3D Object Manipulation in a Single Photograph using Stock 3D Models

Natasha Kholgade[1] Tomas Simon[1] Alexei Efros[2] Yaser Sheikh[1]

[1]Carnegie Mellon University [2]University of California, Berkeley

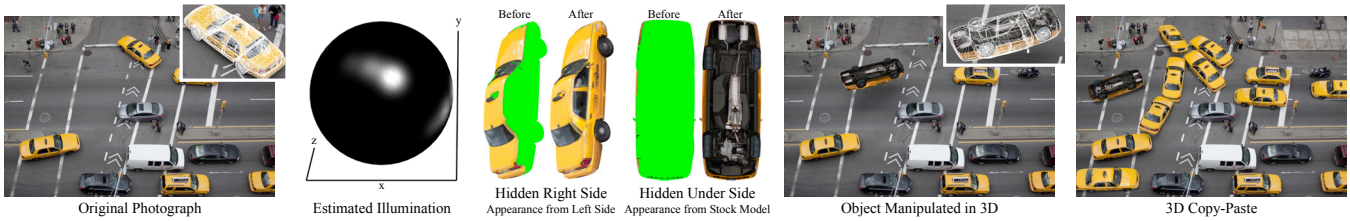


Figure 1: *Using our approach, a user manipulates the taxi cab in a photograph to do a backflip, and copy-pastes the cabs to create a traffic jam (right) by aligning a stock 3D model (inset) obtained from an online repository. Such 3D manipulations often reveal hidden parts of the object. Our approach completes the hidden parts using symmetries and the stock model appearance, while accounting for illumination in 3D.* Photo Credits (leftmost photograph): Flickr user © Lucas Maystre.

Abstract

Photo-editing software restricts the control of objects in a photograph to the 2D image plane. We present a method that enables users to perform the full range of 3D manipulations, including scaling, rotation, translation, and nonrigid deformations, to an object in a photograph. As 3D manipulations often reveal parts of the object that are hidden in the original photograph, our approach uses publicly available 3D models to guide the completion of the geometry and appearance of the revealed areas of the object. The completion process leverages the structure and symmetry in the stock 3D model to factor out the effects of illumination, and to complete the appearance of the object. We demonstrate our system by producing object manipulations that would be impossible in traditional 2D photo-editing programs, such as turning a car over, making a paper-crane flap its wings, or manipulating airplanes in a historical photograph to change its story.

CR Categories: I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual Reality;

Keywords: three-dimensional, photo-editing, 3D models

Links: DL PDF

1 Introduction

One of the central themes of computer graphics is to let the general public move from being passive consumers of visual information (e.g., watching movies or browsing photos) to becoming its active creators and manipulators. One particular area where we have already achieved success is 2D photo-editing software such as Photoshop. Once mainly a tool of professional photographers and designers, it has become mainstream, so much so that 'to photoshop' is now a legitimate English verb [Simpson 2003]. Photoshop lets a user creatively edit the content of a photograph with image operations such as recoloring, cut-and-paste, hole-filling, and filtering. Since the starting point is a real photograph, the final result often appears quite photorealistic as well. However, while photographs are depictions of a three-dimensional world, the allowable geometric operations in photo-editing programs are currently restricted to 2D manipulations in picture space. Three-dimensional manipulations of objects—the sort that we are used to doing naturally in the real world—are simply not possible with photo-editing software; the photograph 'knows' only the pixels of the object's 2D projection, not its actual 3D structure.

Our goal in this paper is to allow users to seamlessly perform *3D manipulation* of objects in a single consumer photograph with the realism and convenience of Photoshop. Instead of simply editing 'what we see' in the photograph, our goal is to manipulate 'what we know' about the scene behind the photograph [Durand 2002]. 3D manipulation of essentially a 2D object sprite is highly underconstrained as it is likely to reveal previously unobserved areas of the object and produce new, scene-dependent shading and shadows. One way to achieve a seamless 'break' from the original photograph is to recreate the scene in 3D in the software's internal representation. However, this operation requires significant effort, that only large special effects companies can afford. It typically also involves external scene data such as light probes, multiple images, and calibration objects, not available with most consumer photographs.

Instead, in this paper, we constrain the recreation of the scene's 3D geometry, illumination, and appearance from the 2D photograph using a publicly available 3D model of the manipulated object as a proxy. Graphics is now entering the age of Big Visual Data: enormous quantities of images and video are uploaded to the Internet daily. With the move towards model standardization and the use of 3D scanning and printing technologies, publicly available 3D data (modeled or scanned using 3D sensors like the Kinect) are also readily available. Public repositories of 3D models (e.g., 3D Warehouse or Turbosquid) are growing rapidly, and several Internet companies are currently in the process of generating 3D models for millions of merchandise items such as toys, shoes, clothing, and household equipment. It is therefore increasingly likely that for most objects in an average user photograph, a stock 3D model will soon be available, if it is not already.

However, it is unreasonable to expect such a model to be a *perfect* match to the depicted object—the visual world is too varied to ever be captured perfectly no matter how large the dataset. Therefore, our approach deals with several types of mismatch between the photographed object and the stock 3D model:

Geometry Mismatch. Interestingly, even among standard, mass-produced household brands (e.g., detergent bottles), there are often subtle geometric variabilities as manufacturers tweak the shape of their products. Of course, for natural objects (e.g., a banana), the geometry of each instance will be slightly different. Even in the cases when a perfect match could be found (e.g., a car of a specific make, model, and year), many 3D models are created with artistic license and their geometry will likely not be metrically accurate, or there are errors due to scanning.

Apperance Mismatch. Although both artists and scanning techniques often provide detailed descriptions of object appearance (surface reflectance), these descriptions may not match the colors and textures (and aging and weathering effects) of the particular instance of the object in the photograph.

Illumination Mismatch. To perform realistic manipulations in 3D, we need to generate plausible lighting effects, such as shadows on an object and on contact surfaces. The environment illumination that generates these effects is not known *a priori*, and the user may not have access to the original scene to take illumination measurements (e.g., in dynamic environments or for legacy photographs).

Our approach uses the pixel information in visible parts of the object to correct the three sources of mismatch. The user semi-automatically aligns the stock 3D model to the photograph using a real-time geometry correction interface that preserves symmetries in the object. Using the aligned model and photograph, our approach automatically estimates environment illumination and appearance information in hidden parts of the object. While a photograph and 3D model may still not contain all the information needed to precisely recreate the scene, our approach sufficiently approximates the illumination, geometry, and appearance of the underlying object and scene to produce *plausible* completion of uncovered areas. Indeed, as shown by the user study in Section 8, our approach plausibly reveals hidden areas of manipulated objects.

The ability to manipulate objects in 3D while maintaining realism greatly expands the repertoire of creative manipulations that can be performed on a photograph. Users are able to quickly perform object-level motions that would be time-consuming or simply impossible in 2D. For example, from just one photograph, users can cause grandma's car to perform a backflip, and fake a baby lifting a heavy sofa. We tie our approach to standard modeling and animation software to animate objects from a single photograph. In this way, we re-imagine typical Photoshop edits—such as object rotation, translation, rescaling, deformation, and copy-paste—as object manipulations in 3D, and enable users to more directly translate what they envision into what they can create.

Contributions. Our key contribution is an approach that allows out-of-plane 3D manipulation of objects in consumer photographs, while providing a seamless break from the original image. To do so, our approach leverages approximate object symmetries and a new non-parametric model of image-based lighting for appearance completion of hidden object parts and for illumination-aware compositing of the manipulated object into the image. We make no assumptions on the structure or nature of the object being manipulated beyond the fact that an approximate stock 3D model is available.

Assumptions. In this paper, we assume a Lambertian model of illumination. We do not model material properties such as refraction, specularities, sub-surface scattering, or inter-reflection. The user study discussed in Section 8 shows that while for some objects, these constraints are necessary to produce plausible 3D manipulations, if their effects are not too pronounced, the results can be perceptually plausible without explicit modeling. In addition, we assume that the user provides a stock 3D model with components for all parts of the objects visible in the original photograph. Finally, we assume that the appearance of the 3D model is self-consistent, i.e., the precise colors of the stock model need not match the photograph, but appearance symmetries should be preserved. For instance, the cliffhanger in Figure 13 is created using the 3D model of a blueish-grey Audi A4 (shown in the supplementary material) to manipulate the green Rover 620 Ti in the photograph.

Notation. For the rest of the paper, we refer to known quantities without using the overline notation, and we use the overline notation for unknown quantities. For instance, the geometry and appearance of the stock 3D model known *a priori* are referred to as $\mathbf{X}$ and $\mathbf{T}$ respectively. The geometry and appearance of the 3D model modified to match the photograph are not known *a priori* and are referred to as $\overline{\mathbf{X}}$ and $\overline{\mathbf{T}}$ respectively. Similarly, the illumination environment which is not known *a priori* is referred to as $\overline{\mathbf{L}}$.

2 Related Work

Modern photo-editing software such as Photoshop provides sophisticated 2D editing operations such as content-aware fill [Barnes et al. 2009] and content-aware photo resizing [Avidan and Shamir 2007]. Many approaches provide 2D edits using low-level assumptions about shape and appearance [Barrett and Cheney 2002; Fang and Hart 2004]. The classic work of Khan et al. [2006] uses insights from human perception to edit material properties of photographed objects, to add transparency, translucency, and gloss, and to change object textures. Goldberg et al. [2012] provide data-driven techniques to add new objects or manipulate existing objects in images in 2D. While these techniques can produce surprisingly realistic results in some cases, their lack of true 3D limits their ability to perform more invasive edits, such as 3D manipulations.

The seminal work of Oh et al. [2001] uses depth-based segmentation to perform viewpoint changes in a photograph. Chen et al. [2011] extend this idea to videos. These methods manipulate visible pixels, and cannot reveal *hidden* parts of objects. To address these limitations, several methods place prior assumptions on photographed objects. Data-driven approaches [Blanz and Vetter 1999] provide drastic view changes by learning deformable models, however, they rely on training data. Debevec et al. [1996] use the regular symmetrical structure of architectural models to reveal novel views of buildings. Kopf et al. [2008] use georeferenced terrain and urban 3D models to relight objects and reveal novel viewpoints in outdoor imagery. Unlike our method, Kopf et al. do not remove the effects of existing illumination. While this works well outdoors, it might not be appropriate in indoor settings where objects cast soft shadows due to area light sources.

Approaches in proxy-based modeling of photographed objects include cuboid proxies [Zheng et al. 2012] and 3-Sweep [Chen et al. 2013]. Unlike our approach, Zheng et al. and Chen et al. (1) cannot reveal hidden areas that are visually distinct from visible areas, limiting the full range of 3D manipulation (e.g., the logo of the laptop from Zheng et al. that we reveal in Figure 7, the underside of the taxi cab in Figure 1, and the face of the wristwatch in Figure 13), (2) cannot represent a wide variety of objects precisely, as cuboids (Zheng et al.) or generalized cylinders (Chen et al.) cannot handle highly deformable objects such as backpacks, clothing, and stuffed toys, intricate or indented objects such as the origami crane in Figure 6 or a pair of scissors, or objects with negative space such as cups, top hats, and shoes, and (3) cannot produce realistic shading and shadows (e.g. in the case of the wristwatch, the top hat, the cliffhanger, the chair, and the fruit in Figure 13, the taxi cab in

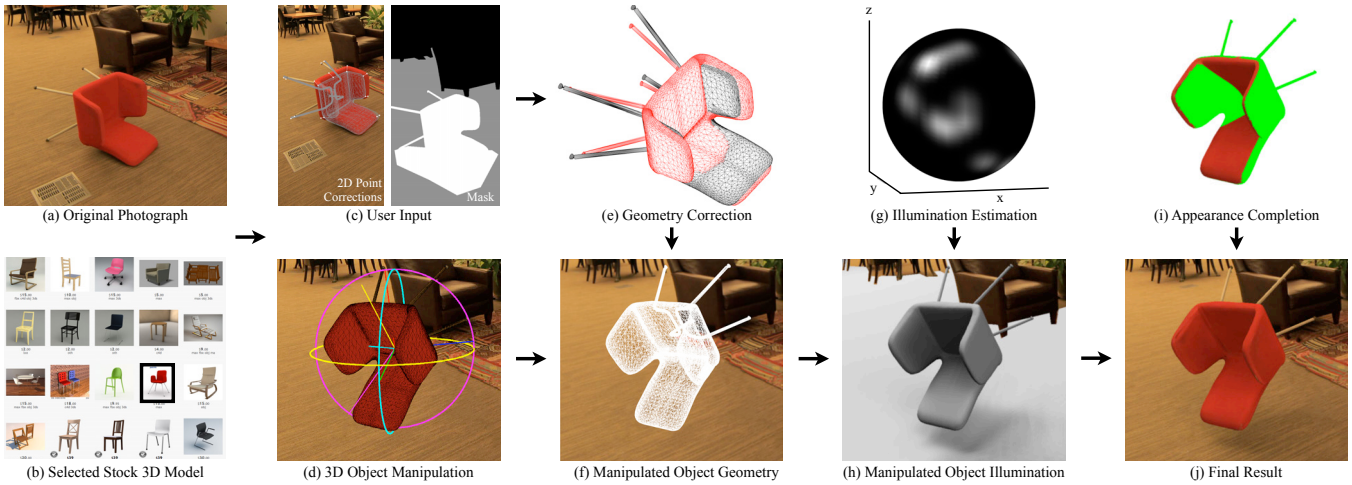


Figure 2: *Overview: (a) Given a photograph and (b) a 3D model from an online repository, the user (c) interactively aligns the model to the photograph and provides a mask for the ground and shadow, which we augment with the object mask and use to fill the background using PatchMatch [Barnes et al. 2009]. (d) The user then performs the desired 3D manipulation. (e) Our approach computes the camera and corrects the 3D geometry, and (f) reveals hidden geometry during the 3D manipulation. (g) It automatically estimates environment illumination and reflectance, (h) to produce shadows and surface illumination during the 3D manipulation. (i) Our approach completes appearance to hidden parts revealed during manipulation, and (j) composites the appearance with the illumination to obtain the final photograph.*

Figure 1, and the crane in Figure 6) to contribute to perceptually plausible 3D manipulation. Zheng et al. use a point light source that does not capture the effect of several area light sources (e.g., in a typical indoor environment). Chen et al. provide no explicit representation of illumination.

An alternative to our proposed method of object manipulation is object insertion, i.e., to inpaint the photographed object and replace it with an inserted object, either in 2D from a large 'photo clip art' library [Lalonde et al. 2007], or in 3D [Debevec 1998; Karsch et al. 2011]. The classic approach of Debevec [1998] renders synthetic 3D objects into the photograph of a real scene by using illumination captured with a mirrored sphere. Karsch et al. [2011] remove the requirement of physical access to the scene by estimating geometry and illumination from the photograph. However, such an insertion-based approach discards useful information about environment illumination and appearance contained in object pixels in the original photograph. In contrast, our approach aims to utilize all information in the original object pixels to estimate the environment illumination and appearance from the input photograph. Furthermore, when creating videos, object insertion methods are unlikely to produce a seamless break from the original photograph, as peculiarities of the particular instance that was photographed (e.g., smudges, defects, or a naturally unique shape) will not exist in a stock 3D model.

Our approach is related to work in the area of geometry alignment, illumination estimation, texture completion, and symmetry detection. In the area of geometry alignment, there exist automated or semi-automated methods [Xu et al. 2011; Prasad et al. 2006; Lim et al. 2013]. In general, as shown by the comparison to Xu et al. [2011] in Section 8, they fail to provide exact alignment crucial for seamless object manipulation. To estimate illumination, we use a basis of von Mises-Fisher kernels that provide the advantage of representing high frequency illumination effects over the classical spherical harmonics representation used in Ramamoorthi and Hanrahan [2001], while avoiding unnaturally sharp cast shadows that arise due to the point light representation used in Mei et al. [2009]. In addition, we impose non-negativity constraints on the basis coefficients that ensure non-negativity of illumination, in contrast to the Haar wavelet basis used in Ng et al. [2003], Okabe et al. [2004], Haber et al. [2009] and Romeiro et al. [2010]. Mixtures of von Mises-Fisher kernels have been estimated for single view relighting [Hara et al. 2008; Panagopoulos et al. 2009], however, these require estimating the number of mixtures separately. Our appearance completion approach is related to methods that texture map 3D models using images [Kraevoy et al. 2003; Tzur and Tal 2009; Gal et al. 2010], however, they do not factor out illumination, and may use multiple images to obtain complete appearance. In using symmetries to complete appearance, our work is related to approaches that extract symmetries from images and 3D models [Hong et al. 2004; Gal and Cohen-Or 2006; Pauly et al. 2005], and that use symmetries to complete geometry [Terzopoulos et al. 1987; Mitra et al. 2006; Mitra and Pauly 2008; Bokeloh et al. 2011], and to infer missing appearance [Kim et al. 2012]. However, our work differs from these approaches in that the approaches are mutually exclusive: approaches focused on symmetries from geometry do not respect appearance constraints, and vice versa. Our approach uses an intersection of geometric symmetry and appearance similarity, and prevents appearance completion between geometrically similar but visually distinct parts, such as the planar underside and top of a taxi-cab, or between visually similar but geometrically distinct parts such as the curved surface of a top-hat and its flat brim.

3 Overview

The user manipulates an object in a photograph as shown in Figure 2(a) by using a stock 3D model. For this photograph, the model was obtained through a word search on the online repository TurboSquid. Other repositories such as 3D Warehouse, (Figure 2(b)) or semi-automated approaches such as those of Xu et al. [2011], Lim et al. [2013], and Aubry et al. [2014] may also be used. The user provides a mask image that labels the ground and shadow pixels. We compute a mask for the object pixels, and use this mask to inpaint the background using the PatchMatch algorithm [Barnes et al. 2009]. For complex backgrounds, the user may touch up the background image after inpainting. Figure 2(c) shows the mask with ground pixels in gray, and object and shadow pixels in white. The user semi-automatically aligns and corrects the stock 3D model to match the photograph using our symmetry-preserving geometry correction interface as shown in Figure 2(c). Using the corrected

3D model and the photograph pixels, our approach computes and factors out the environment illumination (Figure 2(g)), and completes the appearance of hidden areas (Figure 2(i)). Users can then perform their desired 3D manipulations as shown in Figure 2(d), and the illumination, completed appearance, and texture are composited to produce the final output, shown in Figure 2(j).

When a user manipulates an object in the photograph $\mathbf{I} \in \mathbb{R}^{W\times H\times 3}$, shown in Figure 2(a), from the object's original pose $\mathbf{\Theta}$ to a new pose $\mathbf{\Omega}$ as in Figure 2(d), our objective is to produce an edited photograph $\mathbf{J}$ shown in Figure 2(j). Here W and H are the width and height of the photograph. We model $\mathbf{I}$ as a function of the object geometry $\overline{\mathbf{X}}$, object appearance $\overline{\mathbf{T}}$, and the environment illumination $\overline{\mathbf{L}}$, as

$$\mathbf{I} = f(\mathbf{\Theta}, \overline{\mathbf{X}}, \overline{\mathbf{T}}, \overline{\mathbf{L}}). \tag{1}$$

The above equation essentially represents the rendering equation. The manipulated photograph $\mathbf{J}$ can then be produced by replacing the original pose $\mathbf{\Theta}$ with the new pose $\mathbf{\Omega}$, i.e., $\mathbf{J} = f(\mathbf{\Omega}, \overline{\mathbf{X}}, \overline{\mathbf{T}}, \overline{\mathbf{L}})$. However, $\overline{\mathbf{X}}$, $\overline{\mathbf{T}}$, and $\overline{\mathbf{L}}$ are not known *a priori*, and estimating them from a single photograph without any prior assumptions is highly ill-posed [Barron 2012].

We overcome this difficulty by bootstrapping the estimation using the stock 3D model of the object, whose geometry consists of vertices $\mathbf{X}$, and whose appearance is a texture map $\mathbf{T}$ (Figure 2(b)). In general, the stock model geometry and appearance do not precisely match the geometry $\overline{\mathbf{X}}$ and appearance $\overline{\mathbf{T}}$ of the photographed object. We provide a tool through which the user marks 2D point corrections, shown in Figure 2(c). We deform $\mathbf{X}$ to match $\overline{\mathbf{X}}$ using the 2D corrections, as shown in Figure 2(e) and described in Section 4. Additionally, we estimate the ground plane in 3D, using one of two methods: either using vanishing points from user-marked parallel lines in the image, or as the plane intersecting three user-marked points on the base of the object. Ground plane estimation is described in the supplementary material. Manually correcting the illumination and the appearance is difficult, as the illumination sources may not be visible in the photograph. Instead, we present an algorithm to estimate the illumination and appearance using pixels on the object and the ground, as shown in Figure 2(g) and described in Section 5, by optimizing the following objective:

$$\left\{\overline{\mathbf{T}}^{\star}, \overline{\mathbf{L}}^{\star}\right\} = \underset{\overline{\mathbf{T}},\overline{\mathbf{L}}}{\arg\min} \left\|\mathbf{I} - f(\mathbf{\Theta}, \overline{\mathbf{X}}, \overline{\mathbf{T}}, \overline{\mathbf{L}})\right\|_2^2. \tag{2}$$

Equation (2) only estimates the appearance for parts of the object that are visible in the original photograph $\mathbf{I}$ as shown in Figure 2(i). The new pose $\mathbf{\Theta}'$ potentially reveals hidden parts of the object. To produce the manipulated photograph $\mathbf{J}$, we need to complete the hidden appearance. After factoring out the effect of illumination on the appearance in visible areas, we present an algorithm that uses symmetries to complete the appearance of hidden parts from visible areas as described in Section 6. The algorithm uses the stock model appearance for hidden parts of objects that are not symmetric to visible parts. Given the estimated geometry, appearance, and illumination, and the user-manipulated pose of the object, we composite the edited photograph by replacing $\mathbf{\Theta}$ with $\mathbf{\Theta}'$ in Equation (1) as shown in Figures 2(f), 2(h), and 2(j).

4 Geometry Correction

We provide a user-guided approach to correct the geometry of the 3D model to match the photographed object, while ensuring that smoothness and symmetry are preserved over the model. Given the photograph $\mathbf{I}$ and the stock model geometry $\mathbf{X} \in \mathbb{R}^{N\times 3}$ (where N is the number of vertices), we first estimate the original rigid pose $\mathbf{\Theta}$ of the object using a set $\mathcal{A}$ of user-defined 3D-2D correspondences, $\mathbf{X}_j \in \mathbb{R}^3$ on the model and $\overline{\mathbf{x}}_j \in \mathbb{R}^2, j \in \mathcal{A}$ in the photograph. Here, $\mathbf{\Theta} = \{\mathbf{R}, \mathbf{t}\}$, where $\mathbf{R} \in \mathbb{R}^{3\times 3}$ is the object rotation, and $\mathbf{t} \in \mathbb{R}^3$ is the object translation. We use the EPnP algorithm [Lepetit et al. 2009] to estimate $\mathbf{R}$ and $\mathbf{t}$. The algorithm takes as input $\mathbf{X}_j$, $\mathbf{x}_j$, and the matrix $\mathbf{K} \in \mathbb{R}^{3\times 3}$ of camera parameters (i.e., focal length, skew, and pixel aspect ratio). We assume a zero-skew camera, with square pixels and principal point at the photograph center. We use the focal length computed from EXIF tags when available, else we use vanishing points to compute the focal length. We assume that objects in the photograph are at rest on a ground plane. We describe focal length extraction using vanishing points, and ground plane estimation in the supplementary material. It should be noted that there exists a scale ambiguity in computing $\mathbf{t}$. The EPnP algorithm handles the scale ambiguity in terms of translation along the z-axis of the camera.

As shown in Figure 3(a), after the camera is estimated, the user provides a set $\mathcal{B}$ of start points $\mathbf{x}_k \in \mathbb{R}^2, k \in \mathcal{B}$ on the projection of the stock model, and a corresponding set of end points $\overline{\mathbf{x}}_k \in \mathbb{R}^2$ on the photographed object for the purpose of geometry correction. We used a point-to-point correction approach, as opposed to sketch or contour-based approaches [Nealen et al. 2005; Kraevoy et al. 2009], as reliably tracing soft edges can be challenging compared to providing point correspondences. The user only provides the point corrections in 2D. We use them to correct $\mathbf{X}$ to $\overline{\mathbf{X}}$ in 3D by optimizing an objective in $\overline{\mathbf{X}}$ consisting of a correction term E_1, a symmetry prior E_2, and a smoothness prior E_3:

$$E(\overline{\mathbf{X}}) = E_1(\overline{\mathbf{X}}) + E_2(\overline{\mathbf{X}}) + E_3(\overline{\mathbf{X}}). \tag{3}$$

The correction term E_1 forces projections of the points $\overline{\mathbf{X}}_k^{\Theta} = \mathbf{R}\overline{\mathbf{X}}_k + \mathbf{t}$ to match the user-provided 2D corrections $\overline{\mathbf{x}}_k, k \in \mathcal{B}$. Here, $\overline{\mathbf{X}}^{\Theta}$ represents $\overline{\mathbf{X}}$ transformed by $\mathbf{\Theta}$. As shown in Figure 3(b), we compute the ray $\mathbf{v}_k = \mathbf{K}^{-1}[\overline{\mathbf{x}}_k^T \; 1]^T$ back-projected through each $\overline{\mathbf{x}}_k$. E_1 minimizes the sum of distances between each $\overline{\mathbf{X}}_k^{\Theta}$ and the projection $\frac{\mathbf{v}_k\mathbf{v}_k^T}{\|\mathbf{v}_k\|^2}\overline{\mathbf{X}}_k^{\Theta}$ of $\overline{\mathbf{X}}_k^{\Theta}$ onto the ray $\mathbf{v}_k$:

$$E_1(\overline{\mathbf{X}}) = \sum_{k\in\mathcal{B}} \left\| \overline{\mathbf{X}}_k^{\Theta} - \frac{\mathbf{v}_k\mathbf{v}_k^T}{\|\mathbf{v}_k\|_2^2}\overline{\mathbf{X}}_k^{\Theta} \right\|_2^2. \tag{4}$$

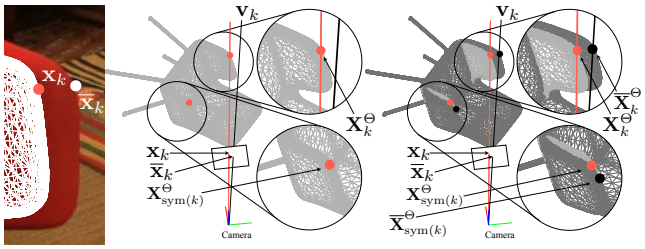
Unlike traditional rotoscoping, the correction term encourages the vertex coordinates to match the photograph only after geometric projection into the camera. The corrected vertices $\overline{\mathbf{X}}_k$ are otherwise free to move along the lines of projection such that the overall deformation energy $E(\overline{\mathbf{X}})$ is minimized.

The smoothness prior E_2 preserves local smoothness over the corrected model. As shown in Figure 3(c), the term ensures that points in the neigborhood of the corrected points $\overline{\mathbf{X}}_k$ move smoothly. In our work, this term refers to the surface deformation energy from the as-rigid-as-possible framework of Sorkine and Alexa [2007]. The framework requires that local deformations within the 1-ring neighborhood $\mathcal{D}_i$ of the i^{th} point in the corrected model should have nearly the same rotations $\overline{\mathbf{R}}_i$ as on the original model:

$$E_2(\overline{\mathbf{X}}) = \sum_{i=1}^{N} \sum_{j\in\mathcal{D}_i} \left\|(\overline{\mathbf{X}}_i - \overline{\mathbf{X}}_j) - \overline{\mathbf{R}}_i(\mathbf{X}_i - \mathbf{X}_j)\right\|_2^2. \tag{5}$$

The local rotation $\overline{\mathbf{R}}_i$ in the neighborhood of a vertex $\overline{\mathbf{X}}_i$ is distinct from the global rigid rotation $\mathbf{R}$.

The symmetry prior E_3 preserves the principal symmetry (or bilateral symmetry) of the model, as shown in Figure 3(c). If a point



(a) User 2D Edit (b) Correction Along Ray in 3D (c) Smoothness and Symmetry Priors

Figure 3: *Geometry correction. (a) The user makes a 2D correction by marking a start-end pair, $(\mathbf{x}_k, \overline{\mathbf{x}}_k)$ in the photograph. (b) Correction term: The back-projected ray $\mathbf{v}_k$ corresponding to $\overline{\mathbf{x}}_k$ is shown in black, and the back-projected ray corresponding to $\mathbf{x}_k$ is shown in red. The top inset shows the 3D point $\mathbf{X}_k^{\boldsymbol{\Theta}}$ for $\mathbf{x}_k$ on the stock model, and the bottom inset shows its symmetric pair $\mathbf{X}_{sym(k)}^{\boldsymbol{\Theta}}$. We deform the stock model geometry (light grey) to the user-specified correction (dark grey) subject to smoothness and symmetry-preserving priors.*

on the stock model $\mathbf{X}_i$ has a symmetric counterpart $\mathbf{X}_{\text{sym}(i)}$, E_3 ensures that $\overline{\mathbf{X}}_i$ remains symmetric to $\overline{\mathbf{X}}_{\text{sym}(i)}$, i.e., that they are related through a symmetric transform,

$$\overline{\mathbf{S}} = \begin{bmatrix} \mathbf{I}_3 - 2\overline{\mathbf{n}}\overline{\mathbf{n}}^T & 2\overline{\mathbf{n}}\overline{d} \end{bmatrix}, \tag{6}$$

where $\mathbf{I}_3$ is the 3×3 identity matrix, and $\overline{\mathbf{n}}$ and $\overline{d}$ are the normal and distance of the principal plane of symmetry in the corrected model geometry $\overline{\mathbf{X}}$. E_3 is thus given as

$$E_3(\overline{\mathbf{X}}) = w \sum_{i=1}^{N} \left\| \overline{\mathbf{S}} [\overline{\mathbf{X}}_i^T \ 1]^T - \overline{\mathbf{X}}_{\text{sym}(i)} \right\|_2^2. \tag{7}$$

Here, w is a user-defined weight, that the user sets to 1 if the object has bilateral symmetry, and 0 otherwise.

To determine $\mathbf{X}_{\text{sym}(i)}$ for every stock model point $\mathbf{X}_i$, we compute the principal plane $\pi = [\mathbf{n}^T \ -d]^T$ on the stock model using RANSAC[1]. We then reflect every $\mathbf{X}_i$ across π, and obtain $\mathbf{X}_{\text{sym}(i)}$ as the nearest neighbor to the reflection of $\mathbf{X}_i$ across π.

The objective function in Equation (3) is non-convex. However, note that given the symmetry $\overline{\mathbf{S}}$ and local rotations $\overline{\mathbf{R}}_i$, the objective is convex in the geometry $\overline{\mathbf{X}}$, and vice versa. We initialize $\overline{\mathbf{R}}_i = \mathbf{I}_3$, where $\mathbf{I}_3$ is the 3×3 identity matrix, and $\overline{\mathbf{S}}$ with the original stock model symmetry $\mathbf{S}$. We alternately solve for the geometry, and the symmetry and local rotations till convergence to a local minimum. Given $\overline{\mathbf{S}}$ and $\overline{\mathbf{R}}_i$, we solve for $\overline{\mathbf{X}}$ by setting up a system of linear equations. Given $\overline{\mathbf{X}}$, we solve for $\overline{\mathbf{R}}_i$ through SVD as described by Sorkine and Alexa [2007]. To solve for $\overline{\mathbf{S}}$, we assume that the bilateral plane of symmetry passes through the center of mass of the object (which we can assume without loss of generality to be at the origin), so that $\overline{d} = 0$. To obtain $\overline{\mathbf{n}}$, we note that the first three columns of $\overline{\mathbf{S}}$ (which we refer to as $\overline{\mathbf{S}}_3$) form an orthogonal matrix, which we extract using SVD, as follows: We create matrices $\mathbf{A} = [\overline{\mathbf{X}}_1, \cdots, \overline{\mathbf{X}}_N]$ and $\mathbf{B} = [\overline{\mathbf{X}}_{\text{sym}(1)} \cdots \overline{\mathbf{X}}_{\text{sym}(N)}]$, perform the SVD decomposition of $\mathbf{AB}^T$ as $\mathbf{U\Sigma V}^T$, and extract $\overline{\mathbf{S}}_3 = \mathbf{VU}^T$. Then, we extract $\mathbf{n}$ as the principal eigenvector of the matrix $\frac{\mathbf{I}_3 - \overline{\mathbf{S}}_3}{2}$. We substitute $\overline{\mathbf{n}}$ and $\overline{d} = 0$ in Equation (6) to get $\overline{\mathbf{S}}$.

[1] At each RANSAC iteration, we randomly choose two points $\mathbf{X}_{i_r}$ and $\mathbf{X}_{j_r}$ on the stock 3D model, and compute their bisector plane π_r with normal $\mathbf{n}_r = \frac{\mathbf{x}_{i_r} - \mathbf{x}_{j_r}}{\|\mathbf{x}_{i_r} - \mathbf{x}_{j_r}\|^2}$, and distance from origin $\frac{1}{2}\mathbf{n}_r^T(\mathbf{X}_{i_r} + \mathbf{X}_{j_r})$. We maintain a score n_r of the number of points that when reflected across π_r have a symmetric neighbor within a small threshold μ. After R iterations, we retain the plane with maximum score as π.

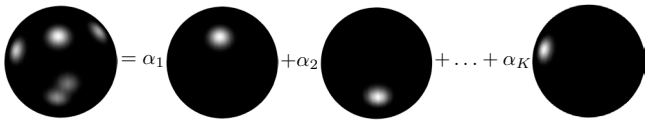


Figure 4: *We represent the environment map as a linear combination of the von Mises-Fisher (vMF) basis. We enforce constraints of sparseness and grouping of basis coefficients to mimic area lighting and produce soft cast shadows.*

5 Illumination and Appearance Estimation

Given the corrected geometry $\overline{\mathbf{X}}^\Theta$, we estimate illumination $\overline{\mathbf{L}}$ and appearance $\overline{\mathbf{T}}$ to produce plausible shadows and lighting effects on the object and the ground. We represent the imaging function f using a Lambertian reflection model. Under this model, the i^{th} pixel $\mathbf{I}_i$ on the object and the ground in the photograph is generated as

$$f_i(\boldsymbol{\Theta}, \overline{\mathbf{X}}, \overline{\mathbf{T}}, \overline{\mathbf{L}}) = \overline{\mathbf{P}}_i \int_\Omega \mathbf{n}_i \cdot \mathbf{s}_i(\omega) v_i(\omega) \overline{\mathbf{L}}(\omega) d\omega + \overline{\boldsymbol{\delta}}_i, \tag{8}$$

where $\overline{\mathbf{P}}_i \in \mathbb{R}^3$ and $\overline{\boldsymbol{\delta}}_i \in \mathbb{R}^3$ model the appearance of the i^{th} pixel. We assume that the appearance $\overline{\mathbf{T}}$ of the object consists of a reflectance map $\overline{\mathbf{P}} \in \mathbb{R}^{U\times V\times 3}$and a residual difference map $\overline{\boldsymbol{\delta}} \in \mathbb{R}^{U\times V\times 3}$. Here, $\overline{\mathbf{P}}$ models the diffuse reflectance of the object's appearance (also termed the albedo). Inspired by Debevec [1998]), we include $\overline{\boldsymbol{\delta}}$ to represent the residual difference between the image pixels and the diffuse reflection model, since the model is only an approximation to the BRDF of the object. U and V are the dimensions of the texture map. For the i^{th} pixel, $\overline{\mathbf{P}}_i$ and $\overline{\boldsymbol{\delta}}_i$ are interpolated from the maps $\overline{\mathbf{P}}$ and $\overline{\boldsymbol{\delta}}$ at the point $\overline{\mathbf{X}}_i^\Theta$. $\overline{\mathbf{X}}_i^\Theta$ is the 3D point back-projected from the i^{th} pixel location to the object's 3D geometry $\overline{\mathbf{X}}$ as transformed by $\boldsymbol{\Theta}$. $\mathbf{n}_i$ is the normal at point $\overline{\mathbf{X}}_i^\Theta$, $\mathbf{s}_i(\omega)$ is the source vector from $\overline{\mathbf{X}}_i^\Theta$ towards the light source along the solid angle ω, and $v_i(\omega)$ is the visibility of this light source from $\overline{\mathbf{X}}_i^\Theta$. $\overline{\mathbf{L}}(\omega)$ is the intensity of the light source along ω. We assume that the light sources lie on a sphere, i.e., that $\overline{\mathbf{L}}(\omega)$ is a spherical environment map.

To estimate these quantities, we optimize the following objective function in $\overline{\mathbf{P}}$ and $\overline{\mathbf{L}}$, consisting of a data term F_1, an illumination prior F_2, and a reflectance prior F_3:

$$F(\overline{\mathbf{P}}, \overline{\mathbf{L}}) = F_1(\overline{\mathbf{P}}, \overline{\mathbf{L}}) + F_2(\overline{\mathbf{L}}) + F_3(\overline{\mathbf{P}}), \tag{9}$$

and we obtain $\overline{\boldsymbol{\delta}}$ as the residual of the data term F_1 in the objective function. F_1 represents the generation of pixels in a single photograph using the illumination model from Equation 8 as follows:

$$F_1(\overline{\mathbf{P}}, \overline{\mathbf{L}}) = \sum_{i=1}^{N_\mathbf{I}} \tau_i \left\| \mathbf{I}_i - \overline{\mathbf{P}}_i \int_\Omega \mathbf{n}_i \cdot \mathbf{s}_i(\omega) v_i(\omega) \overline{\mathbf{L}}(\omega) d\omega \right\|_2^2,$$

$$\text{where } \tau_i = \begin{cases} 1 & \text{for object pixels} \\ & \text{and shadow pixels on ground,} \\ \tau & \text{for non-shadow pixels on ground.} \end{cases}$$

Here, $N_\mathbf{I}$ is the number of pixels covered by the object and the ground in the photograph. τ corresponds to the value in the gray

region of the user-provided mask shown in Figure 2(c). Here, $0 \leq \tau \leq 1$, and a small value of τ in non-shadow areas of the ground emphasizes cast shadows. F_2 and F_3 represent illumination and reflectance priors that regularize the ill-posed optimization of estimating $\overline{\mathbf{P}}$ and $\overline{\mathbf{L}}$ from a single photograph.

To describe the illumination prior F_2, we represent $\overline{\mathbf{L}}(\omega)$ as a linear combination of von Mises-Fisher (vMF) kernels, $\overline{\mathbf{L}}(\omega) = \mathbf{\Gamma}(\omega)\overline{\boldsymbol{\alpha}}$. $\mathbf{\Gamma} \in \mathbb{R}^K$ is a functional basis, shown in Figure 4, and $\overline{\boldsymbol{\alpha}}$ is a vector of basis coefficients. The k-th component of $\mathbf{\Gamma}$ corresponds to the k-th vMF kernel, given by

$$h\left(\mathbf{u}(\omega); \boldsymbol{\mu}_k, \kappa\right) = \frac{\exp\left(\kappa {\boldsymbol{\mu}_k}^T \mathbf{u}(\omega)\right)}{4\pi \sinh \kappa},$$

where $\boldsymbol{\mu}_k$ is the k-th mean direction vector, $\mathbf{u}(\omega)$ is a unit vector along direction ω, and the concentration parameter κ describes the peakiness of the distribution [Fisher 1953].

Through the illumination prior F_2, we force the algorithm to find a sparse set of light sources using an L_1 prior on the coefficients. In addition, according to the elastic net framework [Zou and Hastie 2005], we place an L_2 prior to force groups of correlated coefficients to be turned on. The L_2 prior forces spatially adjacent light sources to be switched on simultaneously to represent illumination sources such as area lights or windows. We thus obtain the following form for F_2:

$$F_2(\overline{\mathbf{L}}) = \lambda_1 \left\|\overline{\boldsymbol{\alpha}}\right\|_1 + \lambda_2 \left\|\overline{\boldsymbol{\alpha}}\right\|_2^2. \tag{10}$$

The reflectance prior F_3 enforces piecewise constancy over the deviation of the reflectance $\overline{\mathbf{P}}$ from the original stock model reflectance $\mathbf{P}$:

$$F_3(\overline{\mathbf{P}}) = \lambda_3 \sum_{i=1}^{N_\mathbf{I}} \sum_{j \in \mathcal{N}_i} \left\|(\overline{\mathbf{P}}_i - \mathbf{P}_i) - (\overline{\mathbf{P}}_j - \mathbf{P}_j)\right\|_1. \tag{11}$$

$\mathbf{P}$ belongs to the stock model appearance $\mathbf{T}$ described in Section 3. The prior F_3 is related to color constancy assumptions about intrinsic images [Land et al. 1971; Karsch et al. 2011]. $\mathcal{N}_i$ represents the 4-neighborhood of the i^{th} pixel in image space.

We optimize the objective F in Equation (9) subject to non-negativity constraints on $\overline{\boldsymbol{\alpha}}$:

$$\left\{\overline{\mathbf{P}}^\star, \overline{\mathbf{L}}^\star\right\} = \underset{\overline{\mathbf{P}}, \overline{\mathbf{L}}}{\arg\min}\ F(\overline{\mathbf{P}}, \overline{\mathbf{L}}),\ \text{s.t.}\ \overline{\boldsymbol{\alpha}} \geq 0. \tag{12}$$

The above optimization is non-convex due to the bilinear interaction of the surface reflectances $\overline{\mathbf{P}}$ with the illumination $\overline{\mathbf{L}}$. If we know the reflectances, we can solve a convex optimization for the illumination, and vice versa. We initialize the reflectances with the stock model reflectance $\mathbf{P}$ for the object, and the median pixel value for the ground plane. We alternately solve for illumination and reflectance until convergence to a local minimum. To represent the vMF kernels and $\overline{\mathbf{L}}$, we discretize the sphere into K directions, and compute K kernels, one per direction. Finally, we compute the appearance difference as the residual of synthesizing the photograph using the diffuse reflection model, i.e.,

$$\overline{\boldsymbol{\delta}}_i^\star = \mathbf{I}_i - \overline{\mathbf{P}}_i^\star \int_\Omega \mathbf{n}_i \cdot \mathbf{s}_i(\omega) v_i(\omega) \overline{\mathbf{L}}^\star(\omega) d\omega. \tag{13}$$

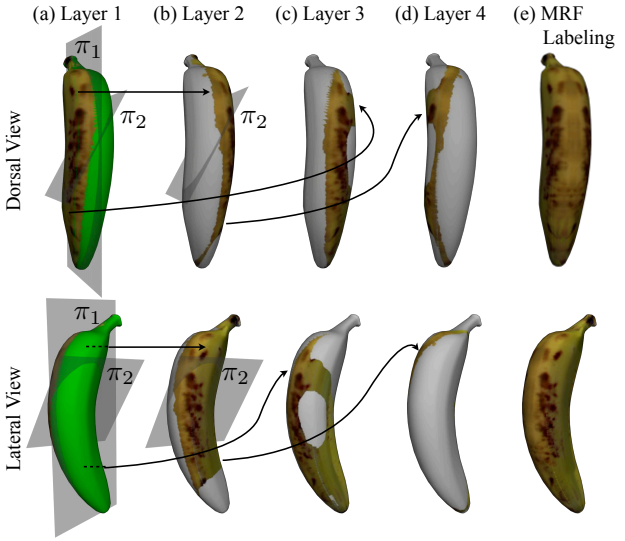


Figure 5: *We build an MRF over the object model to complete appearance. (a) Due to the camera viewpoint, the vertices are partitioned into a visible set $\mathcal{I}_v$ shown with the visible appearance, and a hidden set $\mathcal{I}_h$ shown in green. Initially, the graph has a single layer of appearance candidates, labeled Layer 1, corresponding to visible parts. At the first iteration, we use the bilateral plane of symmetry π_1 to transfer appearance candidates from Layer 1 to Layer 2. At the second iteration, we use an alternate plane of symmetry π_2 to transfer appearance candidates (c) from Layer 1 to Layer 3, and (d) from Layer 2 to Layer 4. We perform inference over an MRF to find the best assignment of appearance candidates from several layers to each vertex. This result was obtained after six iterations.*

6 Appearance Completion to Hidden Parts

The appearance $\overline{\mathbf{T}} = \{\overline{\mathbf{P}}^\star, \overline{\boldsymbol{\delta}}^\star\}$ computed in Section 5 is only available for visible parts of the object, as shown in Figure 5(a). We use multiple planes of symmetry to complete the appearance in hidden parts of the object using the visible parts. We first establish symmetric relationships between hidden and visible parts of the object via planes of symmetry. These symmetric relationships are used to suggest multiple appearance candidates for vertices on the object. The appearance candidates form the labels of a Markov Random Field (MRF) over the vertices. To create the MRF, we first obtain a fine mesh of object vertices $\mathbf{X}_s, s \in \mathcal{I}$ created by mapping the uv-locations on the texture map onto the 3D object geometry. Here $\mathcal{I}$ is a set of indices for all texel locations. We use this fine mesh since the original 3D model mesh usually does not provide one vertex per texel location, and cannot be directly used to completely fill the appearance. We set up the MRF as a graph whose vertices correspond to $\mathbf{X}_s$, and whose edges consist of links from each $\mathbf{X}_s$ to the set $\mathcal{K}_s$ consisting of k nearest neighbors of $\mathbf{X}_s$. As described in Section 6.1, we associate each vertex $\mathbf{X}_s$ with a set of L appearance candidates $(\tilde{\mathbf{P}}_{i_s}, \tilde{\boldsymbol{\delta}}_{i_s}), i \in \{1, 2, \cdots, L\}$ through multiple symmetries. Appearance candidates with the same value of i form a layer, and the algorithm in Section 6.1 grows layers (Figure 5(b) through 5(d)) by transferring appearance candidates from previous layers across planes of symmetry. To obtain the completed appearance, shown in Figure 5(e), we find an assignment of appearance candidates, such that each vertex is assigned one candidate and the assignment satisfies the constraints of neighborhood smoothness, consistency of texture, and matching of visible appearance to the observed pixels. We obtain this assignment, by performing inference over the MRF as described in Section 6.2.

6.1 Relating Hidden & Visible Parts via Symmetries

We establish symmetric relationships between hidden and visible parts of the object by leveraging the regularities of the stock model $\mathbf{X}$. However, we identify the hidden and visible areas using the aligned and corrected model $\overline{\mathbf{X}}^{\Theta}$. To do so, we note that a point $\mathbf{X}_s$ on the stock model is related to the texture map and to a corresponding point $\overline{\mathbf{X}}_s^{\Theta}$ on the corrected model through barycentric coordinates over a triangle mesh specified on the stock model. We compute $\overline{\mathbf{X}}_s^{\Theta}$ on the corrected model using the barycentric coordinates of $\mathbf{X}_s$. We then use pose $\mathbf{\Theta}$ of the object to determine the set of indices $\mathcal{I}_v$ for vertices visible from the camera viewpoint (shown as textured parts of the banana in Figure 5(a)), and the set of indices $\mathcal{I}_h = \mathcal{I} \setminus \mathcal{I}_v$ for vertices hidden from the camera viewpoint (shown in green in Figure 5(a)). While it is possible to pre-compute symmetry planes on an object, our objective is to relate visible parts of an object to hidden parts through symmetries, many of which turn out to be approximate (for instance, different parts of a banana with approximately similar curvature are identified as symmetries using our approach). Pre-computing all such possible symmetries is computationally prohibitive. We proceed iteratively, and in each iteration, we compute a symmetric relationship between $\mathcal{I}_h$ and $\mathcal{I}_v$ using the stock model $\mathbf{X}$. Specifically, we compute planes of symmetry, shown as planes π_1 and π_2 in Figures 5(a) and 5(b). Through this symmetric relationship, we associate appearance candidates $(\tilde{\mathbf{P}}_{i_s}, \tilde{\boldsymbol{\delta}}_{i_s})$ to each vertex $\mathbf{X}_s$ in the graph by growing out layers of appearance candidates.

Algorithm 1 details the use of symmetries to associate appearance candidates $\tilde{\mathbf{P}}_{i_s}$ and $\tilde{\boldsymbol{\delta}}_{i_s}$ through layers. The algorithm initializes the appearance candidates at the first layer for vertices of the object in parts visible in the photograph using the appearance $\overline{\mathbf{P}}$ and $\overline{\boldsymbol{\delta}}$ computed in Section 5. Over M iterations, the algorithm uses M planes of symmetry to populate the graph from layers 1 to $L = 2^M$. The algorithm uses RANSAC to compute the optimal plane of symmetry π_m at the m^{th} iteration. Planes π_1 and π_2 computed for iterations $m = 1$ and $m = 2$ are shown in Figures 5(a) and 5(b). At the m^{th} iteration, the algorithm then generates 2^{m-1} new layers, by transferring appearance candidates from the first 2^{m-1} layers across the plane of symmetry. In Figure 5(b), Layer 2 is generated by transferring appearance candidates from Layer 1 across π_1. Using plane π_2 obtained in iteration $m = 2$, the algorithm generates Layers 3 and 4 (in Figures 5(c) and 5(d)) from Layers 1 and 2. The first 2^{m-1} layers are grown in the previous $m - 1$ iterations. The algorithm uses the criteria of geometric symmetry (captured by the distance $\left\|\mathbf{X}_s - 2\mathbf{n}_m\mathbf{n}_m^T\mathbf{X}_{t^\star} + 2\mathbf{n}_m d_m\right\|$ in vertex space) and appearance similarity (represented by the distance $\|\mathbf{P}_s - \mathbf{P}_{t^\star}\|$ in the image space of the texture map) to compute the symmetry plane π_m and to transfer appearance candidates. For vertices for which no appearance candidates can be generated using geometric symmetry and appearance similarity, as in the case of the underside of the taxi cab in Figure 1, the algorithm defaults to the stock model appearance $\mathbf{P}$ (for which appearance difference $\boldsymbol{\delta}$ is 0).

6.2 Completing Appearance via MRF

To obtain the completed appearance for the entire object from the appearance candidates, shown in Figure 5(e), we need to select a set of candidates such that (1) each vertex on the 3D model is assigned exactly one candidate, (2) the selected candidates satisfy smoothness and consistency constraints, and (3) visible vertices retain their original appearance. To do this, we perform inference over the MRF using tree-reweighted message passing (TRW-S) [Kolmogorov 2006]. While graph-based inference has been used to complete texture in images [Kwatra et al. 2003], our approach uses the layers obtained by geometric and appearance-based symmetries from Section 6.1.

Algorithm 1 Associating Appearance Candidates Through Layers.

Set user-defined values for μ, ν, and M.
$\forall i \in \{1, 2, \cdots, 2^M\}$ & $\forall s \in \mathcal{I}$, set $\tilde{\mathbf{P}}_{i_s} \leftarrow \infty$ & $\tilde{\boldsymbol{\delta}}_{i_s} \leftarrow \infty$.
Initialize $\mathcal{I}_c \leftarrow \mathcal{I}_v$, and $\mathcal{I}_l \leftarrow \mathcal{I}_h$.
$\forall s \in \mathcal{I}_v$, set $\tilde{\mathbf{P}}_{1_s} \leftarrow \infty$ and $\tilde{\boldsymbol{\delta}}_{1_s} \leftarrow \infty$.
for $m = 1$ to M **do**
  Initialize $n_m \leftarrow 0$, $\mathcal{I}_m \leftarrow \emptyset$, $\mathbf{n}_m \leftarrow \begin{bmatrix}0\ 0\ 0\end{bmatrix}^T$, and $d_m \leftarrow 0$.
  RANSAC for optimal plane of symmetry π_m:
  for $r = 1$ to R **do**
    Randomly select $s_r \in \mathcal{I}$ and $t_r \in \mathcal{I}_l$.
    Compute bisector plane $\pi_r = \begin{bmatrix}\mathbf{n}_r^T & -d_r\end{bmatrix}^T$, where
    $\mathbf{n}_r \leftarrow \frac{\mathbf{X}_{s_r} - \mathbf{X}_{t_r}}{\|\mathbf{X}_{s_r} - \mathbf{X}_{t_r}\|}$ and $d_r \leftarrow \frac{1}{2}\mathbf{n}^T(\mathbf{X}_{s_r} + \mathbf{X}_{t_r})$.
    $\mathcal{I}_r \leftarrow \{s : s \in \mathcal{I}_l \wedge \|\mathbf{P}_s - \mathbf{P}_{t^\star}\| < \nu \wedge \left\|\mathbf{X}_s - 2\mathbf{n}_r\mathbf{n}_r^T\mathbf{X}_{t^\star} + 2\mathbf{n}_r d_r\right\| < \mu\}$.
    $t^\star = \arg\min_{t \in \mathcal{I}_c} \left\|\mathbf{X}_s - 2\mathbf{n}_r\mathbf{n}_r^T\mathbf{X}_t + 2\mathbf{n}_r d_r\right\|$.
    if $|\mathcal{I}_r| > n_m$ **then**
      $n_m \leftarrow |\mathcal{I}_r|$, $\mathcal{I}_m \leftarrow \mathcal{I}_r$, $\mathbf{n}_m \leftarrow \mathbf{n}_r$, and $d_m \leftarrow d_r$.
    end if
  end for
  Optimal plane of symmetry $\pi_m = \begin{bmatrix}\mathbf{n}_m^T & -d_m\end{bmatrix}^T$.
  for $i = 1$ to 2^{m-1} **do**
    Set $\tilde{\mathbf{P}}_{j_s} \leftarrow \tilde{\mathbf{P}}_{i_{t^\star}}$ and $\tilde{\boldsymbol{P}}_{j_s} \leftarrow \tilde{\boldsymbol{P}}_{i_{t^\star}}$,
    where $j = i + 2^{m-1} \wedge s \in \mathcal{I} \wedge \|\mathbf{P}_s - \mathbf{P}_{t^\star}\| < \nu \wedge \left\|\mathbf{X}_s - 2\mathbf{n}_m\mathbf{n}_m^T\mathbf{X}_{t^\star} + 2\mathbf{n}_m d_m\right\| < \mu$.
    $t^\star = \arg\min_{t \in \mathcal{I}_c} \left\|\mathbf{X}_s - 2\mathbf{n}_m\mathbf{n}_m^T\mathbf{X}_t + 2\mathbf{n}_m d_m\right\|$.
  end for
  Update $\mathcal{I}_c \leftarrow \mathcal{I}_c \cup \mathcal{I}_m$ and $\mathcal{I}_l \leftarrow \mathcal{I}_l \setminus \mathcal{I}_m$.
end for
if $|\mathcal{I}_l| > 0$ **then**
  $\forall s \in \mathcal{I}_l$, set $\tilde{\mathbf{P}}_{1_s} \leftarrow \mathbf{P}_s$ and $\tilde{\boldsymbol{\delta}}_{1_s} \leftarrow 0$.
end if

For every vertex on the 3D model $\mathbf{X}_s, s \in \mathcal{I}$, we find an assignment of reflectance values that optimizes the following objective:

$$\left\{i_1^\star, ..., i_{|\mathcal{I}|}^\star\right\} = \arg\min_{i_1, \cdots, i_{|\mathcal{I}|}} \sum_{s=1}^{|\mathcal{I}|} \Phi(\tilde{\mathbf{P}}_{i_s}) + \sum_{s=1}^{|\mathcal{I}|} \sum_{t \in \mathcal{K}_s} \Phi(\tilde{\mathbf{P}}_{i_s}, \tilde{\mathbf{P}}_{i_t}). \quad (14)$$

The pairwise term in the objective function, $\Phi(\cdot, \cdot)$ enforces neighborhood smoothness via the Euclidean distance. Here $\mathcal{K}_s$ represents the set of indices for the k nearest neighbors of $\mathbf{X}_s$. We bias the algorithm to select candidates from the same layer using a weighting factor of $0 < \beta < 1$. This provides consistency of texture. We use the following form for the pairwise terms:

$$\Phi(\tilde{\mathbf{P}}_{i_s}, \tilde{\mathbf{P}}_{i_t}) \begin{cases} \beta \left\|\tilde{\mathbf{P}}_{i_s} - \tilde{\mathbf{P}}_{i_t}\right\|_2^2 & \text{if } i_s = i_t, \\ \left\|\tilde{\mathbf{P}}_{i_s} - \tilde{\mathbf{P}}_{i_t}\right\|_2^2 & \text{otherwise.} \end{cases} \quad (15)$$

The unary term $\Phi(\cdot)$ forces visible vertices to receive the reflectance computed in Section 5. We set the unary term at the first layer for visible vertices to ζ, where $0 < \zeta < 1$, else we set it to 1:

$$\Phi(\tilde{\mathbf{P}}_{i_s}) = \begin{cases} \zeta & \text{if } s \in \mathcal{I}_v, i_s = 0, \\ 1 & \text{otherwise.} \end{cases} \quad (16)$$

We use the tree-reweighted message passing algorithm to perform the optimization in Equation (14). We use the computed assignment to obtain the reflectance values $\overline{\mathbf{P}}^\star$ for all vertices in the set $\mathcal{I}$.

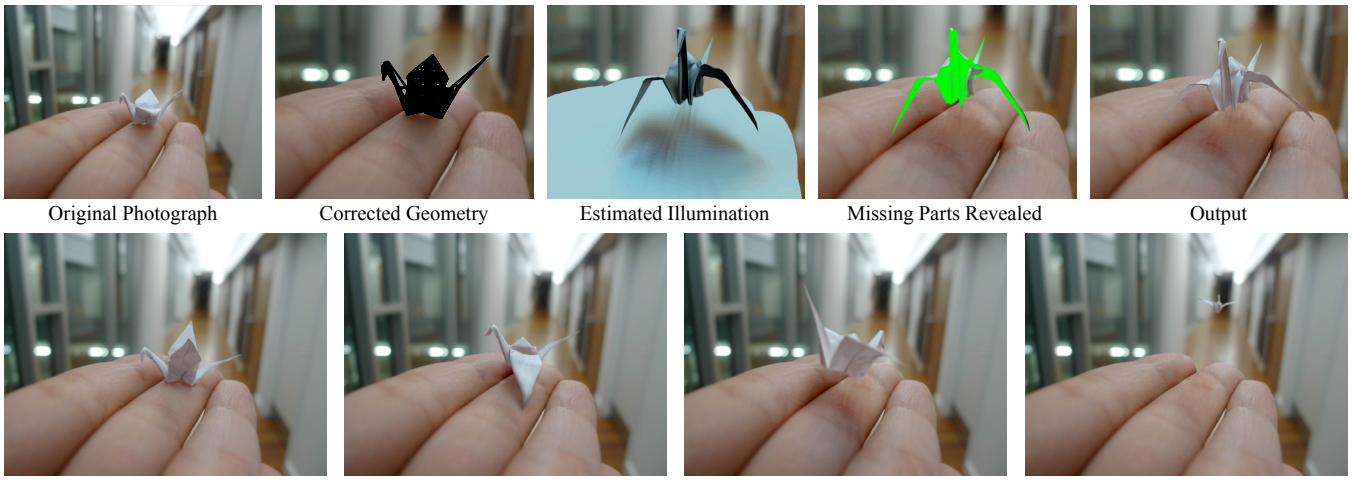


Figure 6: *Top row: 3D manipulation of an origami crane. We show the corrected geometry for the crane in the original photograph, and the estimated illumination, missing parts, and final output for a manipulation. Bottom row: Our approach uses standard animation software to create realistic animations such as the flying origami crane.* Photo Credits: ©Natasha Kholgade.

We build an analogous graph for the appearance difference $\overline{\boldsymbol{\delta}}^{\star}$, and solve an analogous optimization to find its assignment. The assignment step covers the entire object with completed appearance. For areas of the object that do not satisfy the criteria of geometric symmetry and appearance similarity, such as the underside of the taxi cab in Figure 1, the assignment defaults to the stock model appearance. The assignment also defaults to the stock model appearance when after several iterations, the remaining parts of the object are partitioned into several small areas where the object lacks structural symmetries relative to the visible areas. In this case, we allow the user to fill the appearance in these areas on the texture map of the 3D model using PatchMatch [Barnes et al. 2009].

7 Final Composition

The user manipulates the object pose from $\boldsymbol{\Theta}$ to $\boldsymbol{\Omega}$. Given the corrected geometry, estimated illumination, and completed appearance and from Sections 4 to 6, we create the result of the manipulation $\mathbf{J}$ by replacing $\boldsymbol{\Theta}$ with $\boldsymbol{\Omega}$ in Equation (1). We use ray-tracing to render each pixel according to Equation (8), using the illumination $\overline{\mathbf{L}}^{\star}$, geometry $\overline{\mathbf{X}}$, and reflectance $\overline{\mathbf{P}}^{\star}$. We add the appearance difference $\overline{\boldsymbol{\delta}}^{\star}$ to the rendering to produce the final pixels on the manipulated object. We render pixels for the object and the ground using this method, while leaving the rest of the photograph unchanged (such as the corridor in Figure 6). To handle aliasing, we perform the illumination estimation, appearance completion, and compositing using a super-sampled version of the photograph. We filter and subsample the composite to create an anti-aliased result.

8 Results

We perform a variety of 3D manipulations to photographs, such as rigid manipulation of photographed objects, deformation, and 3D copy-paste. We use a separately captured background photograph for the chair, while for all other photos, we fill the background using Context-Aware Fill in Photoshop. The top row of Figure 6 shows the aligned geometry, estimated illumination, missing parts, and final result of a manipulation performed to an origami crane. Though the illumination may not be accurate, in combination with the corrected reflectance, it produces plausible results such as the shadows of the wing on the hand and illumination changes on the crane. Our approach can directly be tied to standard modeling and animation software to create realistic animations from a single photograph, such as the flying origami crane in the bottom row of Figure 6. Figure 7 shows the result of our approach on the laptop from Zheng et al. [2012]. Unlike their approach, we reveal the hidden logo using the stock 3D model while maintaining plausible illumination over the object and the contact surface.

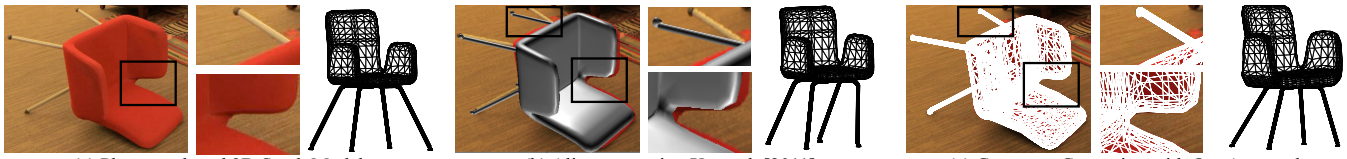
In Figure 13, we show 3D manipulations such as rotation, translation, copy-paste, and deformation on the chair from Figure 2, a pen, a subject's watch, some fruit, a painting, a car on a cliff, a top hat, and a historical photograph of airplanes shot during World War II. As the shown in the second column, our approach replicates the original photograph in the first column to provide seamless transition from the original view to new manipulations of the object. We show intermediate outputs in supplementary material. Our approach allows users to create dynamic compositions such as the levitating chair, the watches flying about the subject, and the falling fruit in Figure 13, and the taxi jam from Figure 1. Users can add creative effects to the photograph, such as the pen strokes on the paper, in conjunction with 3D manipulations. The illumination estimated in Section 5 can be used to plausibly relight new 3D objects inserted into the scene as shown in Figure 12. We also perform non-rigid deformations to objects using the geometry correction approach of Section 4, such as converting the top hat in Figure 13 into a magician's hat. Through our approach, the user changes the story of the car photograph in Figure 13 from two subjects posing next to a parked car to them watching the car as it falls off the cliff.

Figure 11 shows an example where the taxi-cab at the top of the photograph in Figure 1 is manipulated using 3D models of three different cars. Accurate alignment is obtained for the first two models. Our appearance completion algorithm determines plausible appearance, even when the stock 3D model of the second car deviates from the original photograph. For significantly different geometry, such as that of the third car, alignment may be less accurate, leading to artifacts such as transfer of appearance from the body to the windshield or the wheels. However, the approach maintains plausible illumination with similar environment maps, as the geometry of the 3D models provides sufficient cues for diffuse illumination.

While our approach is designed for digital photographs, it provides plausible results for vintage photographs such as the historical World War II photograph in Figure 13 and the non-photorealistic media such as the vegetables' painting in Figure 13. The appear-

Original Photograph from Zheng et al. [2012] | Object manipulation using Zheng et al. [2012] | Object manipulation using our approach | Revealing unseen parts using our approach

Figure 7: *We perform a 3D rotation of the laptop in a photograph from Zheng et al. [2012]* (Copyright: ©ACM). *Unlike their approach, we can reveal the hidden cover and logo of the laptop.*



(a) Photograph and 3D Stock Model | (b) Alignment using Xu et al. [2011] | (c) Geometry Correction with Our Approach

Figure 8: *Comparison of geometry correction by our approach against the alignment of Xu et al. [2011]. As shown in the insets, Xu et al. do not align the leg and the seat accurately. Through our approach, the user accurately aligns the model to the photograph.*

ance estimation described in Section 5 estimates grayscale appearances for the black-and-white photograph of the airplanes. Using our approach, the user manipulates the airplanes to pose them as if they were pointing towards the camera, an effect that would be nearly impossible to capture in the actual scene. In the case of paintings, our approach maintains the style of the painting and the grain of the sheet, by transferring these through the fine-scale detail difference in the appearance completion described in Section 6.

Geometry Evaluation. Figure 8 shows results of geometry correction through our user-guided approach compared to the semi-automated approach of Xu et al. [2011] on the chair. In the system of Xu et al., the user input involves seeding a graph-cut segmentation algorithm with a bounding box, and rigidly aligning the model. Their approach automatically segments the photographed object based on connected components in the model, and deforms the 3D model to resemble the photograph. Their approach approximates the form of most of the objects. However, as shown by the insets in Figure 8(b), it fails to exactly match the model to the photograph. Through our approach (shown in Figure 8(c)), users can accurately correct the geometry to match the photographed objects. Supplementary material shows comparisons of our approach with Xu et al. for photographs of the crane, taxi-cab, banana, and mango.

Illumination Evaluation. To evaluate the illumination estimation, we captured fifteen ground truth photographs of the chair from Figure 2 in various orientations and locations using a Canon EOS 5d Mark II Digital SLR camera mounted on a tripod and fitted with an aspheric lens. The photographs are shown in the supplementary material. We also capture a photograph of the scene without the object to provide a ground-truth background image. We aligned the 3D model of the chair to each of the fifteen photographs using our geometry correction approach from Section 4, and evaluated our illumination and reflectance estimation approach from Section 5 against a ground truth light probe, and three approaches: (1) Haar wavelets with positivity constraints on coefficients [2009], (2) L_1-sparse high frequency illumination with spherical harmonics for low-frequency illumination [Mei et al. 2009], and (3) environment map completion using projected background [Khan et al. 2006]. We fill the parts of the Khan et al. environment map not seen in the image using the PatchMatch algorithm [Barnes et al. 2009].

For all experiments, we estimated the illumination using Photograph 1 (shown at the left of Figure 10(a)), and synthesized images corresponding to the poses of the chair in Photographs 2 to 15 by applying the estimated illumination to the geometry of each pose. The synthesized images contain all steps of our approach except appearance completion on the chair. We computed mean-squared reconstruction errors between Photographs 2 to 15 and their corresponding synthesized images for three types of subregions in the photograph: object and ground, ground only, and object only. As a baseline, we obtained the ground truth illumination by capturing a high dynamic range (HDR) image of a light probe (a 4-inch chrome sphere inspired by Debevec [1998]). Figure 9 shows the mean-squared reconstruction error (MSE) for the von Mises-Fisher basis compared against Haber et al., Mei et al., Khan et al., and the light probe, for increasing numbers of basis components (i.e. K). We used the same discretization for the sphere as the number of components. The number of basis components is a power of two to ensure that the Haar wavelet basis is orthonormal. We used $\lambda_1 = .01$ and $\lambda_2 = 1$ for our method, a regularization weight of 1 for the approach of Haber et al., and a regularization weight of .01 for the method of Mei et al. The reconstruction error for Haar wavelets increases with increasing number of basis components, since in attempting to capture high frequency information (such as the edges of shadows), they introduce artifacts of negative illumination such as highlights.

Figure 10 shows environment maps for the light probe, the background image projected out according to Khan et al., and the three approaches to estimate illumination. The second and third rows of Figure 10 show images of the chair in Photograph 2 synthesized using the light probe, the background image, and the three illumination estimation approaches with $K = 4096$. The ground truth for Photograph 2 is shown at the top-left. As shown in the figure, the approach of Khan et al. does not capture the influence of ceiling lights which are not observed in the original photograph. The Haar wavelets' approach introduces highlights due to negative light, while the approach of Mei et al. generates sharp cast shadows. Our approach produces smooth cast shadows and captures the effect of lights not seen in the original photograph. Additional examples can be found in the supplementary material.

Geometry Correction Timings. We evaluated the time taken for four users to perform geometry correction of 3D models using our user-guided approach. Table 1 shows the times (in minutes) to align various 3D models to photographs used in this paper. The time

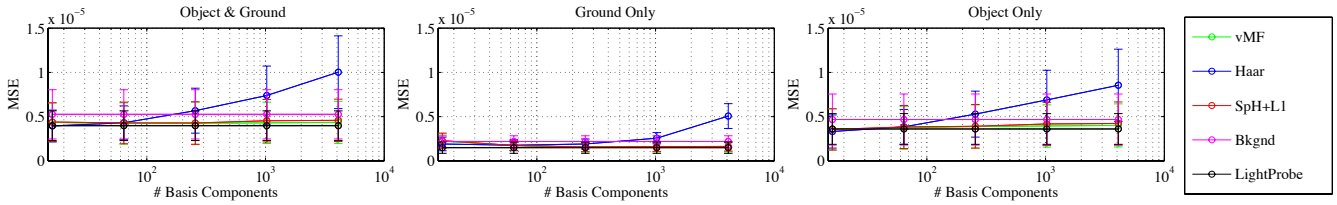


Figure 9: *Plots of mean-squared reconstruction error (MSE) versus number of basis components for the vMF basis in green (method used in this paper) compared to the Haar basis [Haber et al. 2009], spherical harmonics with L_1 prior (Sph+L1) [Mei et al. 2009], background image projected (Bkgnd) [Khan et al. 2006], and a light probe, on the object and the ground, ground only, and object only.*

Table 1: *Times taken (minutes) to align 3D models to photographs.*

User	Banana	Mango	Top hat	Taxi	Chair	Crane
1	12.43	7.10	8.72	16.09	45.17	32.22
2	6.33	3.08	10.08	7.75	20.32	14.92
3	2.23	2.42	4.93	2.57	6.12	22.07
4	5.63	5.13	6.17	7.52	8.53	19.03

spent on aligning is directly related to the complexity of the model in terms of the number of connected components. Users spent the least amount of time on aligning the mango, banana, and top hat models, as these models consist of a single connected component each. Though the taxi consists of 24 connected components, its times are comparable to the simpler models, as the stock model geometry is close to the photographed taxi. Long alignment times of the chair and origami crane are due to the number of connected components (3 and 8 respectively), and the large disparity between their stock models and their photographs. We show user alignment examples in the supplementary material. The supplementary video shows a session to correct the banana model using the tool.

User Study. We evaluated the perceived realism of 3D object manipulation in photographs through a two-alternative forced choice user study. We asked participants to compare and choose the more realistic image between the original photograph and an edited result produced using our approach. We recruited 39 participants mostly from a pool of graduate students in computer science, and we conducted the study using a webpage-based survey. Each participant viewed four image pairs. Each image pair presented a choice between an original photograph and one of two 'edited' conditions: either the final result of our approach (Condition 1), or an intermediate result also produced using our method (Condition 2). Specifically, Condition 2 was generated using the corrected geometry, illumination, and surface reflectance, but without using the appearance difference. The presentation order was randomized, and each user saw each of the original photographs once. Across all participants, we obtained a total of 40 responses for the original photo/final result image pairs, and 38 responses for the original photo/intermediate output image pairs. 71.05% of the time (i.e., on 54 of 76 image pairs), participants chose the original photograph as being more realistic than Condition 2. This preference was found to be statistically significant: we used a one-sample, one-tailed t-test, which rejected the null hypothesis that people do not prefer the original photograph over Condition 2 at the 5% level ($p < 0.001$). 52.5% of the time (i.e., on 42 of 80 image pairs), the participants chose Condition 1 over the original photograph. The difference was not found to be significant at the 5% level. In some cases, participants reported choosing the more realistic image based on plausible configurations of the object, e.g., an upright well-placed chair as opposed to one fallen on the ground. Images used are shown in the supplementary material.

Parameters. During geometry correction, the user can turn on symmetry by setting $w = 1$ or turn it off by setting $w = 0$. For illumination extraction and appearance correction, we set the illumination parameters λ_1 and λ_2 to small values for smooth shadows due to dispersed illumination, i.e., $\lambda_1 = .001$ to $.01$, $\lambda_2 = .1$ to 1. For strong directed shadows due to sparse illumination concentrated in one region, we set high values: $\lambda_1 = 1$ to 5, $\lambda_2 = 10$ to 20. To emphasize ground shadows, we set $\tau = .01$ to $.25$. The value of K is set to 2500. We set $\lambda_3 = .5$ to emphasize piecewise constancy of the appearance deviations. In the appearance completion section, we set the number of nearest neighbors k for all $\mathcal{K}_s$ to 5. We set M to 1 for rigid objects such as the chair, the laptop, and the cars, and to 7 for flexible objects such as fruit. We set $\beta = .005$ to emphasize consistency of appearance within a layer, and we set ζ to a small value (.01) to force visible vertices to be assigned the observed appearance. For all RANSAC operations, we set $R = 5000$, $\mu = .01$ times the bounding box of the object, and $\nu = .001$. We vary the geometric symmetry tolerance μ from .01 to .1 times the bounding box for increasingly approximate symmetries. Parameters we vary the most in our algorithm are λ_1, λ_2, and τ. Parameter settings for examples used in the paper can be found in supplementary material.

9 Discussion

We present an approach for performing intuitive 3D manipulations on photographs, without requiring users to access the original scene or the moment when the photograph was taken. 3D manipulations greatly expands the repertoire of creative manipulations that artists can perform on photographs. For instance, controlling the composition of dynamic events (such as Halsman's *Dalí Atomicus*) is particularly challenging for artists. Manipulation of photographed objects in 3D will allow artists far greater freedom to experiment with many different compositions. Through our approach, artists can conveniently create stop motion animations that defy physical constraints. While we have introduced our approach to improve the experience of editing photographs, it can be used to correct the geometry and appearance of 3D models in public repositories.

The fundamental limitation of our approach is related to sampling. We may lack pixel samples for a photographed object if it is small in image-space, in which case, manipulating it to move it closer to the camera can produce a blurred result. However, as cameras today exceed tens of megapixels in resolution, this problem is much less of an issue, and may be addressed via final touch-ups in 2D. Failures can occur if an object is photographed with the camera's look-at vector perpendicular to the normal of the bilateral plane of symmetry, e.g., if a wine-bottle is photographed from the top. In these cases, symmetry constraints are difficult to exploit, and the completion has to rely heavily on the texture provided with the 3D model. Another class of failures is caused when the illumination model fails to account for some lighting effects, and the algorithm attempts to explain the residual effect during appearance completion. This can result in lighting effects appearing as texture artifacts.

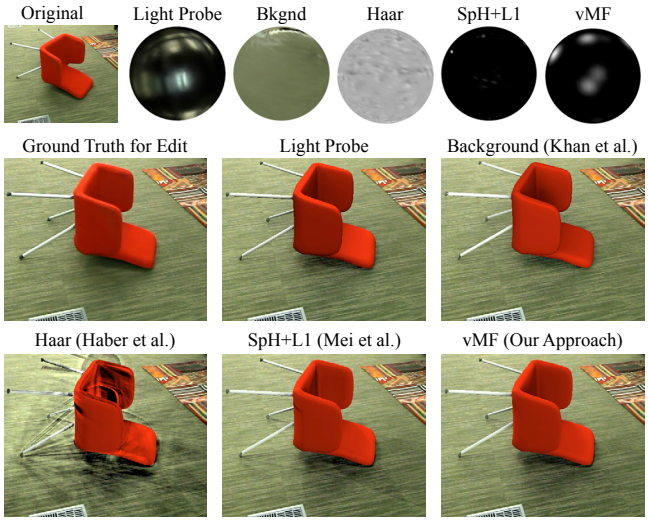


Figure 10: [Figure is best viewed on a monitor] *Comparison of von Mises-Fisher (vMF) basis for illumination estimation versus the Haar basis [Haber et al. 2009], spherical harmonics with L_1-prior on high frequency illumination (SpH+L1) [Mei et al. 2009], background image backprojected (Bkgnd) [Khan et al. 2006], and a light probe. Top row: original photograph and environment maps for $K = 4096$. Bottom two rows: ground truth and synthesized images for a 3D manipulation on the chair. Our approach produces soft cast shadows, avoids highlights, and captures the effect of lights not seen in the original photograph.*

Figure 11: *We manipulate the taxi-cab at the top of the photograph in Figure 1 using 3D models of three different car makes. As the 3D model deviates from the photograph, the alignment may become less accurate, leading to artifacts such as mapping of appearance from the body to the windshield and wheels for the third car. In all cases, the approach maintains plausible illumination with similar environment maps.*

Figure 12: *The illumination generated using our approach can be used to plausibly relight new objects inserted into the scene.*

Finally, failures can occur if the model from the 3D repository is not correctly or too coarsely designed, particularly in cases of objects with complex geometry and small parts.

While large collections of 3D models are available online, there are several objects for which models may not be found. The exponential trend in the availability of online 3D models suggests that models not found today will be available online in the near future. We expect that rising ubiquity in 3D scanning and printing technologies, and the tendency towards standardization in object design and manufacture, will contribute to the increase in model availability. The more pressing question will soon be not whether a particular model exists online, but rather, whether the user can find the model in a database of millions. A crucial area of future research will be to automate the search and alignment of 3D models to photographs. Finally, while we address manipulations of photographs in 3D, extending these ideas to editing videos would vastly expand creative control in the temporal domain.

Acknowledgments. This work was funded in part by the Google Research Grant. We would like to thank James McCann, Srinivasa Narasimhan, Sean Banerjee, Leonid Sigal, and Jessica Hodgins for their valuable comments on the paper. In addition, we thank Spencer Diaz and Moshe Mahler for help with animations.

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 *Proc. CVPR*.

AVIDAN, S., AND SHAMIR, A. 2007. Seam carving for content-aware image resizing. In *Proc. ACM SIGGRAPH*.

BARNES, C., SHECHTMAN, E., FINKELSTEIN, A., AND GOLDMAN, D. B. 2009. Patchmatch: a randomized correspondence algorithm for structural image editing. In *Proc. ACM SIGGRAPH*, 24:1–24:11.

BARRETT, W. A., AND CHENEY, A. S. 2002. Object-based image editing. In *Proc. ACM SIGGRAPH*, 777–784.

BARRON, J. T. 2012. Shape, albedo, and illumination from a single image of an unknown object. In *CVPR*, 334–341.

BLANZ, V., AND VETTER, T. 1999. A morphable model for the synthesis of 3d faces. In *Proc. ACM SIGGRAPH*, 187–194.

BOKELOH, M., WAND, M., KOLTUN, V., AND SEIDEL, H.-P. 2011. Pattern-aware shape deformation using sliding dockers. *ACM Trans. Graph. 30*, 6 (Dec.), 123:1–123:10.

CHEN, J., PARIS, S., WANG, J., MATUSIK, W., COHEN, M., AND DURAND, F. 2011. The video mesh: A data structure for image-based three-dimensional video editing. In *ICCP*, 1–8.

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

DEBEVEC, P. E., TAYLOR, C. J., AND MALIK, J. 1996. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Proc. ACM SIGGRAPH*, 11–20.

DEBEVEC, P. 1998. Rendering synthetic objects into real scenes: bridging traditional and image-based graphics with global illumination and high dynamic range photography. In *Proc. ACM SIGGRAPH*, 189–198.

DURAND, F. 2002. An invitation to discuss computer depiction. In *Proc. ACM NPAR*, 111–124.

FANG, H., AND HART, J. C. 2004. Textureshop: texture synthesis as a photograph editing tool. *Proc. ACM SIGGRAPH*, 354–359.

FISHER, R. 1953. Dispersion on a sphere. In *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences*, vol. 217, 295–305.

GAL, R., AND COHEN-OR, D. 2006. Salient geometric features for partial shape matching and similarity. *ACM Trans. Graph. 25*, 1 (Jan.), 130–150.

GAL, R., WEXLER, Y., OFEK, E., HOPPE, H., AND COHEN-OR, D. 2010. Seamless montage for texturing models. *Comput. Graph. Forum 29*, 2, 479–486.

GOLDBERG, C., CHEN, T., ZHANG, F.-L., SHAMIR, A., AND HU, S.-M. 2012. Data-driven object manipulation in images. *Computer Graphics Forum 31*, 2pt1, 265–274.

HABER, T., FUCHS, C., BEKAERT, P., SEIDEL, H.-P., GOESELE, M., AND LENSCH, H. P. A. 2009. Relighting objects from image collections. In *CVPR*, IEEE, 627–634.

HARA, K., NISHINO, K., AND IKEUCHI, K. 2008. Mixture of spherical distributions for single-view relighting. *IEEE Trans. Pattern Anal. Mach. Intell. 30*, 1, 25–35.

HONG, W., YANG, A. Y., HUANG, K., AND MA, Y. 2004. On symmetry and multiple-view geometry: Structure, pose, and calibration from a single image. *IJCV 60*, 3 (Dec.), 241–265.

KARSCH, K., HEDAU, V., FORSYTH, D., AND HOIEM, D. 2011. Rendering synthetic objects into legacy photographs. In *Proc. ACM SIGGRAPH Asia*, 157:1–157:12.

KHAN, E. A., REINHARD, E., FLEMING, R. W., AND BÜLTHOFF, H. H. 2006. Image-based material editing. In *Proc. ACM SIGGRAPH*, 654–663.

KIM, V. G., LIPMAN, Y., AND FUNKHOUSER, T. 2012. Symmetry-guided texture synthesis and manipulation. *ACM Trans. Graph. 31*, 3 (June), 22:1–22:14.

KOLMOGOROV, V. 2006. Convergent tree-reweighted message passing for energy minimization. *IEEE TPAMI 28*, 10 (Oct.), 1568–1583.

KOPF, J., NEUBERT, B., CHEN, B., COHEN, M., COHEN-OR, D., DEUSSEN, O., UYTTENDAELE, M., AND LISCHINSKI, D. 2008. Deep photo: model-based photograph enhancement and viewing. In *Proc. ACM SIGGRAPH Asia*, 116:1–116:10.

KRAEVOY, V., SHEFFER, A., AND GOTSMAN, C. 2003. Matchmaker: constructing constrained texture maps. *ACM Trans. Graph. 22*, 3 (July), 326–333.

KRAEVOY, V., SHEFFER, A., AND VAN DE PANNE, M. 2009. Modeling from contour drawings. In *Proceedings of the 6th Eurographics Symposium on Sketch-Based Interfaces and Modeling*, ACM, New York, NY, USA, SBIM '09, 37–44.

KWATRA, V., SCHODL, A., ESSA, I., TURK, G., AND BOBICK, A. 2003. Graphcut textures: Image and video synthesis using graph cuts. *ACM Trans. Graph. 22*, 3 (July), 277–286.

LALONDE, J.-F., HOIEM, D., EFROS, A. A., ROTHER, C., WINN, J., AND CRIMINISI, A. 2007. Photo clip art. *Proc. ACM SIGGRAPH 26*, 3 (August), 3.

LAND, E. H., JOHN, AND MCCANN, J. 1971. Lightness and retinex theory. *Journal of the Optical Society of America*, 1–11.

LEPETIT, V., MORENO-NOGUER, F., AND FUA, P. 2009. Epnp: An accurate o(n) solution to the pnp problem. *Int. J. Comput. Vision 81*, 155–166.

LIM, J., PIRIAVASH, H., AND TORRALBA, A. 2013. Parsing ikea objects: Fine pose estimation. In *ICCV*.

MEI, X., LING, H., AND JACOBS, D. 2009. Sparse representation of cast shadows via $l1$-regularized least squares. In *ICCV*.

MITRA, N. J., AND PAULY, M. 2008. Symmetry for architectural design. In *Advances in Architectural Geometry*, 13–16.

MITRA, N. J., GUIBAS, L. J., AND PAULY, M. 2006. Partial and approximate symmetry detection for 3d geometry. In *Proc. ACM SIGGRAPH*, 560–568.

NEALEN, A., SORKINE, O., ALEXA, M., AND COHEN-OR, D. 2005. A sketch-based interface for detail-preserving mesh editing. *ACM Trans. Graph. 24*, 3 (July), 1142–1147.

NG, R., RAMAMOORTHI, R., AND HANRAHAN, P. 2003. All-frequency shadows using non-linear wavelet lighting approximation. In *Proc. ACM SIGGRAPH*, 376–381.

OH, B. M., CHEN, M., DORSEY, J., AND DURAND, F. 2001. Image-based modeling and photo editing. In *Proc. ACM SIGGRAPH*, 433–442.

OKABE, T., SATO, I., AND SATO, Y. 2004. Spherical harmonics vs. haar wavelets: Basis for recovering illumination from cast shadows. In *CVPR*, 50–57.

PANAGOPOULOS, A., SAMARAS, D., AND PARAGIOS, N. 2009. Robust shadow and illumination estimation using a mixture model. In *CVPR*, 651–658.

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

PRASAD, M., ZISSERMAN, A., AND FITZGIBBON, A. W. 2006. Single view reconstruction of curved surfaces. In *CVPR*.

RAMAMOORTHI, R., AND HANRAHAN, P. 2001. On the relationship between radiance and irradiance: determining the illumination from images of a convex lambertian object. *J. Opt. Soc. Am. A 18*, 10, 2448–2459.

ROMEIRO, F., AND ZICKLER, T. 2010. Blind reflectometry. In *ECCV*, 45–58.

SIMPSON, J., 2003. Oxford English Dictionary Online, 2nd edition. `http://www.oed.com/`, July.

SORKINE, O., AND ALEXA, M. 2007. As-rigid-as-possible surface modeling. In *Proc. SGP*, 109–116.

TERZOPOULOS, D., WITKIN, A., AND KASS, M. 1987. Symmetry-seeking models and 3d object reconstruction. *International Journal of Computer Vision 1*, 211–221.

TZUR, Y., AND TAL, A. 2009. Flexistickers: photogrammetric texture mapping using casual images. *ACM Trans. Graph. 28*, 3 (July), 45:1–45:10.

XU, K., ZHENG, H., ZHANG, H., COHEN-OR, D., , LIU, L., AND XIONG, Y. 2011. Photo-inspired model-driven 3d object modeling. *ACM Transactions on Graphics 30*, 4.

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

ZOU, H., AND HASTIE, T. 2005. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society, Series B 67*, 301–320.

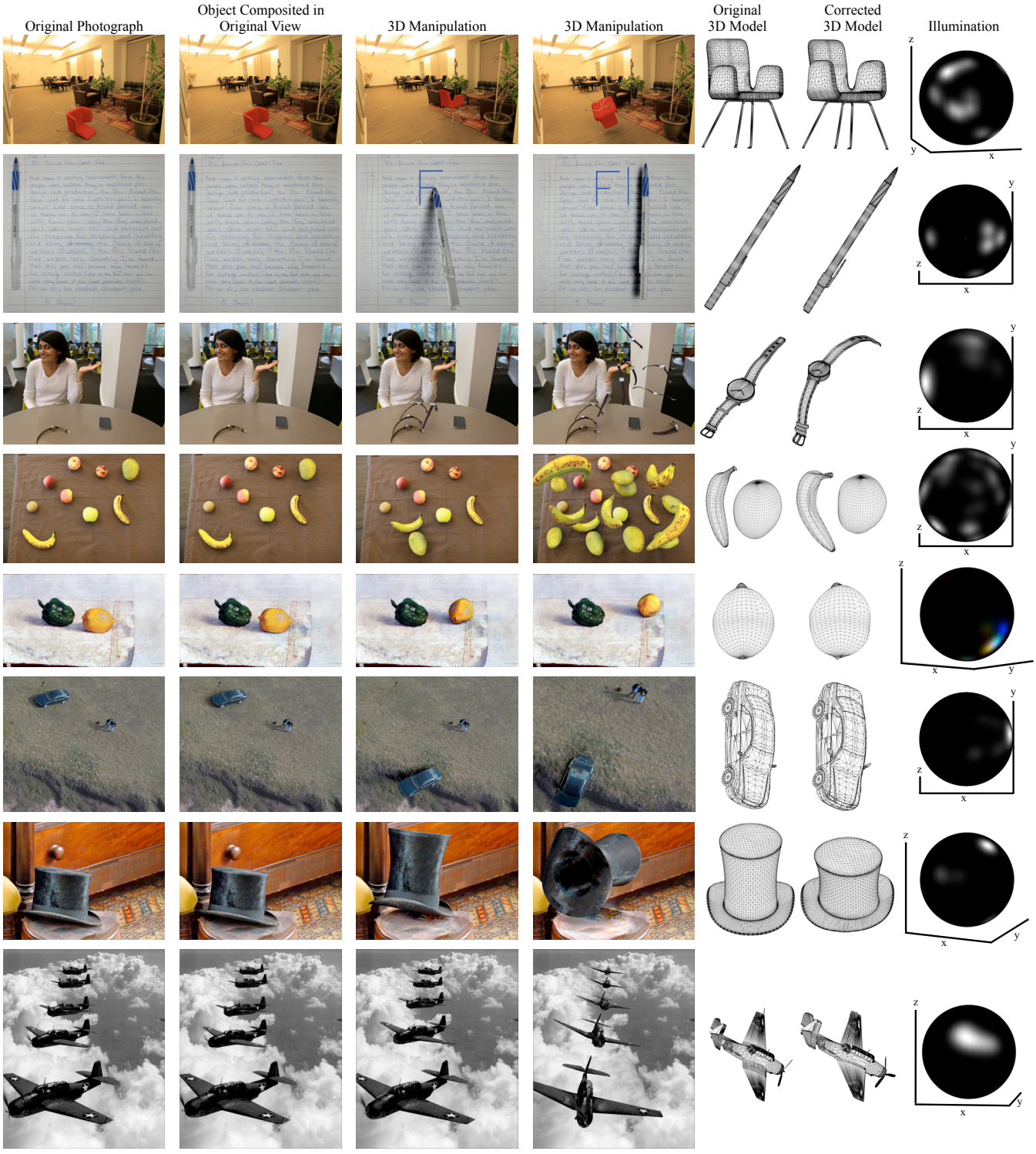


Figure 13: *3D Manipulations (rotation, translation, copy-paste, deformation) to a chair, a pen* (Photo Credits: © Christopher Davis)*, a subject's watch, fruit, a painting* (Credits: © Odilon Redon)*, a car on a cliff* (Photo Credits: © rpriegu)*, a top hat* (Photo Credits: © tony_the_bald_eagle)*, and a historical photograph of World War II Avengers* (Photo Credits: © Naval Photographic Center)*. As the shown in the second column, our approach plausibly replicates the the original photograph in the first column, which enables our approach to achieve a seamless transition in image appearance when new manipulations are done to the object. The illumination is shown as an environment map whose x- and y-axes represent the image plane, and whose z-axis represents the direction of the camera into the scene. Through our approach, users can create dynamic compositions such as levitating chairs, flying watches, falling fruit, and diving cars, combine 3D manipulations with creative effects such as pen strokes and painting styles, and create object deformations such as the top hat resized and curved to a magician's hat.*