\in\mathcal{L}}P(S_{l'}|R,V)P(N_{l',k}|R)\\&P(C_{l',k}|S_{l'})P(D_{l',k}|S_{l'})\\=&P(R)P(S_l|R,V)P(N_{l,k}|R)P(C_{l,k}|S_l)P(D_{l,k}|S_l)\\&\prod_{l^*\in\mathcal{L},l^*\neq l}\sum_{S_{l^*}}P(S_{l^*}|R,V)P(N_{l^*,k}|R)\\&P(C_{l^*,k}|S_{l^*})P(D_{l^*,k}|S_{l^*}).\end{aligned}$$