Appearance Modeling via Proxy-to-Image Alignment

HUI HUANG, KE XIE, and LIN MA, Shenzhen University
DANI LISCHINSKI, Hebrew University of Jerusalem
MINGLUN GONG, Memorial University of Newfoundland
XIN TONG, Microsoft Research Asia
DANIEL COHEN-OR, Shenzhen University and Tel Aviv University

Endowing 3D objects with realistic surface appearance is a challenging and time-demanding task, as real-world surfaces typically exhibit a plethora of spatially variant geometric and photometric detail. Not surprisingly, computer artists commonly use images of real-world objects as an inspiration and a reference for their digital creations. However, despite two decades of research on image-based modeling, there are still no tools available for automatically extracting the detailed appearance (microgeometry and texture) of a 3D surface from a single image. In this article, we present a novel user-assisted approach for quickly and easily extracting a nonparametric appearance model from a single photograph of a reference object.

The extraction process requires a user-provided proxy, whose geometry roughly approximates that of the object in the image. Since the proxy is just a rough approximation, it is necessary to align and deform it so as to match the reference object. The main contribution of this work is a novel technique to perform such an alignment, which enables accurate joint recovery of geometric detail and reflectance. The correlations between the recovered geometry at various scales and the spatially varying reflectance constitute a nonparametric appearance model. Once extracted, the appearance model may then be applied to various 3D shapes, whose large-scale geometry may differ considerably from that of the original reference object. Thus, our approach makes it possible to construct an appearance library, allowing users to easily enrich detail-less 3D shapes with realistic geometric detail and surface texture.

CCS Concepts: • **Computing methodologies → Texturing**; *Image-based rendering*; Reflectance modeling;

Additional Key Words and Phrases: Appearance modeling, shape alignment, shape deformation, intrinsic image decomposition, detail extraction, appearance transfer, texture synthesis

ACM Reference format:
Hui Huang, Ke Xie, Lin Ma, Dani Lischinski, Minglun Gong, Xin Tong, and Daniel Cohen-Or. 2018. Appearance Modeling via Proxy-to-Image Alignment. *ACM Trans. Graph.* 37, 1, Article 10 (January 2018), 15 pages.
https://doi.org/10.1145/3158353

1 INTRODUCTION

Today's high-end computer games and motion picture special effects require creating highly detailed 3D models with photorealistic appearance. Modeling such objects from scratch is extremely difficult and time consuming even for expert modelers. This is particularly true for complex objects with irregular shapes and visually interesting fine-scale detail. Not surprisingly, computer artists commonly use images of real-world objects to serve as an inspiration and a reference for their digital creations. This is greatly assisted by the ubiquity of digital imagery of just about any conceivable object and the ability to quickly find an image exhibiting a desired object appearance on the Internet.

However, despite the considerable amount of research dedicated to image-based modeling of geometry and appearance, it is still not feasible to automatically extract a fully detailed realistic 3D object model from a single image: state-of-the-art automatic computer vision methods still rely on a variety of simplifying assumptions, which may not hold in practice.

Creating a fully detailed object model is difficult; however, producing a coarse, rough 3D shape is a much easier task. With currently available interactive modeling tools, an experienced modeler can create a rough shape, such as the smooth fire hydrant and R2D2 models in Figure 1, very quickly and effortlessly. Alternatively, in some cases, a coarse model can be found in a repository of 3D models. However, enriching the coarse model with medium- and fine-scale geometric deformations and displacements, as well as a realistic surface texture, is a challenging and daunting task even for professionals, and may involve a combination of several sophisticated modeling and texturing tools. For example, it would take an experienced modeler about 3 hours to create a model similar to the one shown on the right of Figure 1 using existing commercial tools.

In this article, we focus on this last challenging and time-consuming step. We address both the process of extracting a nonparametric appearance model from a photograph and the process of applying such a model to a target 3D shape. For the extraction process, we assume that the user has chosen a photo of a *reference object* that exhibits the desired appearance, and that he or she has created or obtained a rough geometric *source proxy* approximating this object (e.g., in Figure 1 on the left). We use this proxy to recover the geometric detail exhibited by the reference object at multiple scales, jointly with a detailed spatially variant reflectance texture.

Since the source proxy is just a coarse approximation of the reference object's geometry, it is first necessary to register the source proxy with the reference object by aligning and nonrigidly deforming the proxy shape. This registration paves the way for accurate

This work was supported in part by the NSFC (61522213, 61761146002), 973 Program (2015CB352501), Guangdong Science and Technology Program (2015A030312015), Natural Science Foundation of Shenzhen University (827-000196), NSERC (2017-06086), ISF (2366/16), and ISF-NSFC Joint Research Program (2472/17, 2217/15).

Authors' addresses: H. Huang (corresponding author), K. Xie, and L. Ma, Shenzhen University, China; emails: {hhzhiyan, ke.xie.siat, majcjc}@gmail.com; D. Lischinski, Hebrew University of Jerusalem, Israel; email: danix3d@gmail.com; M. Gong, Memorial University of Newfoundland, Canada; email: gongml@gmail.com; X. Tong, Microsoft Research Asia, China; email: xtong@microsoft.com; D. Cohen-Or, Shenzhen University, China and Tel Aviv University, Israel; email: cohenor@gmail.com.

https://doi.org/10.1145/3158353

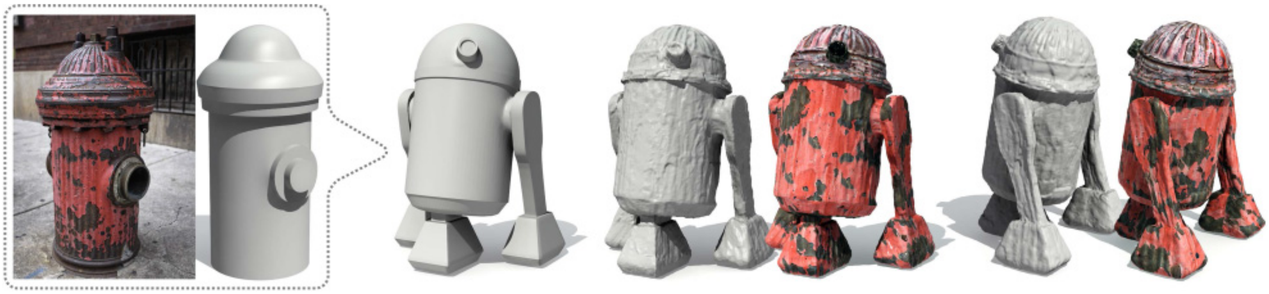

Fig. 1. Assisted by a rough 3D proxy, our approach can extract the geometric and photometric appearance of a fire hydrant from a single photo (left) and transfer it to a new target shape (R2-D2 from *Star Wars*).

joint recovery of geometric detail and diffuse reflectance by enhancing the state-of-the-art method of Barron and Malik (2015). The recovered geometric details, the diffuse reflectance texture, and the correlations between the two constitute our nonparametric appearance model.

Given a detail-less *target shape* we can now transfer the extracted appearance model to this shape, yielding a richly detailed 3D model, whose fine-scale appearance greatly resembles that exhibited by the reference object (as shown in Figure 1 on the right). We first transfer a medium-scale geometric deformation field, and use the result to transfer fine-scale displacements and reflectance in a geometry-correlated manner. We demonstrate that such bi-scale appearance transfer is effective even when the target shape is significantly different from that of the original reference object.

In summary, our contributions are the following:

- A novel method for aligning and deforming a coarse 3D proxy to match a 2D image of the reference object.
- Leveraging the aligned proxy to enhance the performance of Barron and Malik's method (2015) for simultaneous extraction of illumination, shape, and reflectance, and introducing a novel bi-scale deformation representation.
- A new two-step method for geometry-correlated transfer of appearance extracted from the reference object in a single 2D image to a new 3D target shape.

The first two contributions pave the way for constructing a useful appearance library. As a proof of concept, Figure 14 will later demonstrate a small library with five different categories of realistic materials (stone, metal, wood, fabric, and bread). The availability of such libraries, along with our bi-scale appearance transfer technique, would allow users to easily enrich new 3D shape models with medium- and fine-scale geometric deformations and displacements, as well as realistic surface texture.

2 RELATED WORK

Image-based 3D modeling. Much work has been done over the years on creating textured 3D models from photographs. An early example is the pioneering Façade system (Debevec et al. 1996) for creating an architectural model, typically from multiple photographs of a building, with many follow-ups in research and commercial products (Oliveira 2002). Oh et al. (2001) developed a set of tools for fitting a 3D model to a photograph and used the bilateral filter to decouple the illumination from uniformly textured areas, making it possible to perform texture replacement.

More recent and relevant examples include Zheng et al. (2012), who fit objects with cuboid proxies, and the 3-Sweep system (Chen et al. 2013) that offers an intuitive UI for fitting generalized cylinders to objects. The applicability of these approaches is limited to objects having suitably restricted geometry.

Kholgade et al. (2014) fit a stock 3D model to an object in a single image to perform 3D manipulations on that object. Since the stock model cannot be expected to match the object, considerable user assistance is necessary to perform geometric alignment: the user specifies a set of pairwise point correspondences between the image and the model. Once the model is aligned, the environment illumination is estimated, and the object's texture is recovered. A somewhat more automated approach that uses model collections is described by Rematas et al. (2017).

The preceding techniques are better suited for modeling the large-scale geometry of man-made objects, and they model the appearance using color textures recovered from the image (with or without accounting for illumination). In contrast, our goal is to extract a visually complex appearance model that may be transferred to new shapes, which may be rather different from that of the reference object. Importantly, our model captures fine geometric details (at two different scales), in addition to the diffuse reflectance texture.

Model to image fitting. Kraevoy et al. (2009) create new models through deforming 3D templates to fit manually generated 2D contour drawings. Somewhat similarly, we utilize coarse 3D proxies, which are deformed to align with edges in input photos. Like Kraevoy et al. (2009), we also use a hidden Markov model (HMM) (Rabiner 1989) to compute the optimal point correspondences between 3D vertices and 2D edge pixels. However, we solve a much more challenging problem, as automatically detected image edges are much noisier, more fragmented, and more ambiguous than the clean contour drawings used in Kraevoy et al. (2009).

Xu et al. (2011) automatically deform a 3D candidate from an available set of candidate models to fit a photographed object under the guidance of silhouette correspondence. As shown by Kholgade et al. (2014), the alignments resulting from Xu et al. (2011) are not exact enough to support precise appearance recovery. Su et al. (2014) and Huang et al. (2015) recover the geometry of an object from a single image by leveraging shape collections, an approach that is currently viable only for a limited set of object classes for which such collections are currently available. Furthermore, Huang et al. (2015) fit a model to an image by combining together different parts of the shapes in the collection, which requires the shapes to be segmented. This works well for models of man-made objects, such as furniture, but is not well suited for more general and less regular objects, such as some of the reference objects in our examples.

Wang et al. (2016) develop a pipeline for transporting texture from images of real objects to 3D models of similar objects. Their key assumptions include that the reference object has homogeneous part-level textures and a similar 3D model that has been segmented into parts is available. In contrast, we do not make such assumptions and instead extract an appearance model that may be then applied to a variety of different shapes; furthermore, our model includes medium- and fine-scale geometric deformations in addition to reflectance.

Geometry and appearance of real objects may be captured simultaneously using an RGBD camera. The challenges in such a process are rather different from our scenario: overcoming noisy depth data and imprecise camera poses. These challenges have been tackled by refining the 3D reconstruction using shape-from-shading (see Wu et al. (2014) and Yu et al. (2013)), and by using joint optimization of camera poses and geometry (see Wu et al. (2016) and Zhou and Koltun (2014)). Our setting is different in that we reconstruct from a single image and use a coarse proxy, rather than a sequence of depth maps.

Image-based material editing and modeling. Several methods support replacing the material of an object in an image. Fang and Hart (2004) and Zelinka et al. (2005) synthesize a texture across the surface of an object in an image by using surface normals recovered via shape-from-shading to guide the synthesis. Diamanti et al. (2015) also use example-based texture synthesis to replace the texture of an object in an image but rely on user-provided annotations both inside the target image region and on the texture exemplars. Khan et al. (2006) infer the shape and surrounding lighting of a object in a photograph and render its appearance with altered material. Xue et al. (2008) model the reflectance of weathered surface pixels in a photograph as a manifold and use it for editing the weathering effects in the image. All of these approaches only recover partial aspects of appearance to modify the appearance of a particular object in the context of the original input image, whereas our goal is to extract an appearance model that may then be used to create stand-alone detailed 3D models.

Various tools are also proposed for recovering the microgeometry and reflectance of materials from a single input image. For example, Dischler et al. (2002) and Wang et al. (2003) describe interactive methods for modeling bump and displacement maps. The AppGen system (Dong et al. 2011) enables the user to extract a material (diffuse albedo map, bump map, and a spatially varying specular coefficient) from a single image of a roughly planar surface lit by directional lighting. Our method also recovers appearance models (geometric details at two scales and diffuse albedo map) for different materials, but the only assistance required from the user is to provide a relatively coarse proxy approximating the visible part of an object of interest in the image. Our goals are similar to those of AppGen, but our approach is not limited to nearly planar surfaces, and the extracted nonparametric appearance model may be applied to objects with rather different shapes.

Intrinsic image decomposition. These techniques factor an image into a product of reflectance and shading. The problem is severely ill-posed and thus requires strong assumptions (Horn 1986) or user assistance (Bousseau et al. 2009) to be solved. Barron and Malik (2015) unify intrinsic decomposition and shape-from-shading techniques and recover, from a single image of an object, its shape, diffuse reflectance, and illumination. This decomposition is highly ambiguous, as the same pixel color may be explained by an infinite number of combinations of these three components. Several rather restrictive priors are therefore employed in Barron and Malik (2015). For example, the object shape is assumed to be smooth (bend rarely), the distribution of orientations is assumed to be isotropic, and the normals on the object's contour are assumed to be perpendicular to the view direction. These assumptions often result in significant deviations from the actual object shape, which in turn yields incorrect estimates of the underlying shape and the reflectance across its surface. Furthermore, the shape smoothness prior prevents the faithful extraction of fine-scale geometric details, which are crucial for a realistic appearance of 3D models.

We adopt Barron's approach in our appearance extraction step (Section 5.1). However, instead of relying on general shape priors, we assume that a coarse geometric proxy is available, which can be either modeled very quickly by an experienced modeling software user or obtained from an online 3D repository. We show that after properly aligning and deforming the coarse proxy with the object in the image, we can extract the medium- and fine-scale geometry, as well as the spatially varying albedo map, much more accurately.

Example-based texture synthesis. A complete literature review of example-based texture synthesis methods is outside the scope of this work, and we refer the reader to the excellent survey by Wei et al. (2009). Here, we use nonparametric texture synthesis to synthesize the extracted displacement fields and diffuse reflectance textures on the target proxy. To obtain realistic results, we extend the method of Mertens et al. (2006), where the synthesis is guided by the correlation with a geometric feature field extracted from the source and target shapes. However, in contrast to Mertens et al. (2006), we transfer bi-scale geometric detail in addition to the diffuse reflectance texture. Furthermore, since the initial level of the target shape's geometric detail may differ from that of the source shape, we perform the geometric detail transfer in two subsequent stages, where the result of the first stage enables the correlations used to guide the synthesis during the second stage. Due to our use of state-of-the-art texture optimization techniques in the second stage, our approach is able to cope with more structured and less homogeneous textures, as demonstrated in Figure 1.

Boneel et al. (2010) also apply example-based guided texture synthesis on simple 3D proxies. The goal is to render natural landscape scenes without the needs of creating detailed 3D landscape models. That work is not concerned with capturing and transferring the appearances of real reference objects.

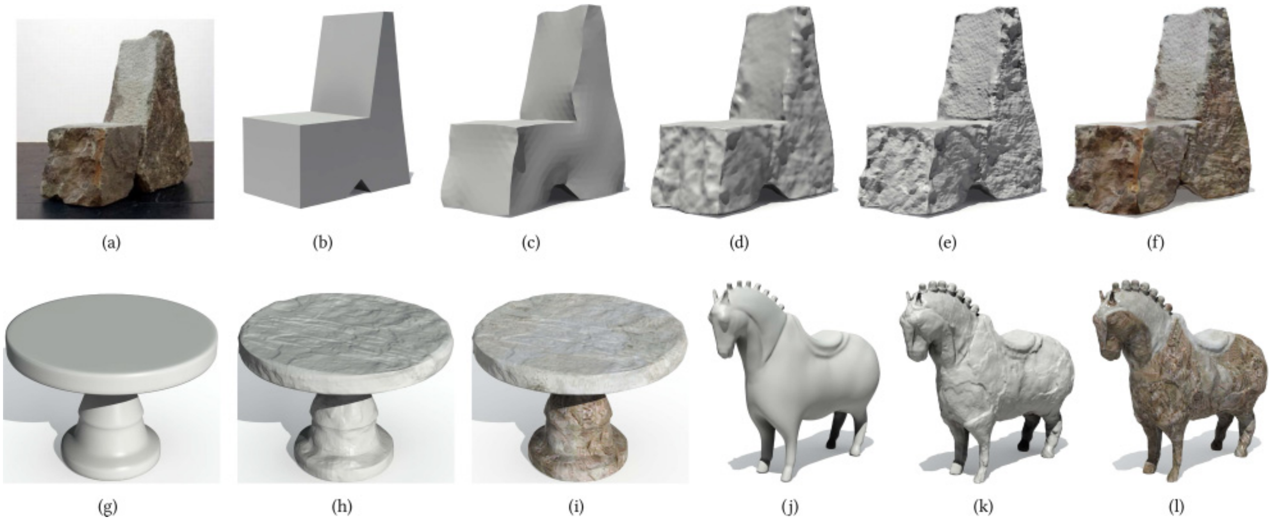

Fig. 2. Given a photograph of a reference object (a) and an initial coarse 3D proxy (b), we first deform the proxy (c) to align with the object in the photo and then extract a medium-scale deformation (d) and a fine-scale displacement (e). These geometric details, together with the extracted diffuse reflectance, form a nonparametric appearance model (f), which can be easily transferred to new 3D shapes in the bottom row: detail-less target shapes (g) and (j), after geometric detail transfer (h) and (k), and after diffuse reflectance transfer (i) and (l).

3 OVERVIEW

The goal of our work is to develop a new tool that would enable users to easily endow a 3D shape with richly detailed realistic appearance extracted from a photograph of a reference object. More specifically, assuming that the reference object exhibits some interesting geometric surface details (deformations and displacements), and a natural color texture (diffuse reflectance map), we would like to learn a nonparametric *appearance model* that captures both, as well as the correlation between the appearance and various higher-level geometric features. Once such a model is learned, it can be applied to new object shapes as well.

As mentioned earlier, we employ a coarse 3D proxy to facilitate the extraction of the appearance model. Figure 2(a) and (b) show a reference object photograph along with a suitable proxy as an example. To make use of the coarse proxy, we must first register and align it with the object image (Figure 2(c)). This is a challenging task, as the proxy can be quite coarse and may have different part scales. Our solution to this problem is our main technical contribution, described in detail in Section 4.

Having aligned and deformed the proxy to better fit the reference object in the image, we apply an enhanced version of Barron and Malik's algorithm (2015) to extract the illumination, the diffuse reflectance, and the depth map of the reference object, as described in Section 5.1. The resulting depth map provides a much more detailed and accurate shape approximation of the object's visible part than the aligned proxy. To make the extracted geometric and photometric details transferable to other models, we further extract a two-stage deformation between the aligned proxy and the detailed depth map: a medium-scale deformation followed by a fine-scale displacement field. The two-scale deformation (Figure 2(d) through (e)) together with the diffuse reflectance constitute our nonparametric appearance model (Figure 2(f)); see Section 5.2.

Finally, Section 6 describes how, given a coarsely modeled target shape, we apply the extracted appearance model to this object, which yields a deformed 3D model with detailed displacement and reflectance maps (Figure 2(g)), thereby completing our modeling pipeline. More examples are shown in Section 7.

4 PROXY ALIGNMENT

To properly guide the appearance extraction using the initial coarse proxy P_{init}, we first need to position and deform P_{init} so that its 2D projection aligns well with the visible part of the reference object in the image. Our alignment process attempts to match the edges of P_{init} with the salient object image edges. To minimize undesirable distortions to P_{init}, a global rigid transformation is applied first, followed by a nonrigid deformation defined by a set of per-vertex displacements. The global transformation is derived from an edge-saliency potential field (Section 4.1) combined with edge point correspondences (Section 4.2). The subsequent nonrigid deformation is obtained via a constrained optimization (Section 4.3).

4.1 Edge-Saliency Potential Field

To align P_{init} with the dominant features in the input image, we first use structured forests (Dollár and Zitnick 2013) to extract an edge map (Figure 3(b)). The edge map assigns an edge response value $s_i \in [0, 1]$ to each pixel p_i, indicating the likelihood that p_i is located on a salient edge.

Since the extracted edges are generally fragmented, directly pairing them with edges of P_{init} is prone to error. To alleviate this problem, we extend the edge map to a scalar edge-saliency potential field (Figure 3(c)) over the image space, such that (1) the field has smaller values (blue) at areas closer to an edge to attract edges of P_{init}, and (2) the more salient an edge is, the smaller the field values in its vicinity so that the proxy can align with dominant

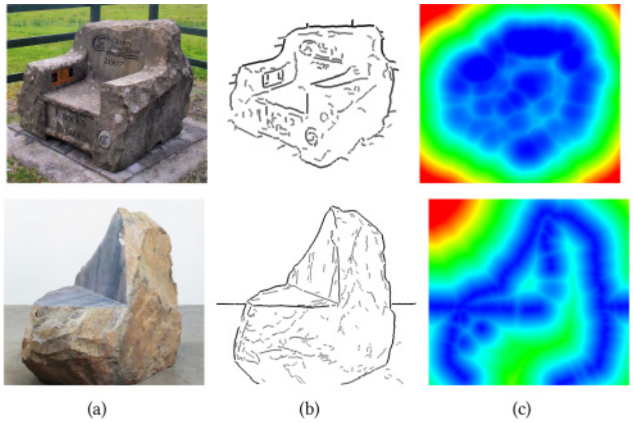

Fig. 3. Given an input image (a), we generate an edge map (b) with edge response values predicted using Dollár and Zitnick (2013), which yields our scalar edge-saliency potential field (c) as defined in Equation (1).

features. Formally, we define the edge-saliency potential field $\mathcal{F}$ at each pixel p_i as

$$\mathcal{F}(p_i) = \min_{j \in E} \left(\|p_i - p_j\|^2 + \omega (1 - s_j)^2 \right)^{\frac{1}{2}}, \tag{1}$$

where E denotes the set containing the indices of edge pixels, and the positions of all pixels are normalized into $[-0.5, 0.5]$. The parameter ω balances the influence of the distance and edge saliency and is set to 0.1 by default.

4.2 Correspondence Search

Our experiments show that the edge potential field can robustly guide the registration of the 3D proxy, especially when the initial alignment is poor (e.g., see Figure 5). However, it is difficult to precisely control the edge-to-edge alignment due to the diffusion of field values. To address this limitation, we augment the field-based alignment with pointwise correspondences. In other words, we first uniformly sample vertices along the sharp edges of P_{init}, which can be easily detected based on local curvatures, especially for coarse CAD models. Next, for each vertex whose projection into the image is visible, we search for its best-matching edge pixel in the nonmaximal suppression edge map (Figure 3(b)). As in our edge-saliency potential field, here the saliency of edge pixels is also used to assist in the matching process, as described in the following.

To automatically compute the optimal point correspondences, an HMM (Kraevoy et al. 2009; Rabiner 1989) is applied. The HMM emission probability is computed using the matching score $\mathcal{S}(\bar{v}_i, p_j)$ between a projected edge vertex $\bar{v}_i$ and an edge pixel p_j:

$$P(\bar{v}_i | p_j) \propto e^{-\frac{1}{2\mathcal{S}^2(\bar{v}_i, p_j)}}. \tag{2}$$

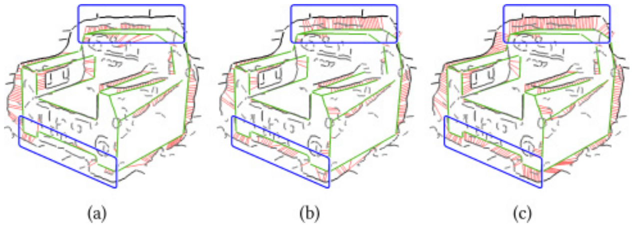

Fig. 4. Comparison of point correspondence strategies: (a) HMM correspondences computed by Kraevoy et al. (2009); (b) HMM correspondences when accounting for edge saliency and distance continuity; (c) our HMM correspondences, accounting for edge saliency, distance continuity, and orientation consistency; see improved correspondences inside blue rectangles.

The matching score should be high if (1) the projected vertex $\bar{v}_i$ is close to p_j, (2) the projected edge orientation at $\bar{v}_i$ and the detected edge orientation at p_j are similar, and (3) the saliency at p_j is high. We hence empirically define the score as

$$\mathcal{S}(\bar{v}_i, p_j) = \frac{s_j^a |\mathbf{t}_i^T \mathbf{t}_j|}{\|\bar{v}_i - p_j\|^b}, \tag{3}$$

where s_j is the saliency at p_j, and the unit vectors $\mathbf{t}_i$ and $\mathbf{t}_j$ denote the orientations of the edges at $\bar{v}_i$ and p_j, respectively. Two constant parameters are set to $a = 0.7$ and $b = 0.5$ by default.

Since the automatically detected image edges are generally noisy and fragmented, both distance continuity and orientation consistency are considered when computing the HMM transition probability:

$$P(p_j | p_{j-1}) \propto e^{-\frac{(1 - d_j/d_i)^2}{2\sigma^2}} e^{-\frac{(1 - \mathbf{t}_i^T \mathbf{t}_j)^2}{2\sigma^2}}, \tag{4}$$

where $d_i = \|\bar{v}_i - \bar{v}_{i-1}\|$, $d_j = \|p_j - p_{j-1}\|$, and $\sigma = 5$ by default.

The HMM problem is solved using the Viterbi algorithm (Rabiner 1989), giving us consistent matches between edge vertices on the proxy and edge pixels in the image (Figure 4).

4.3 Optimizing the Pose and Shape of 3D Proxy

Armed with the edge-saliency potential field and the pointwise correspondences, we can now optimize P_{init}'s pose and shape to achieve the best alignment with the reference object. To evaluate how well P_{init}'s projection aligns with the image edges, the data term accumulates the total field values along the projected visible edges and the sum of distances between the corresponding matching points:

$$E_d(P) = \sum_{i \in V} (\mathcal{F}(\bar{v}_i))^2 + \|\bar{v}_i - \mathcal{M}(v_i)\|^2, \tag{5}$$

where V is the set containing the indices of detected visible edge vertices on P_{init}, and $\mathcal{M}(v_i)$ are the edge pixels corresponding to $\bar{v}_i$, determined as described earlier (Section 4.2).

To best align the proxy P_{init}, we first search for a rigid transformation $\mathcal{T}$ (rotation and translation) that minimizes the preceding data term. Note that here we assume that the input image is captured by a perspective camera with known focal length.[1] Thus, the size of the projection may be adjusted through the distance

[1] For photos taken with unknown focal lengths, a default value is used. As a result, the aligned proxy model may subject to perspective distortion, but this does not have a strong impact on our goal of nonparametric appearance model extraction.

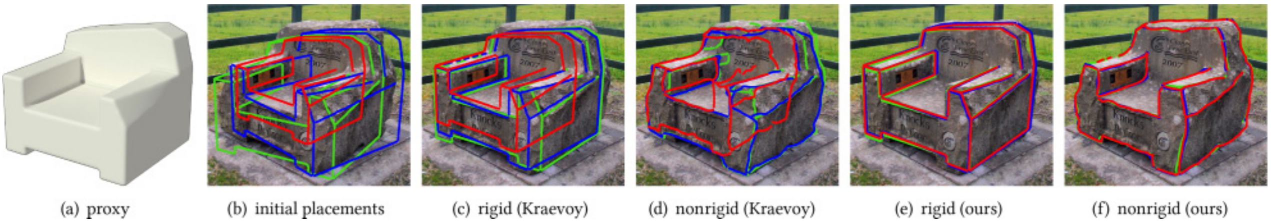

Fig. 5. Optimizing the pose and shape of a coarse 3D proxy (a). Casually placing the proxy with different initial positions and orientations, shown as red, green, and blue wireframes in (b), the resulting rigid (c) and nonrigid (d) transformations based on point correspondences defined in Kraevoy et al. (2009) fail to properly register the proxy. In comparison, our rigid transformations (e) can noticeably improve the proxy registration from different initial placements (b). The alignment is further improved through nonrigid deformation of proxy edges (f).

between P_{init} and the camera. To compute the optimal transformation, we minimize the following objective function:

$$\mathcal{T} = \underset{\mathcal{T}}{\operatorname{argmin}}\, E_d(\mathcal{T}(P)). \tag{6}$$

Once the optimal rigid transformation is found, we further deform P_{init} using nonrigid deformation to match the observed edges. This is done by minimizing both the data term E_d and an as-rigid-as-possible term E_s (Sorkine and Alexa 2007). The latter shape-preserving term is necessary since E_d is only defined on visible edge vertices. The deformation of other vertices is constrained by E_s, which attempts to maintain the original shape of P_{init}:

$$E_s = \sum_{i\in M}\sum_{j\in N(i)} w_{ij}\|(v_i - v_j) - \mathcal{T}_i(v_i' - v_j')\|^2, \tag{7}$$

where $\{v_i\}$ are the deformed 3D vertices of P_{init} and $\{v_i'\}$ are the original untransformed ones, M denotes the set containing all vertex indices in P_{init}, and $N(i)$ the set containing vertices connected to v_i. The transformation $\mathcal{T}_i$ is local within the neighboring set $N(i)$, and w_{ij} is the cotangent weight (Meyer et al. 2003). In addition, we can optionally enforce the flatness of selected planar surfaces by

$$E_p = \sum_{i\in M}\|v_i - Proj(v_i)\|^2, \tag{8}$$

where $Proj(v_i)$ is its projection on the PCA plane computed from all vertices on the planar surface and is updated at each iteration.

The aligned proxy P_{align} is initialized to $\mathcal{T}(P_{init})$ and then further deformed by minimizing the following objective function:

$$P_{align} = \underset{P}{\operatorname{argmin}}\left(E_d(P) + E_s(P) + E_p(P)\right). \tag{9}$$

The preceding optimization is solved using the Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Nocedal and Wright 2006).

4.4 Experimental Validation

We performed several experiments to test the sensitivity of our proxy alignment method to the initial placement (position and orientation) of the proxy. The results of one such experiment is shown in Figure 5. The same proxy was placed in several different positions and orientations (shown in the wireframe in Figure 5(b)). These different initial placements yield nearly the same registered proxy, as shown in Figure 5(e) through (f). In contrast, Kraevoy's method (2009) shows more sensitivity to the initial placement (Figure 5(c) and (d)). Figure 7 shows the results of a quantitative stress test using a different image with nine significantly different initial proxy positions and orientations. Once again, the final nonrigid alignment results are nearly identical, except when the initial placement is extremely poor.

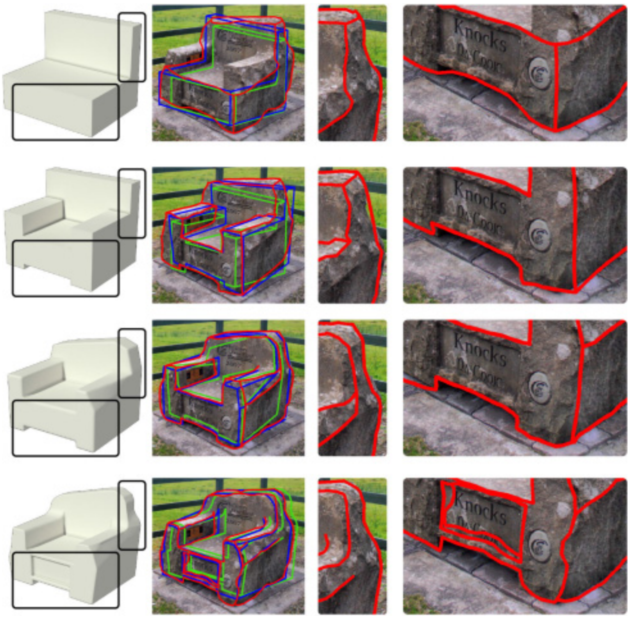

Fig. 6. Aligning an image object with a progressively refined set of 3D proxies (shown in the left column). The initial poses (green), rigid alignments (blue), and the final nonrigid deformations (red) of these proxies are shown in the second column. The zoom-in regions with only the final deformation results are shown in the right two columns for easier examination.

We also tested the sensitivity of our method to the accuracy of the proxy. The method was applied to a sequence of progressively finer proxies of the stone chair (shown in the leftmost column of Figure 6). It may be seen that although the simplest proxy, consisting of merely two boxes, is too coarse, slightly more detailed proxies result in a satisfactory alignment and registration (second and third rows). Interestingly, the most refined proxy (bottom row) yields a less satisfactory result. A quantitative test using a different reference image is shown in Figure 8. In our

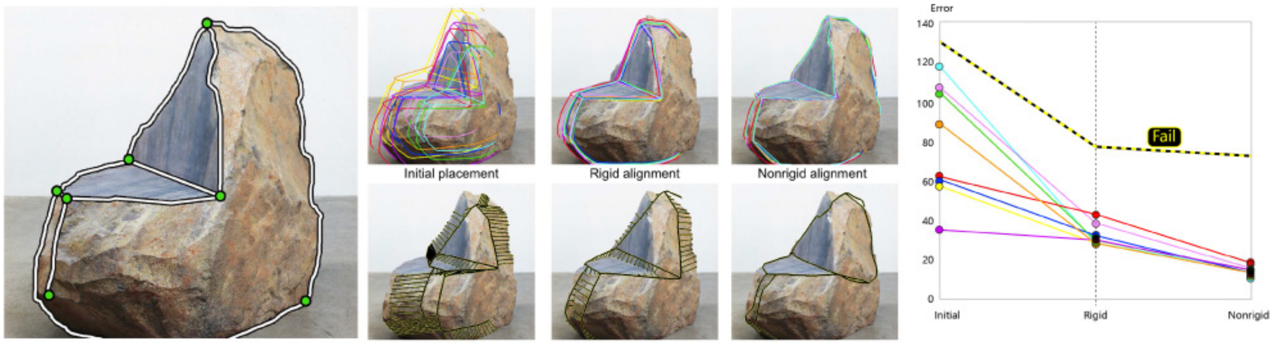

Fig. 7. Stress test for initial proxy placements, which reports the average alignment error for the visible proxy edges (on the right, using pixel units) for various initial proxy placements. The alignment error is computed with respect to manually defined ground truth edges, shown on the left. The qualitative results are shown in the middle with colors that correspond to the error plot lines. One failure case is shown at the second row, where our HMM correspondences are partially wrong due to the extreme deviation of the initial placement (black).

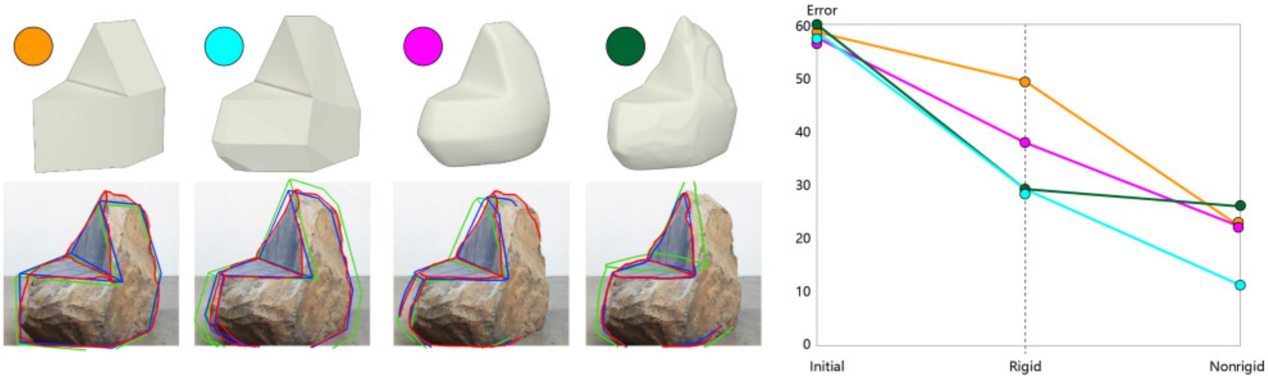

Fig. 8. Aligning an image object with a progressively finer set of 3D proxies (initial poses shown in green). The detected edge map and our computed edge-saliency potential field are shown in the second row of Figure 3, guiding the rigid alignment (shown in blue) and nonrigid deformation (shown in red) of proxies. Even the very coarse initial proxy on the left is successfully aligned. The alignment errors are plotted on the right, using colors corresponding to those of the disks next to each of the four proxies.

experience, for good results, the proxy should capture the main large-scale geometric parts of the object in the image, such as the arm rests of the stone chair. Trying to model geometric features that do not manifest themselves as highly salient edges in the image is unnecessary and, in fact, could prove counterproductive, as evidenced by the results of the finest proxies in Figures 6 and 8.

5 APPEARANCE EXTRACTION

Having a 3D proxy P_{align} aligned with the reference object in image $\mathbf{I}$, we next extract a joint geometric-photometric appearance model, which can be applied to other objects. The process starts with recovering a depth map $\mathbf{Z}$ and a reflectance map $\mathbf{R}$ for the reference object's visible part (Section 5.1). This is followed by the extraction of the appearance model that consists of two parts: (1) a medium-scale deformation field $\mathcal{D}_m$ and (2) a fine-scale displacement field $\mathcal{D}_f$ and the correlated reflectance map $\mathbf{R}$ (Section 5.2).

5.1 Detail Recovery

Following Barron and Malik's SIRFS method (2015), we assume that the object surface in the input photograph is Lambertian and represent the input image as $\mathbf{I} = \mathbf{R} + \mathrm{S}(\mathbf{Z}, \mathbf{L})$, where $\mathbf{I}$ is the log-image of the input, $\mathbf{R}$ is the log-reflectance image, and $\mathbf{S}$ is a shading function that generates the log-shading image of the depth map $\mathbf{Z}$ under the low-frequency illumination $\mathbf{L}$, represented using a small number of spherical harmonics. Accordingly, the depth map $\mathbf{Z}$ and the illumination $\mathbf{L}$ can be computed using the following optimization (Barron and Malik 2015):

$$(\mathbf{Z}, \mathbf{L}) = \underset{\mathbf{Z},\mathbf{L}}{\operatorname{argmin}}\, g(\mathbf{I} - \mathrm{S}(\mathbf{Z}, \mathbf{L})) + f(\mathbf{Z}) + h(\mathbf{L}), \tag{10}$$

where g, f, and h are cost functions or priors for reflectance, geometry, and lighting, respectively.

Differently from SIRFS, which focuses on recovering large-scale geometry and smooth reflectance without a proxy, our method is based on a well-aligned large-scale proxy P_{align} and focuses on recovering reflectance and finer geometric details. We thus apply enhanced cost functions for reflectance and geometry.

Reflectance priors. For reflectance, our cost function $g(\mathbf{R})$ consists of three terms:

$$g(\mathbf{R}) = \lambda_e g_e(\mathbf{R}) + \lambda_a g_a(\mathbf{R}) + \lambda_r g_r(\mathbf{R}), \tag{11}$$

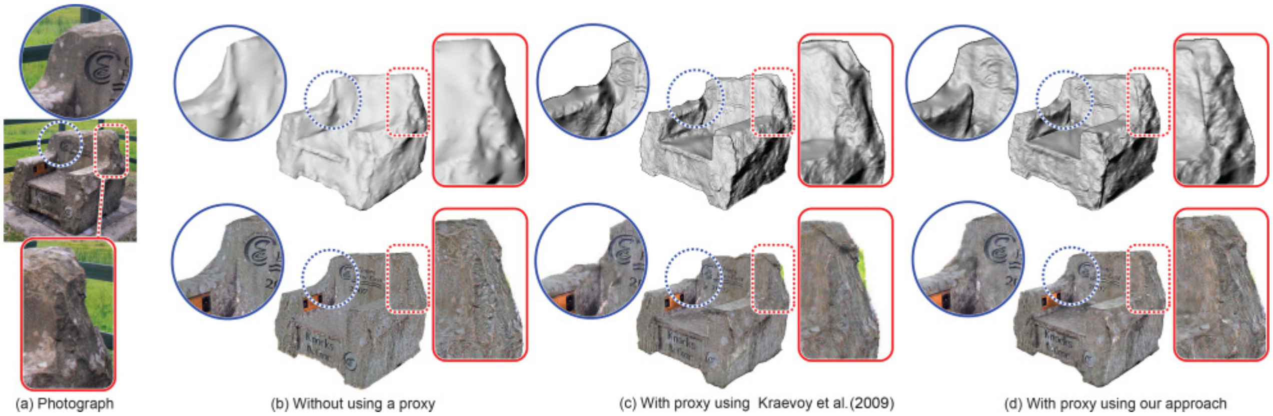
(a) Photograph
(b) Without using a proxy
(c) With proxy using Kraevoy et al.(2009)
(d) With proxy using our approach

Fig. 9. Intrinsic decomposition results (top row: **Z**; bottom row: **R**) obtained using different approaches for an input photograph (a). Without using a proxy, the original SIRFS method outputs overly smooth **Z** and noisy **R** (b). More geometric details are recovered using our approach and a 3D proxy (b, c). Nevertheless, when the imprecise alignment obtained using Kraevoy et al. (2009) (the green one in Figure 5(c), which is the best of the three) is used, artifacts show up along sharp edges in both **Z** and **R** (b). These artifacts are not present in our approach (d); see zoomed-in views for a better comparison.

where g_e and g_a are the parsimony and absolute priors that we inherit from SIRFS with weight parameters λ_e, λ_a, and λ_r. The former expects a small number of different reflectance values in the input image, whereas the latter constrains the reflectance values following a learned model; please refer to Barron and Malik (2015) for details. Here g_r is a novel Retinex prior, which replaces the smooth prior used in SIRFS, as the latter assumes that reflectance is piecewise constant and does not hold for object surfaces with rich textures. Instead, our Retinex prior g_r assumes that in each local region, pixels with similar chromaticity values have similar reflectance. Accordingly, it is defined as

$$g_r(\mathbf{R}) = \sum_i \sum_{j \in N(i)} \alpha(c_i, c_j)||R_i - R_j||^2, \tag{12}$$

where $N(i)$ is a 5×5 window centered at pixel i, c_i, and c_j are chromaticities of pixels i and j, and R_i and R_j are their reflectances. We set the weight function $\alpha(c_i, c_j) = e^{-\|c_i - c_j\|/4}$, which rewards pixel pairs with small chromaticity difference.

Geometry prior. For improved recovery of geometric detail, we dropped the contour, isotropic, and smooth priors proposed in the original SIRFS method, as our proxy prior naturally provides the contour normals and our reference objects typically include rich geometric details. Hence, the cost function is simply defined by

$$f(\mathbf{Z}) = \lambda_p f_p(\mathbf{Z}), \tag{13}$$

where the proxy prior f_p constrains the smoothed version of the reconstructed depth map **Z** to be consistent with P_{align}—for instance:

$$f_p(\mathbf{Z}) = \sum_i ||G(\mathbf{Z}_i, r) - \mathbf{Z}_i(P_{align})||^2, \tag{14}$$

where $G(\mathbf{Z}_i, r)$ is the depth at pixel i after filtering by a Gaussian with radius r, whereas $\mathbf{Z}_i(P_{align})$ is the depth of P_{align} at the same pixel. Our l_2-based proxy prior is slightly different from the l_p-based proxy prior used in SIRFS, as the geometric details in our input photograph are not distributed in a sparse manner.

Optimization. We solve the optimization in (10) with the multiscale solver described in Barron and Malik (2015). The difference of our method is that instead of using a plane as initialization, we sample the proxy geometry as the initial depth map for optimization. In our current implementation, we use the original weight settings ($\lambda_e = 3.36$ and $\lambda_a = 4.75$) for the parsimony and absolute priors inherited from SIRFS, and set $\lambda_r = 5$ and $\lambda_p = 1$ by default for the new priors introduced by our method. For illumination, we follow the same approach as SIRFS and apply the laboratory-like prior for h. After solving for the depth map and the illumination, we compute the log-reflectance image as the difference $\mathbf{R} = \mathbf{I} - \mathbf{S}(\mathbf{Z}, \mathbf{L})$.

Figure 9 compares the intrinsic decomposition by our method to two other alternatives. Note that thanks to the well-aligned proxy P_{align}, our enhanced SIRFS decomposition algorithm successfully recovers the reflectance and geometric details over the surface. Without a proxy (Figure 9(a) and (d)) or with a misaligned proxy (Figure 9(b)), the SIRFS decomposition results contain noticeable artifacts in both recovered **Z** and **R**.

5.2 Bi-Scale Deformation Extraction

Recall that our ultimate goal is not to recover the depth map **Z** and the reflectance map **R** in themselves but rather to transfer the appearance captured by these two maps to other 3D shapes. Hence, our next step is to extract a joint geometric-photometric appearance model based on **Z** and **R**, which consists of both a deformation operator $\mathcal{D}$ and the reflectance map **R**. The deformation operator $\mathcal{D}$ maps the visible part of the aligned proxy to the recovered depth map (i.e., $\mathcal{D}(P_{align}) = \mathbf{Z}$). We model $\mathcal{D}$ as a composition of two operators: $\mathcal{D} = \mathcal{D}_f \circ \mathcal{D}_m$, where $\mathcal{D}_m$ is a medium-scale deformation field and $\mathcal{D}_f$ is a fine-scale displacement field; see Figures 10 and 11.

There are three reasons for decomposing $\mathcal{D}$ into two steps. First of all, since P_{align} only matches the reference object along a few salient edges, the full deformation $\mathcal{D}$ can be very significant; applying such a large deformation directly to a coarse shape

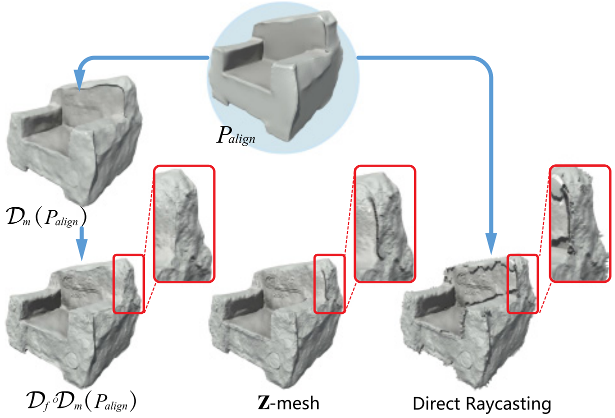
P_{align}
$\mathcal{D}_m(P_{align})$
$\mathcal{D}_f \circ \mathcal{D}_m(P_{align})$
Z-mesh
Direct Raycasting

Fig. 10. Bi-scale deformation extraction. Using the normal map derived from the depth map **Z**, we first extract the medium-scale deformation $\mathcal{D}_m$ (middle left). The final-scale displacement $\mathcal{D}_f$ (bottom left) is then extracted based on the differences between $\mathcal{D}_m(P_{align})$ and **Z**-mesh (bottom middle). In comparison, directly extracting full deformation in a single step results in noticeable artifacts along sharp edges (bottom right).

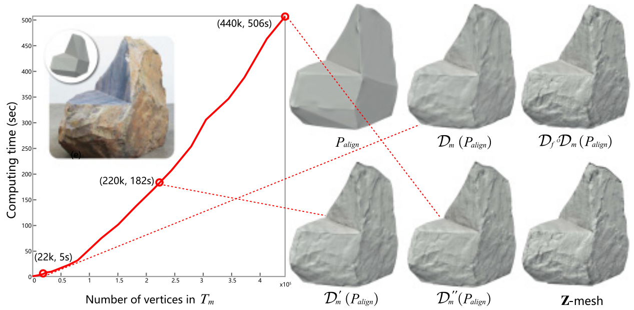

Fig. 11. Our bi-scale deformation $\mathcal{D} = \mathcal{D}_f \circ \mathcal{D}_m$ maps the visible regions of the aligned proxy P_{align} obtained using an initial proxy P_{init} and a reference image (top left) to the recovered depth map (bottom right) such that $\mathcal{D}_f \circ \mathcal{D}_m(P_{align}) \approx \mathbf{Z}$. The plot shows the increase in the computation cost of the deformation $\mathcal{D}_m$ as the tessellation T_m increases, and three of resulting models $\mathcal{D}_m(P_{align})$, $\mathcal{D}'_m(P_{align})$ and $\mathcal{D}''_m(P_{align})$ are shown on the right. Although using finer tessellations T_m increases the cost, the resulting deformations are still not as accurate as our bi-scale deformation (top right).

can result in visible artifacts, especially in the vicinity of sharp edges—for example, see the direct raycasting result in Figure 10 (bottom right). We overcome this by regularizing $\mathcal{D}_m$ with an as-rigid-as-possible constraint to avoid extreme meshing distortions, as described in the following. The resulting deformed shape $\mathcal{D}_m(P_{align})$ is then close to **Z** but missing fine geometric details. These details are then captured by $\mathcal{D}_f$ as a displacement along the normal at each vertex.

Second, transferring the medium-scale deformation field $\mathcal{D}_m$ to the smooth target shape first ensures that the initially smooth target shape now has a sufficient amount of geometric detail across its surface, which in turn enables the subsequent geometry-correlated transfer of fine-scale displacement $\mathcal{D}_f$ and reflectance texture **R** in a joint and correlated manner.

Finally, decomposing $\mathcal{D}$ into medium-scale and fine-scale operators and then applying these two operators separately can dramatically increase the number of possible geometric patterns, making it easier to synthesize rich geometric details on the target shape using only limited exemplars from a single photo.

In the following, we assume that a parameterization, also known as a UV-map, is available for the initial proxy, which is also inherited by the aligned proxy P_{align}. A suitable parameterization can be automatically generated by the modeling software. In all of our experiments, we used the UVLayout software[2] for this purpose. When the reference object consists of several regions, each featuring a different appearance, the proxy should be split (by the user) into several pieces accordingly. For example, the stone chair in Figure 11 can be split into three pieces, as shown in Figure 12, each of which is automatically embedded into the plane by the UVLayout software.

We represent the medium-scale deformation field $\mathcal{D}_m$ using a medium-scale tessellation T_m of the aligned proxy P_{align}. The tessellation is obtained using the midpoint subdivision algorithm. We then deform the resulting mesh by *normal transfer* (Jones et al. 2003). Specifically, we project each visible triangle $t \in T_m$ onto the normal map derived from **Z** and compute the average normal N_t over the projection area. Next, we compute a deformation field $\mathcal{D}_m$ that attempts to match the normals $\{N_t\}$. The deformation is represented as a set of displacement vectors for the vertices of T_m, where each displacement vector is expressed in the local frame at the corresponding vertex. The deformation is regularized using a local as-rigid-as-possible shape preserving constraint, similar to the term E_s defined in Equation (7), to avoid artifacts. Figures 10 and 11 visualize the resulting deformations on two different examples.

The fine-scale displacement map $\mathcal{D}_f$ is then computed based on the difference between $\mathcal{D}_m(P_{align})$ and **Z**-mesh, a mesh created from the depth map **Z**. A second, much finer tessellation T_f is created for this purpose. For each vertex $v \in T_f$, we compute the displacement by casting a ray along the normal at v such that v is displaced to its corresponding location on **Z**-mesh. The density of T_f is thus naturally set as the same of the resolution of the input image. Both the fine-scale displacement map $\mathcal{D}_f$ and the recovered surface reflectance **R** are represented as a four-channel RGBD (RGB + displacement) texture over the UV-map (e.g., see Figure 12).

Our experiments indicate that the tessellation of T_m, for computing the medium-scale deformation, should be neither too sparse (causing obvious meshing artifacts, particularly near sharp edges), nor too dense (causing much higher computation effort); see Figure 11. We have found that computing $\mathcal{D}_m$ directly on the finest tessellation T_f is counterproductive: although the resulting deformation field takes about 10 times longer to compute, the amount and precision of captured geometric detail is lower than with our bi-scale approach; compare the model $\mathcal{D}''_m(P_{align})$ in Figure 11 to the $\mathcal{D}_f \circ \mathcal{D}_m(P_{align})$ result in the top row of Figure 11. Empirically, we set by default the number of vertices in T_m as 5% of that in T_f.

To summarize, to capture the significant deformation between P_{align} and **Z** without introducing unwanted distortions and

[2] https://www.uvlayout.com.

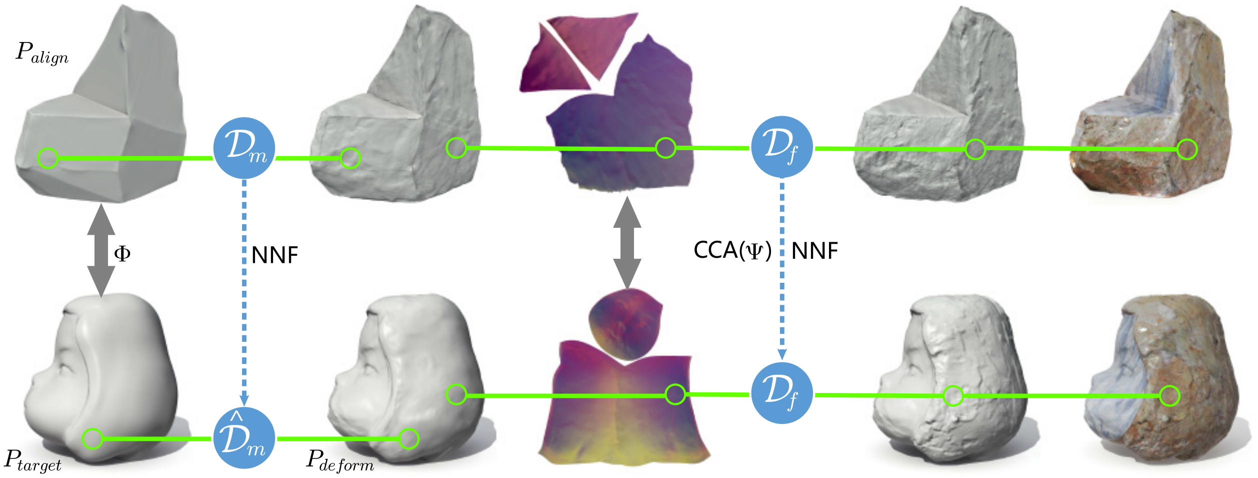

Fig. 12. Appearance transfer. Based on the correspondences between the geometric features Φ of the source proxy P_{align} (top left) and the target shape P_{target} (bottom left), we first transfer the medium-scale deformation to P_{target}. The fine-scale displacements and the reflectances are then synthesized over the UV-map of P_{target} using correspondences between the geometric feature vectors Ψ. Applying the displacement and the reflectance texture yields the final appearance transfer result (bottom right).

artifacts, we do not extract it directly by casting rays but rather compute first the medium-scale $\mathcal{D}_m$ deformation using an as-rigid-as-possible term to keep it well behaved and under control. This regularized deformation, however, cannot extract all of the fine details. Thus, the second, fine-scale displacement mapping $\mathcal{D}_f$ is extracted by casting a ray along each normal.

6 APPEARANCE TRANSFER

Having extracted an appearance model, our goal is now to transfer it to a new *target shape* P_{target} provided by the user. This allows us to add realistic geometric and photometric details to P_{target}. Since P_{target} could be overly smooth or coarse, a two-step process is applied. It first transfers medium-scale geometric deformations (Section 6.1), followed by applying a more detailed displacement field, along with the surface reflectance, using geometry-correlated texture transfer (Section 6.2).

6.1 Medium-Scale Deformation Transfer

Given a target shape P_{target}, we first perform the $\mathcal{D}_m$ deformation transfer in a geometry-correlated manner, inspired by Mertens et al. (2006). We assume that the target shape is also provided with its planar embedding (UV-map), similarly to the aligned proxy. Moreover, we are able to compute the global symmetry plane using the technique proposed in Xu et al. (2009) for both P_{align} and P_{target}, which can guide us to align the two models.

Next, over each of the two models, we compute a 9D geometric feature vector Φ for each vertex that consists of (1) normalized height (1D), (2) projection onto the symmetry plane with symmetry-reflected normal (4D), and (3) symmetry-reflected directional occlusion (4D). We interpolate the per-vertex feature vectors Φ across each of the two UV-maps and then use the PatchMatch algorithm (Barnes et al. 2009) to compute a nearest-neighbor field (NNF) between the two Φ-feature maps. The resulting NNF maps each small patch on P_{target} to one on P_{align} that is the most similar in terms of the aforementioned geometric features. We can thus use the NNF to transfer the corresponding deformation operation, denoted as $\hat{\mathcal{D}}_m(\cdot)$, to target shape P_{target}. See Figure 12 (left) for an illustration. Because of overlap between neighboring patches, multiple $\mathcal{D}_m$ displacement vectors may be mapped to the same vertex of P_{target}, in which case voting takes place to compute a single displacement vector. The entire surface of the target is then deformed by minimizing the energy function:

$$P_{deform} = \underset{P}{\operatorname{argmin}} \left(\|\hat{\mathcal{D}}_m(P_{target}) - P\|^2 + \beta E_s(P) \right), \qquad (15)$$

where E_s is the as-rigid-as-possible term defined in (7), weighted with a parameter β. Larger values of β favor an as-rigid-as-possible deformation, whereas smaller values enable more nonrigid local deformations to be applied. The effect of β is demonstrated in Figure 13. We used $\beta = 1$ to generate our results.

6.2 Displacement and Reflectance Transfer

Having transferred the medium-scale geometric deformation, the resulting deformed target shape P_{deform} now has sufficient amount of geometric detail for performing geometry-correlated transfer of the remaining fine-scale displacements jointly with the reflectance. Recall that the preceding are represented as an RGBD texture over the UV-map of the aligned proxy. Our goal is now to synthesize an RGBD texture over the UV-map of the target shape.

To perform the transfer in a geometry-guided manner, we first construct a guidance field over the reference and target UV-maps. For each UV point, we compute a 13D geometric feature vector Ψ at the corresponding location on the 3D shape. The feature vector Ψ consists of the aforementioned geometric feature vector Φ plus multiscale solid angle curvature (4D) (Mertens et al. 2006). Following Mertens et al. (2006), we also use canonical correlation analysis (CCA) to transform the 13D geometric feature space into a 4D space, where the correlation with the RGBD texture values is maximal.

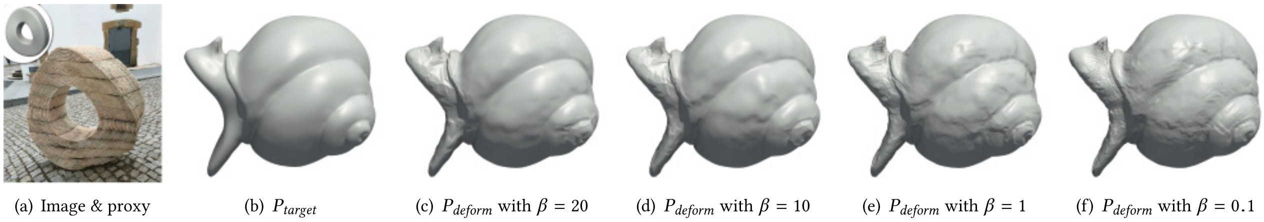
(a) Image & proxy
(b) P_{target}
(c) P_{deform} with $\beta = 20$
(d) P_{deform} with $\beta = 10$
(e) P_{deform} with $\beta = 1$
(f) P_{deform} with $\beta = 0.1$

Fig. 13. The effect of the β parameter in Equation (15) on medium-scale deformation transfer to a target shape P_{target}.

Finally, example-based texture synthesis is carried out using texture optimization (Darabi et al. 2012; Kwatra et al. 2005). We extend the self-tuning texture optimization of Kaspar et al. (2015) by adding the computed geometric guidance fields as a soft constraint. Specifically, we define the distance between a reference patch **s** and a target patch **t** as follows:

$$\hat{d}(\mathbf{r},\mathbf{t}) = \gamma(d_f)d_f(\mathbf{r},\mathbf{t}) + (1-\gamma(d_f))d_t(\mathbf{r},\mathbf{t}) + \mu\Omega(\mathbf{s}), \qquad (16)$$

where d_f is the Euclidean distance between the two vectors formed by concatenating the feature values inside each patch, whereas d_t is the Euclidean distance between two vectors of concatenated RGBD values. The weighting function $\gamma(d_f) = e^{-d_f^2/4}$ is monotonically decreasing with respect to d_f. This means that in areas where the target guidance field is matched well, the patch distance is dominated by d_f, whereas in other areas, it is dominated by d_t. The last *occurrence* term, $\mu\Omega(\mathbf{s})$, is added to discourage repetitions, as proposed in Kaspar et al. (2015): each exemplar pixel has an occurrence count, and each time a patch is selected as the best match, the occurrence count of each pixel inside the patch is incremented. $\Omega(\mathbf{s})$ is set to the sum of the occurrence counts of all pixels inside **s**, and in our results, we set the default weight of this term to $\mu = 0.01$. See Figure 12 (right) for an illustration.

7 RESULTS

The proposed approach was implemented and successfully used to extract appearance models for a variety of materials from single photos. Specifically, as simple proof-of-concept, we have constructed a small appearance library containing five material categories: stone, metal, wood, fabric, and bread, as shown in the top row of Figure 14. We then applied the extracted appearance models to a variety of target shapes, as shown in Figures 1, 2, 14, 15, 16, and 17.

These results demonstrate that our approach successfully extracts the complex geometric details from different photos, such as the vertical ridges on the fire hydrant in Figure 1 and the rough surfaces of the stone chair in Figure 2. These geometric details, together with the corresponding coherent reflectance maps, form easy-to-use appearance models. Once applied to detail-less target shapes, these appearance models can effectively endow these shapes with realistic geometric and photometric details. Furthermore, applying different appearance models to the same target shape can produce quite different medium- and fine-scale geometric details, as can be well seen by examining each of the rows in Figure 14. Also compare the zoomed-in views of the duck model in Figures 15 and 17.

Table 1. The Time Our Modeler Used to Model Shapes Manually (e.g., 15 Minutes for the Fire Hydrant in Figure 1 and 10 Minutes for the Chair and the Table in Figure 2(b) and (g)

Models	Figure 1	Figure 2	Figure 5	Figure 11	Targets
Time	15m	10m	12m	4m	5–15m

Note: In general, it takes our modeler between 5 and 15 minutes to create the detail-less target shapes shown in this article.

The appearance models shown in Figure 14 are intended to serve as a proof-of-concept that a diverse and rich appearance library could be constructed by individual modelers, as well as by the modeling community as a whole. Each appearance model in such a library need only to be extracted once, and then it may be used by different users on many new shapes. For this reason, even in cases where a suitable 3D proxy for the reference object is nontrivial to model, this one-time modeling cost (e.g., see Table 1) will be amortized as the extracted appearance is applied to new shapes. Note that the same modeling effort must also be spent in the traditional modeling workflow, before the modeler can even begin the time-consuming process of creating the fine surface detail and texturing the model. In our approach, this latter time-consuming modeling stage is replaced by automatically applying one of the appearance models from the library. The entire computational process of proxy alignment, detail extraction, and appearance modeling generally takes about 15 minutes, and the average appearance transfer time for each model is less than 10 minutes. In comparison, manually creating such detailed 3D models by manipulating the meshes and applying suitable textures can be extremely tedious and time consuming.

Even though our approach is designed to automate the process of geometric detail modeling, it does provide users with some degrees of control over the modeling results. When the reference object contains regions with different appearances, such as the black hole on fire hydrant (Figure 1), the flat surfaces on the stone chair (Figure 12), and the white top on the wood stool (Figure 17), the user can designate different parts on the target shape and assign a different appearance to each part. Each designated part is parameterized automatically using the UVLayout software. Naturally, different parts of a target shape may also have completely different appearances applied to them, as demonstrated in Figure 16. The magnitude of displacements can also be tuned when deforming the target shapes, yielding surfaces with different levels of roughness; see Figure 15.

Limitations. To maximize the usability of the proposed approach, we constrain ourselves to modeling appearance from sin-

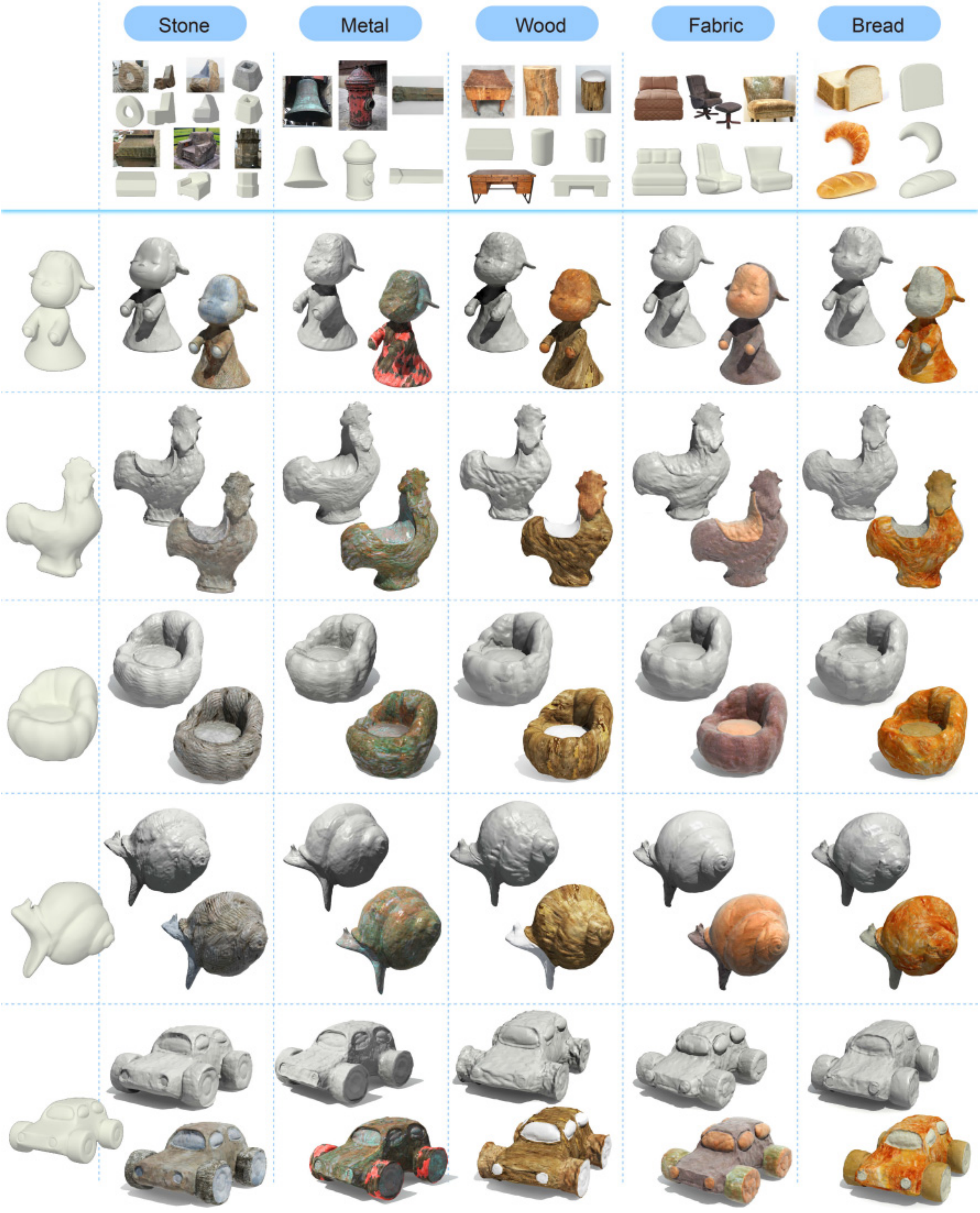

Fig. 14. Using single input photos and coarse 3D proxies (top), we construct a small library of appearance models for five different categories of materials. This allows users to easily add photorealistic details to target shapes and experiment with different appearances. Although bumpiness is introduced to all target shapes, close inspection shows that the character of the bumps is quite different among the different materials. Note that most resulting shapes utilize at least two appearance models extracted from different photos for their different parts. The different parts, and their assigned appearance, are indicated by the user.

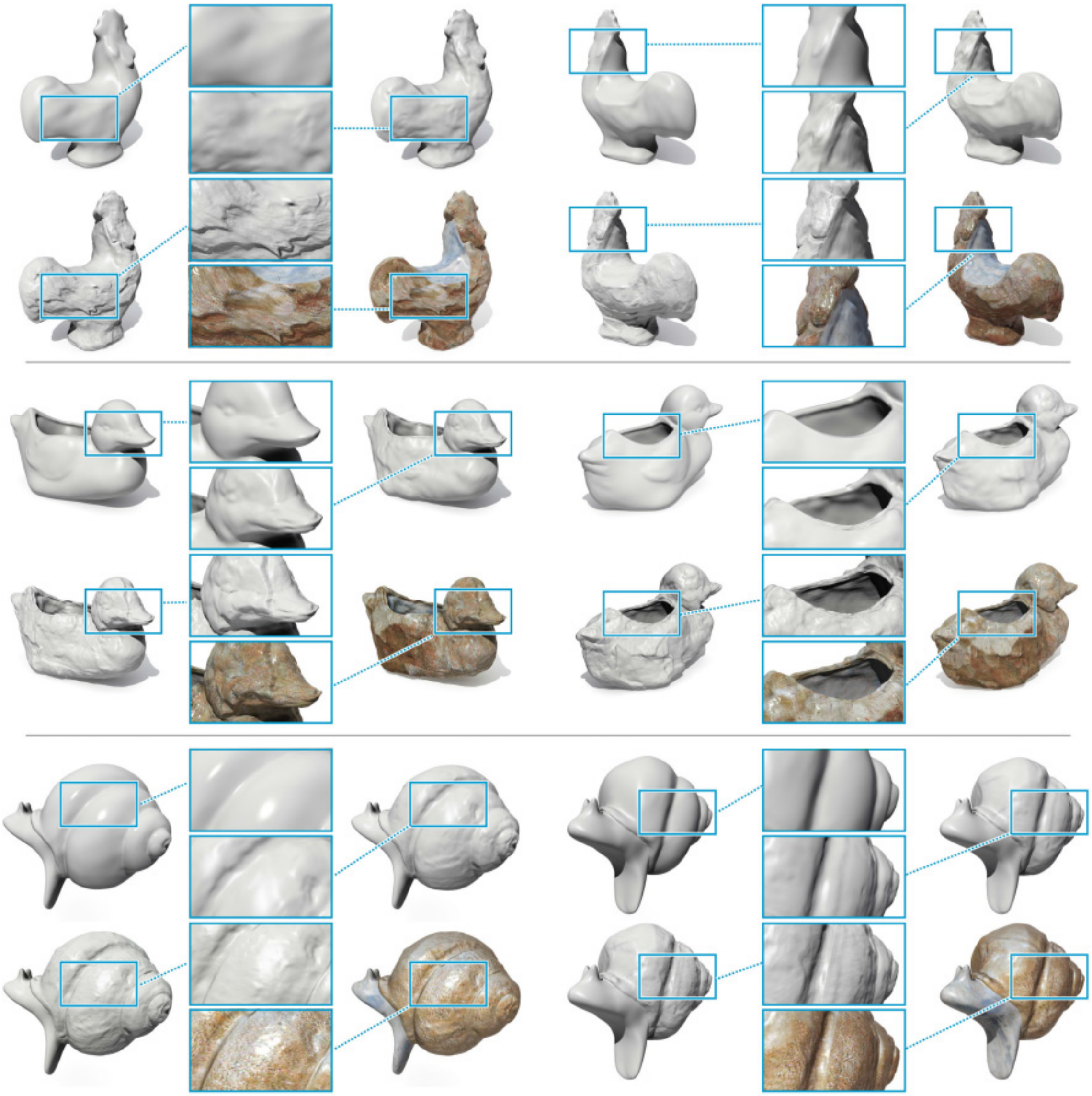

Fig. 15. Adjustment of displacement magnitude during appearance transfer. The stone appearance model extracted from Figure 11 is applied to different target shapes with different displacement magnitude settings. The resulting surface detail can therefore be rougher (rooster) or smoother (snail). For each shape, four models are shown: the user-provided proxy P_{target} (top left), the deformed model P_{deform} (top right), the final displaced geometry $\mathcal{D}_f(P_{deform})$ (bottom left), and the texture mapped result (bottom right).

gle input photos. Even though we adopt a state-of-the-art approach (Barron and Malik 2015) and further enhance it using the aligned 3D proxy, there is still some ambiguity between geometry $\mathbf{Z}$ and reflectance $\mathbf{R}$. Thus, some variations in albedo may be captured as geometric deformations, whereas some geometric details may be captured as changes in the albedo. Figure 18 shows such an example. Although our proxy alignment process successfully deforms the initial proxy to match the photo, the depth map extracted using Equation (10) fails to capture the indented radial wood growth rings. Thus, rather than being captured as a geometric detail, the rings become part of the reflectance texture instead.

8 CONCLUSIONS AND FUTURE WORK

With today's interactive modeling tools and 3D repositories, it is easy to create or find simple 3D object models. However, traversing "the last mile" between these simple, sterile looking models and richly detailed realistic looking ones can be a daunting task.

In this work, we have addressed this challenging stage of the modeling pipeline via proxy-based appearance extraction from a single image and geometry-correlated transfer of the extracted appearance onto new shapes. Given a photograph of an object possessing the desired appearance, we have shown how to extract both geometric and photometric surface details by registering the

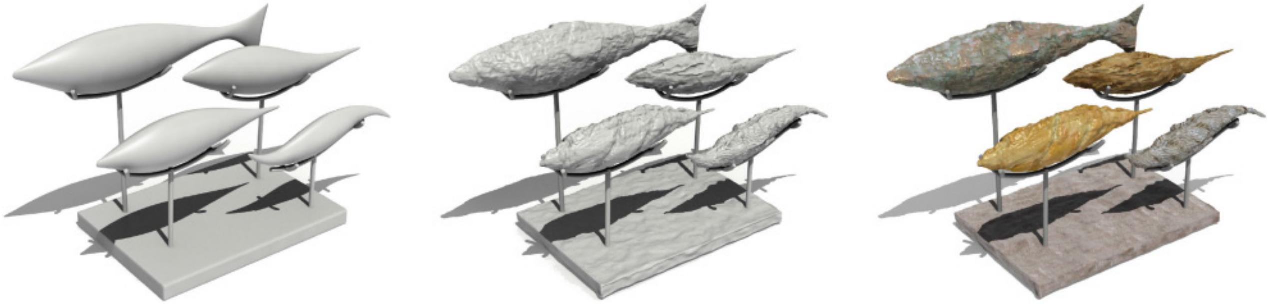

Fig. 16. A simple input scene (left) is enriched using appearance models extracted from photos of different materials (refer to Figure 14). From top to bottom, the four fish models have metal, wood, bread, and stone appearances applied, respectively. The base has fabric appearance applied.

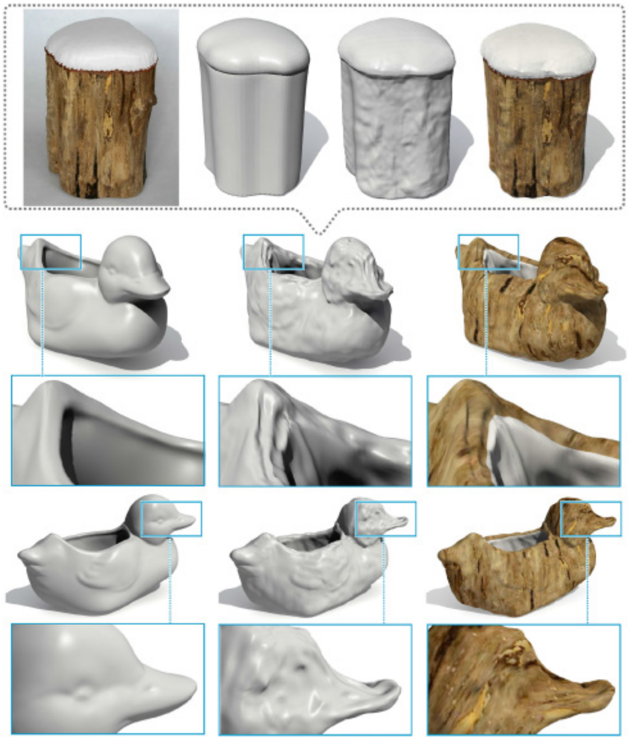

Fig. 17. Applying multiple appearance models extracted from a single reference object image (wood stool) to different parts of a target shape (duck model). The white top material is applied to the interior of the duck model.

proxy with the image. We have demonstrated that once the proxy is deformed and aligned with the reference object in the input photo, the large-scale geometry information that it carries can greatly assist the recovery of middle-scale and fine-scale surface details. The separation of geometric details into two scales assists us in performing the transfer of the finer-scale geometric and photometric details in a geometry-correlated fashion. Experimental results demonstrate that the proposed algorithm can effectively extract photorealistic geometric details from different types of materials and convincingly transfer them to various target shapes.

In future work, we plan to address some of the limitations of our current approach. First, the geometry and appearance extraction (using our modification of Barron and Malik's approach (2015)) should be made more robust by automatic tuning of the parameters so as to perform best of a given shape. Next, the current approach assumes diffuse reflectance illuminated by low-frequency illumination. We would like to extend the approach to handle more general reflectance models and more directional illumination. Having a fairly good approximation of the object's shape should help us cope with effects, such as self-shadowing from directional light sources, as well as model ambient occlusion of the low-frequency illumination. Having a better illumination model will, in turn, enable more accurate geometry and reflectance reconstruction.

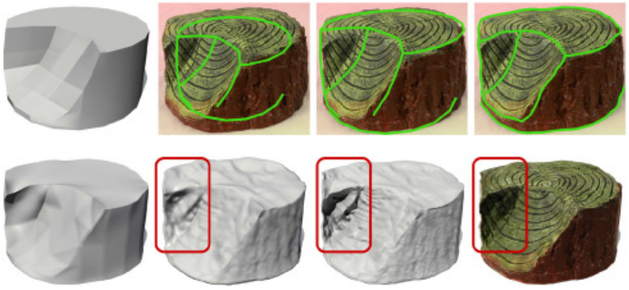

Fig. 18. A failure case, where the geometric details extracted (bottom row) do not correctly capture the ones on the reference object, although the proxy alignment process is very successful (top row). Specifically, the indented growth rings are not part of the recovered geometry, being captured as reflectance details instead. Furthermore, in highlighted areas with red boxes, the extracted surfaces appear to be overly rough.

It would be interesting to attempt to further automate our approach by automatic recovery of candidate proxy shapes from 3D shape repositories, at least for man-made objects. In addition, we would like to leverage Internet image collections to construct a rich appearance model library. We believe that it can be a valuable tool for artists when making their 3D creations.

ACKNOWLEDGMENTS

We thank the anonymous reviewers for their valuable comments.

REFERENCES

Connelly Barnes, Eli Shechtman, Adam Finkelstein, and Dan B. Goldman. 2009. PatchMatch: A randomized correspondence algorithm for structural image editing. *ACM Transactions on Graphics* 28, 3, 24:1–24:11.

Jonathan T. Barron and Jitendra Malik. 2015. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37, 8, 1670–1687.

Nicolas Bonneel, Michiel Van de Panne, Sylvain Lefebvre, and George Drettakis. 2010. Proxy-guided texture synthesis for rendering natural scenes. In *Proceedings of the Conference on Vision, Modeling, and Visualization.*

Adrien Bousseau, Sylvain Paris, and Frédo Durand. 2009. User-assisted intrinsic images. *ACM Transactions on Graphics* 28, 5, 130:1–130:10.

Tao Chen, Zhe Zhu, Ariel Shamir, Shi-Min Hu, and Daniel Cohen-Or. 2013. 3-sweep: Extracting editable objects from a single photo. *ACM Transactions on Graphics* 32, 6, 195:1–195:10.

Soheil Darabi, Eli Shechtman, Connelly Barnes, Dan B. Goldman, and Pradeep Sen. 2012. Image melding: Combining inconsistent images using patch-based synthesis. *ACM Transactions on Graphics* 31, 4, 82–91.

Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. 1996. Modeling and rendering architecture from photographs: A hybrid geometry-and image-based approach. In *Proceedings of the 1996 SIGGRAPH Conference.* 11–20.

Olga Diamanti, Connelly Barnes, Sylvain Paris, Eli Shechtman, and Olga Sorkine-Hornung. 2015. Synthesis of complex image appearance from limited exemplars. *ACM Transactions on Graphics* 34, 2, 22:1–22:14.

J.-M. Dischler, K. Maritaud, and D. Ghazanfarpour. 2002. Coherent bump map recovery from a single texture image. In *Proceedings of the Canadian Conference on Graphics Interface.* 201–208.

Piotr Dollár and C. Lawrence Zitnick. 2013. Structured forests for fast edge detection. In *Proceedings of the International Conference on Computer Vision.* 1841–1848.

Yue Dong, Xin Tong, Fabio Pellacini, and Baining Guo. 2011. AppGen: Interactive material modeling from a single image. *ACM Transactions on Graphics* 30, 6, 146:1–146:10.

Hui Fang and John C. Hart. 2004. Textureshop: Texture synthesis as a photograph editing tool. *ACM Transactions on Graphics* 23, 3, 354–359.

Berthold Horn. 1986. *Robot Vision.* MIT Press, Cambridge, NY.

Qixing Huang, Hai Wang, and Vladlen Koltun. 2015. Single-view reconstruction via joint analysis of image and shape collections. *ACM Transactions on Graphics* 34, 4, 87:1–87:10.

Thouis R. Jones, Frédo Durand, and Mathieu Desbrun. 2003. Non-iterative, feature-preserving mesh smoothing. *ACM Transactions on Graphics* 22, 3, 943–949.

Alexandre Kaspar, Boris Neubert, Dani Lischinski, Mark Pauly, and Johannes Kopf. 2015. Self tuning texture optimization. *Computer Graphics Forum* 34, 2, 349–359.

Erum Arif Khan, Erik Reinhard, Roland W. Fleming, and Heinrich H. Bülthoff. 2006. Image-based material editing. *ACM Transactions on Graphics* 25, 3, 654–663.

Natasha Kholgade, Tomas Simon, Alexei Efros, and Yaser Sheikh. 2014. 3D object manipulation in a single photograph using stock 3D models. *ACM Transactions on Graphics* 33, 4, 127:1–127:12.

Vladislav Kraevoy, Alla Sheffer, and Michiel van de Panne. 2009. Modeling from contour drawings. In *Proceedings of the Eurographics Symposium on Sketch-Based Interfaces and Modeling.* 37–44.

Vivek Kwatra, Irfan Essa, Aaron Bobick, and Nipun Kwatra. 2005. Texture optimization for example-based synthesis. *ACM Transactions on Graphics* 24, 3, 795–802.

Tom Mertens, Jan Kautz, Jiawen Chen, Philippe Bekaert, and Frédo Durand. 2006. Texture transfer using geometry correlation. In *Proceedings of the Eurographics Symposium on Rendering*, Vol. 273. 273–284.

Mark Meyer, Mathieu Desbrun, Peter Schröder, and Alan H. Barr. 2003. Discrete differential-geometry operators for triangulated 2-manifolds. In *Visualization and Mathematics III.* Springer, 35–57.

Jorge Nocedal and Stephen J. Wright. 2006. *Numerical Optimization* (2nd ed.). Springer-Verlag, New York, NY.

Byong Mok Oh, Max Chen, Julie Dorsey, and Frédo Durand. 2001. Image-based modeling and photo editing. In *Proceedings of the 2001 SIGGRAPH Conference.* 433–442.

Manuel M. Oliveira. 2002. Image-based modeling and rendering techniques: A survey. *RITA—Revista de Informática Teórica e Aplicada* 9, 2, 37–66.

Lawrence R. Rabiner. 1989. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE* 77, 257–286.

Konstantinos Rematas, Chuong Nguyen, Mario Fritz, and Tinne Tuytelaars. 2017. Novel views of objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 39, 8, 1576–1590.

Olga Sorkine and Marc Alexa. 2007. As-rigid-as-possible surface modeling. *Computer Graphics Forum* 4, 109–116.

Hao Su, Qixing Huang, Niloy J. Mitra, Yangyan Li, and Leonidas Guibas. 2014. Estimating image depth using shape collections. *ACM Transactions on Graphics* 33, 4, 37:1–37:11.

Tuanfeng Y. Wang, Hao Su, Qixing Huang, Jingwei Huang, Leonidas Guibas, and Niloy J. Mitra. 2016. Unsupervised texture transfer from images to model collections. *ACM Trans.actions on Graphics* 35, 6, 177:1–177:13.

Xi Wang, Lifeng Wang, Ligang Liu, Shimin Hu, and Baining Guo. 2003. Interactive modeling of tree bark. In *Proceedings of the Pacific Conference on Computer Graphics and Applications.*83–90.

Li-Yi Wei, Sylvain Lefebvre, Vivek Kwatra, and Greg Turk. 2009. State of the art in example-based texture synthesis. In *Eurographics State of the Art (EG-STAR) Reports.* 93–117.

Chenglei Wu, Michael Zollhöfer, Matthias Nießner, Marc Stamminger, Shahram Izadi, and Christian Theobalt. 2014. Real-time shading-based refinement for consumer depth cameras. *ACM Transactions on Graphics* 33, 6, 200:1–200:10.

Hongzhi Wu, Zhaotian Wang, and Kun Zhou. 2016. Simultaneous localization and appearance estimation with a consumer RGB-D camera. *IEEE Transactions on Visualization and Computer Graphics* 22, 8, 2012–2023.

Kai Xu, Hao Zhang, Andrea Tagliasacchi, Ligang Liu, Guo Li, Min Meng, and Yueshan Xiong. 2009. Partial intrinsic reflectional symmetry of 3D shapes. *ACM Transactions on Graphics* 28, 5, 138:1–138:10.

Kai Xu, Hanlin Zheng, Hao Zhang, Daniel Cohen-Or, Ligang Liu, and Yueshan Xiong. 2011. Photo-inspired model-driven 3D object modeling. *ACM Transactions on Graphics* 30, 4, 80:1–80:10.

Su Xue, Jiaping Wang, Xin Tong, Qionghai Dai, and Baining Guo. 2008. Image-based material weathering. *Computer Graphics Forum* 27, 2, 617–626.

Lap-Fai Yu, Sai-Kit Yeung, Yu-Wing Tai, and Stephen Lin. 2013. Shading-based shape refinement of RGB-D images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* 1415–1422.

Steve Zelinka, Hui Fang, Michael Garland, and John C. Hart. 2005. Interactive material replacement in photographs. In *Proceedings of the Canadian Conference on Graphics Interface.* 227–232.

Youyi Zheng, Xiang Chen, Ming-Ming Cheng, Kun Zhou, Shi-Min Hu, and Niloy J. Mitra. 2012. Interactive images: Cuboid proxies for smart image manipulation. *ACM Transactions on Graphics* 31, 4, 99:1–99:11.

Qian-Yi Zhou and Vladlen Koltun. 2014. Color map optimization for 3D reconstruction with consumer depth cameras. *ACM Transactions on Graphics* 33, 4, 155:1–155:10.

Received November 2016; revised September 2017; accepted October 2017