DIGITAL ACCESS TO SCHOLARSHIP AT HARVARD

Rendering Deformable Surface Reflectance Fields

(Article begins on next page)

Rendering Deformable Surface Reflectance Fields

Tim Weyrich, Hanspeter Pfister, *Senior Member*, *IEEE*, and Markus Gross, *Senior Member*, *IEEE*

Abstract—Animation of photorealistic computer graphics models is an important goal for many applications. Image-based modeling has emerged as a promising approach to capture and visualize real-world objects. Animating image-based models, however, is still a largely unsolved problem. In this paper, we extend a popular image-based representation called surface reflectance field to animate and render deformable real-world objects under arbitrary illumination. Deforming the surface reflectance field is achieved by modifying the underlying impostor geometry. We augment the impostor by a local parameterization that allows the correct evaluation of acquired reflectance images, preserving the original light model on the deformed surface. We present a deferred shading scheme to handle the increased amount of data involved in shading the deformable surface reflectance field. We show animations of various objects that were acquired with 3D photography.

Index Terms—Computer graphics, image-based rendering.

✦

1 INTRODUCTION

V ISUALIZATION and animation of realistic 3D computer graphics models are important for many applications, such as computer games, movies, advertisement, virtual environments, or e-commerce. Broadly speaking, there are three approaches to reproducing the visual appearance of real objects: explicit modeling with parametric representations, pure image-based approaches, and hybrid approaches that use a combination of both. In all cases, the reflectance properties of 3D objects are typically captured with methods of 3D photography.

For explicit appearance models, parametric BRDF models are fit to the acquired data. Parametric BRDFs can be efficiently rendered on modern graphics hardware and the underlying geometry can be animated and deformed using well-developed techniques such as skinning and vertex blending. However, parametric BRDFs cannot capture many effects of real-world materials, such as translucency, interreflections, self-shadowing, and subsurface scattering. Pure image-based techniques, on the other hand, are well-suited to acquiring and representing complex object appearance. However, most of them impose restrictions on the viewpoints and the lack of a 3D geometry model makes deformations very difficult or impossible.

Consequently, hybrid approaches have become very popular. They parameterize an image-based model on an impostor geometry that can be used for animation.

This representation is commonly known as a surface light field, introduced by Miller et al. [23]. Wood et al. [31] presented a simple technique for its animation. However, surface light fields can only capture an object under fixed

illumination. This is a severe limitation if the object is rendered in a new environment or under dynamically changing lights. A more general hybrid representation is the *surface reflectance field*, which captures the object appearance for many possible light configurations. Objects with arbitrary reflectance properties can be rendered from any viewpoint under new illumination. To date, however, there has been no publication on the animation of surface reflectance fields.

In general, animating an image-based or hybrid appearance representation requires a scheme to evaluate the image-based data set to simulate varying object deformations. A key question is how to preserve the visual appearance of the deformed object surface, that is, how to preserve the perceptual impression of material properties under different lighting conditions.

In this paper, we present a method for animating surface reflectance fields with arbitrary geometric deformations. We develop a shading scheme that aims to preserve the appearance of object materials during deformation. Our method uses a local parmeterization of the impostor geometry that enables arbitrary warps. Our shading method uses this local parameterization to approximately preserve the spatially varying BRDFs of the undeformed object. We present a cache-optimized shading strategy to minimize computation time. Our technique is applicable to other hybrid representations that contain an impostor geometry.

2 PREVIOUS WORK

Three-dimensional (3D) photography loosely describes methods that capture object shape and appearance from images. Some 3D photography approaches fit a parametric BRDF model to the acquired appearance data [28], [32], [14], [13], [22], [16]. However, parametric BRDFs cannot represent many of the reflectance properties of real-life objects [11]. In contrast, image-based appearance representations (with geometry) make no assumptions about the reflection property of materials.

- *T. Weyrich and M. Gross are with the Computer Graphics Laboratory, ETH Zentrum, 8092 Zurich, Switzerland.*
  E-mail: {weyrich, gross}@inf.ethz.ch.
- *H. Pfister is with MERL—Mitsubishi Electric Research Laboratories, Cambridge, MA 02139. E-mail: pfister@merl.com.*

The lumigraph [10] is the first method to combine an impostor geometry (a visual hull of the object) with an image-based representation. Images of the object are stored as a dense light field [17] and the impostor geometry is used to improve light field interpolation, i.e., to minimize blooming and ghosting. To reduce the amount of image data, view-dependent texture mapping [27], [8], [7] uses simple geometry and sparse texture data. This method is extremely effective despite the approximate 3D shape, but it has some limitations for highly specular surfaces due to the relatively small number of textures.

Surface light fields [23], [31], [3] are a more general and efficient representation because they parameterize the image data onto the object surface. Surface light fields can either be stored on accurate high-density geometry [31] or on coarse triangular meshes for objects with low geometric complexity [24]. Some techniques [25], [3] agressively compress the data such that the models can be rendered in real-time on modern graphics hardware. To improve the appearance of complex object silhouettes, surface light fields can be combined with view-dependent opacity data into *opacity light fields* [29]. Unstructured lumigraph rendering [2] is a very effective method for rendering both surface and opacity light fields.

Although surface light fields are capable of reproducing important global effects such as interreflections and self-shadowing, they only show the object under fixed lighting. To overcome this limitation, recent approaches have used surface reflectance fields. The reflectance field of an object is the radiant light from a surface under every possible incident field of illumination. In practice, the reflectance field is sampled sparsely and interpolated during rendering. To further reduce the amount of data, most reflectance fields are acquired for a single view [5], [11], [15]. For approximately planar geometry and diffuse surfaces, the data can be compressed further by fitting a parametric function to the reflectance field [18].

In our work, we use the surface reflectance field data acquired by the 3D photography system of Matusik et al. [20]. The system acquires reflectance fields for over 400 views using cameras, turntables, and a rotating array of lights. The acquired data also includes view-dependent opacity information, called the *opacity reflectance field* (not considered in this paper). The impostor geometry is the visual hull of the objects.

Although it is an important aspect for many practical applications, the animation of image-based data has received very little attention in the literature. Wood et al. [31] describe arbitrary deformations on a surface light field and produce plausible renderings of the deformed model. However, their method does not deal properly with the diffuse component of the surface color and it only works for purely reflective isotropic BRDFs. Feature-based light field morphing [33] morphs two light fields into each other, based on the concept of ray-correspondencies. The method requires substantial user input to specify corresponding feature polygons between the two objects and it is not applicable to the general animation setting. Both methods work only for static illumination. Furukawa et al. [9] presented a scanning system to capture objects and spatially varying BRDFs, also called Bidirectional Texture Functions (BTFs) [4]. They use tensor product expansion to compress the BTF data and show results with surface deformations.

Fig. 1. Example reflectance image.

Their system is the first to render deformations of a relightable, image-based object representation. However, they rely on a tight impostor geometry to support the BTF representation. Thus, the object geometry has to be acquired with a range scanning device. Moreover, their paper does not explicitly address appearance preservation under nonuniform, skewed deformations.

In this paper, we describe an animation method for surface reflectance fields mapped onto *approximate geometry*. Our method allows us to place the objects in new environments with arbitrary illumination, including dynamically changing lights. We carefully analyze the conditions that have to be met to preserve the appearance of the object (Section 4.1) and we present a novel method to approximately preserve spatially varying BRDFs during deformations (Section 4.2). We discuss the limitations of our approach (Section 4.4) and show results using objects that are difficult to handle for image-based approaches, including objects with specularities, transparency, and self-shadowing (Section 6).

3 OVERVIEW

A surface reflectance field consists of a large set of image-based reflectance data of an object, in conjunction with an impostor geometry used for rendering. The reflectance data is given by a collection of *reflectance images*. A reflectance image is a stack of camera images showing the object from the same viewpoint under varying directional illumination (Fig. 1). A surface reflectance field consists of reflectance images from many viewpoints around the object. For our objects, we use approximately 400 reflectance images, each with 60 high dynamic range images of the object under different directional illumination.

Rendering a surface reflectance field can be understood as a two step procedure. First, the reflectance images are used to compute an image of the object under the new illumination for each viewpoint (Fig. 2). These images are then rendered together using the impostor geometry with unstructured lumigraph interpolation [2]. Our impostor geometry is the visual hull of the object that can easily be determined from observed silhouette images. However, our method is independent of the choice of impostor geometry.

We animate and deform a surface reflectance field (SRF) by first applying a 3D warp to the impostor geometry. In order to preserve the warped object's appearance, it becomes necessary to blend the reflectance images invidiually for each point of the impostor geometry. We developed a new look-up function to evaluate the reflectance images of the warped SRF. This function depends on the warp and is
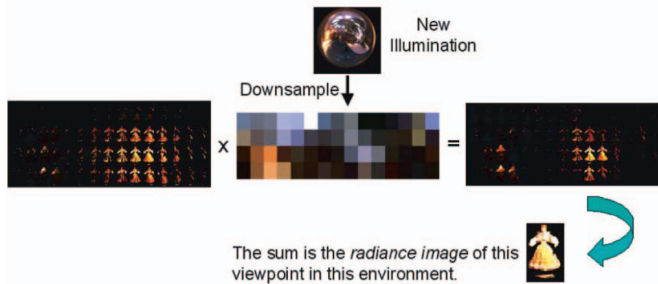
Fig. 2. Relighting using a reflectance image. The environment map of the new illumination is downsampled to the resolution of the reflectance image. Images are multiplied with the corresponding environment map color and added up to yield an image of the object under new illumination.

used to shade the warped impostor geometry during image generation (Fig. 3). Shading a point on the impostor surface requires applying this look-up function to the surface point, the viewing ray, and the incident light direction. Once warped back into the acquisition frame, the point can be shaded by blending the reflectance images according to the mapped light direction and viewing ray.

In the remainder of this paper, we develop this look-up scheme. We discuss some shading issues and present our implementation based on a point-based impostor geometry. Section 6 shows some examples of deformed and animated surface reflectance fields.

4 DEFORMABLE SURFACE REFLECTANCE FIELDS

Neglecting global light transport, a surface reflectance field can be understood as a discrete sampling of the object's BRDFs, knowing neither the exact location of the surface nor its normals. Ignoring wavelength and time, a BRDF is a scalar function $\mathrm{BRDF}(l, v)$, describing the fraction of light that is reflected in direction v as the surface is lit from direction l. Similar to the BRDF notation, we use $\mathrm{SRF}(p, l, v)$ to denote the reflectance of the SRF for an impostor point p,
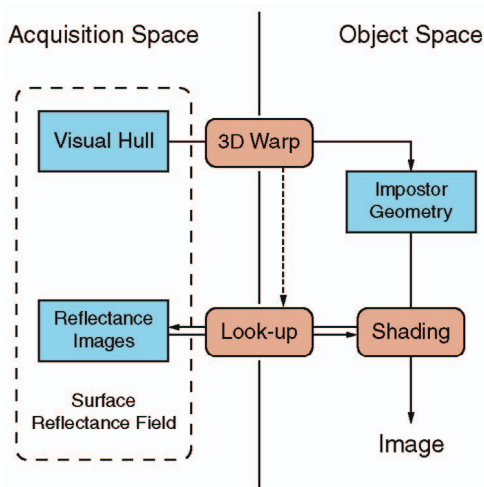

Fig. 3. Rendering a deformed surface reflectance field requires switching between reference frames. The acquired impostor geometry is warped into object space. During the shading process, queries to the reflectance images must be transformed back into the acquisition frame using a local parameterization of the impostor geometry.
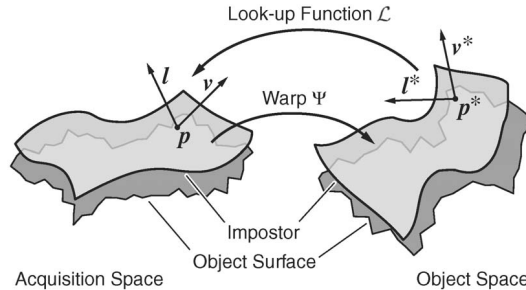

Fig. 4. During rendering, reflectance queries must be mapped into acquisition space. Given lighting and viewing direction l^* and v^* for an impostor point p^*, an appropriate surface reflectance field query $SRF(p, l, v)$ with p, l, and v in acquisition space has to be found. The look-up function $\mathcal{L}: (p^*, l^*, v^*) \mapsto (p, l, v)$ should be appearance preserving.

a light direction l, and a viewing direction v. This relationship is fundamental for our analysis of appearance preservation during surface reflectance field animation.

The initial step in animating an SRF is to deform the impostor geometry. In this section, we assume the impostor warp is defined by a differentiable warp function

$$\Psi: \mathbb{R}^3 \to \mathbb{R}^3. \qquad (1)$$

Shading an object includes queries to the surface reflectance field to determine the object's local reflectance. If the object is warped, the shading operation has to map a query (p^*, l^*, v^*) in object space to a query (p, l, v) in the original acquisition space (Fig. 4). By intuition, the required mapping approximately follows the inverse warp Ψ^{-1}.

However, applying Ψ^{-1} to the lighting and viewing directions is not necessarily appearance preserving, especially in the case of nonuniform deformations. Section 4.1 develops a mapping $\mathcal{L}$ to perform this operation, leading to a new impostor parameterization presented in Section 4.2. The final shading process is discussed in Section 4.3.

4.1 Approximate BRDF Preservation

In general, a mapping from object space to SRF acquisition space that preserves all aspects of the object's appearance cannot exist. This is due to the lack of exact object geometry and material properties in the SRF representation. Both would be needed to allow a prediction of complex nonlocal effects such as self-shadowing and interreflections.

Thus, we make the simplifying assumption, that the observed object can be described completely by its local BRDFs. We develop a look-up function that tries to preserve the main characteristics of the original object BRDFs. We do not aim at modeling changes in the BRDF due to the deformation of the material's microstructure. This would require precise knowledge of the structure, e.g., its microfacet distribution. Instead, we want to map the original BRDF to the deformed surface.

The desired look-up function is a mapping

$$\mathcal{L}: (p^*, l^*, v^*) \mapsto (p, l, v). \qquad (2)$$

We enforce appearance preservation by imposing three conditions:

1. Suppose the viewing ray in object space $p^* + sv^*$, $s \in \mathbb{R}$ intersects the warped object geometry at a point q^*
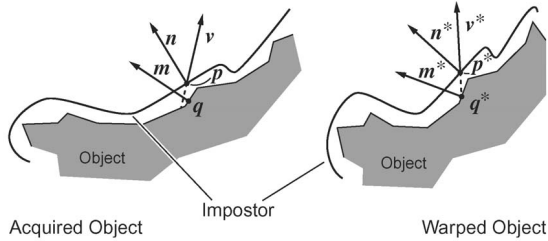
Fig. 5. A viewing ray $p^* + sv^*$, $s \in \mathbb{R}$, in object space intersects the warped object at q^*. The corresponding viewing ray in the surface reflectance field's acquisition frame should intersect the object at the original object point q. However, the impostor geometry of the SRF is typically not the same as the actual object geometry, so q^* and q remain unknown. Instead, we are using $p = \Psi^{-1}(p^*)$ as the origin of the viewing ray $p + tv$, $t \in \mathbb{R}$ in acquisition space.

Fig. 6. The bidirectional reflection distribution function (BRDF) describes the fraction of light from l that is reflected toward v. The spatial characteristics of a BRDF can be preserved by preserving the angles ϑ_l, ϑ_v, and γ when changing to acquisition space. Anisotropic materials require the preservation of the azimuthal orientations φ_l and φ_v relative to the surface frame. ϑ_l and ϑ_v are relative to the surface normal m. In an SRF, m is not exactly known. Instead, the impostor normal n is taken as an approximation.

(see Fig. 5). Then, the viewing ray in acquisition space $p + tv$, $t \in \mathbb{R}$ should intersect the original object at the corresponding point $q = \Psi^{-1}(q^*)$ to ensure the reflected light originates from the same surface point with the same BRDF. Unfortunately, this condition cannot be guaranteed since the true object geometry is typically not known, i.e., the exact location of the points q and q^* cannot be determined. However, it is a reasonable approximation to force the two viewing rays to intersect corresponding points of the impostor geometry since the impostor point p is very likely close to q. This can be achieved by choosing

$$p := \Psi^{-1}(p^*). \tag{3}$$

2. l and the surface normal m in q have to enclose the same angle as l^* and the normal m^* in q^*. This is a necessary condition to retain the reflectance characteristics of the object, as, for example, the shape of the reflectance lobes of the corresponding BRDF. The same condition applies for v and v^*. Preserving the shape of the reflectance lobes requires preserving the angle between l^* and v^* as well (see Fig. 6).
3. In order to preserve the effect of anisotropic object BRDFs, l and v should have the same azimuthal orientation relative to the object surface as l^* and v^* on the warped object. (See Fig. 6.)

As p immediately follows from (3), the mapping $(p^*, l^*, v^*) \mapsto (l, v)$ remains to be found. This can be rewritten as a locally affine mapping

$$\mathcal{L}_{p^*}: (l^*, v^*) \mapsto (l, v) \tag{4}$$

of lighting and viewing directions in the vicinity of p^*. Note that $\mathcal{L}_{p^*}$ is a function of p^*.

According to the second condition, $\mathcal{L}_{p^*}$ has to be angle preserving. By convention, l, v, l^*, and v^* are unit vectors, so $\mathcal{L}_{p^*}$ needs to be length preserving as well. This implies that $\mathcal{L}_{p^*}$ is an isometry, which means that the effect of $\mathcal{L}$ on l^* and v^* can be described by a rotation or a reflection, respectively, as a function of location p^*. In the remainder of this paper, we assume $\mathcal{L}_{p^*}$ to be a rotation. The special case of a reflection can be handled similarly and is left out for simplicity.
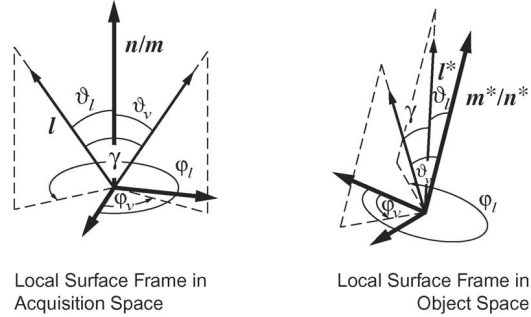
Using this observation and interpreting (3) as a translation of p^* by $\Psi^{-1}(p^*) - p^*$, the total effect of $\mathcal{L}$ on a local setting around p^* can be expressed as a *rigid transformation*, translating the point p^* onto p while rotating lighting and viewing directions.

Condition 2 further restricts $\mathcal{L}$ to rigid transformations that map the warped object normal at q^* onto the normal at the original point q. We do not know the exact object geometry. Instead, we refer to the impostor normal in p as an approximation of the true surface normal. Consequently, the searched $\mathcal{L}$ maps the tangential plane of the warped impostor point p^* onto the tangential plane in p.

According to condition 3, $\mathcal{L}$ should be chosen to preserve l^*'s and v^*'s orientation inside the tangential plane. The following section presents a local impostor parameterization that enables one to track the transformation of the local tangential frame in order to find a mapping $\mathcal{L}$ that fulfills the conditions.

4.2 Local Impostor Parameterization

We augment the impostor geometry with a local parameterization that allows us to determine the look-up function $\mathcal{L}$ at each impostor point p^*. The parameterization is independent of the geometric representation and can be applied to triangular meshes as well as to point sampled geometry. Depending on the representation, p^* can be a mesh vertex or a nonconnected surface point, respectively. This section develops the set of parameters that are stored at every point p^* (see Table 1).

Given an arbitrary warp $\Psi: \mathbb{R}^3 \to \mathbb{R}^3$, finding an explicit inverse function Ψ^{-1} is impossible in general. Constraining the SRF deformation to cases where an explicit inverse warp function exists would be too limiting. Consequently, another design goal for the parameterization was to determine $\mathcal{L}$ without using a closed form of Ψ^{-1}.

As shown above, $\mathcal{L}$ can be decomposed into a translational part and a rotation. The translation moves the warped impostor point back to its original position (see (3)). Storing the original position p_0 in each impostor point p^* allows the application of the translation without using Ψ^{-1}.

The rotational part $\mathcal{L}_{p^*}$ (see (4)) aligns the warped tangential frame of the impostor point with the corresponding

TABLE 1
The Local Impostor Parameterization Allows for the
Determination of the Look-Up Function $\mathcal{L}$ during Rendering

Parameter	Description
p_0	Original position of the impostor point p^* in acquisition frame.
r_0	Orientation R_0 of the tangential system of p^* in the acquisition frame, stored as a Rodrigues vector.
(u, v)	Local tangential frame of p^*.

The parameterization consists of a set of vectors (p_0, r_0, u, v) in $\mathbb{R}^3$, stored at each point of the discretized impostor geometry. p_0 and r_0 are fixed, while u and v have to be adapted whenever the geometry is deformed.

tangential system in the acquisition frame (see Fig. 7). To avoid the application of Ψ^{-1}, the tangential orientation R_0 in the acquisition frame must be explicitly stored in p^*. R_0 is defined by the rotation matrix (u_0, v_0, n_0), built by the tangential system in acquisition space. u_0 and v_0 must be orthogonal and can be arbitrarily chosen. To minimize its memory footprint, R_0 can be stored as a Rodrigues vector [1]

$$r_0 = k \tan\frac{\vartheta}{2}, \qquad (5)$$

given R_0 as a rotation of angle ϑ around an axis k. This definition provides a minimal (i.e., three-dimensional) parameterization of R_0. It contains a singularity for $180°$ rotations that can be avoided in our context as R_0 can be arbitrarily chosen.

Additionally, a tangential coordinate system (u, v) is attached to p^*. While p_0 and R_0 remain unchanged during the deformation, (u, v) is subject to the same warp as the geometry: After a deformation by Ψ, (u, v) is set to $\Psi((u_0, v_0))$. Applying the warp function Ψ to a local tangential system is a standard problem in differential geometry. The tangential system has to be mapped by taking directional derivatives of Ψ. In our implementation, we use central differences to determine u and v:

$$u = \frac{1}{2\varepsilon}(\Psi(p_0 + \varepsilon u_0) - \Psi(p_0 - \varepsilon u_0)),$$
$$v = \frac{1}{2\varepsilon}(\Psi(p_0 + \varepsilon v_0) - \Psi(p_0 - \varepsilon v_0)) \qquad (6)$$

for a small $\varepsilon > 0$. In particular, this only restricts Ψ to be differentiable in a vicinity of the impostor geometry.

Knowing R_0 and (u, v), the rotation $\mathcal{L}_{p^*}$ can be easily reconstructed during rendering. Let

$$\bar{n} = \frac{u \times v}{\|u \times v\|} \qquad (7)$$

and

$$\bar{b} = \frac{b}{\|b\|} \quad \text{with} \quad b = \frac{u}{\|u\|} + \frac{v}{\|v\|} \qquad (8)$$

be the normal and the normalized bisecting vector of the warped tangential system spanned by u and v, respectively (see Fig. 7). Then,
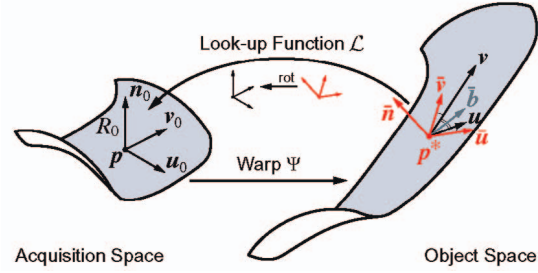

Fig. 7. For each point on the impostor geometry, the local impostor parameterization provides two coordinate systems: The orthogonal tangential system R_0 in p_0 in the acquisition frame and its warped counterpart on the rendered impostor geometry in object space, defined by p^* and (u, v). During rendering, p_0, R_0, and the orthogonalized tangential system $(\bar{u}, \bar{v})$ are used to determine the appearance-preserving back-projection of the SRF query.

$$\bar{u} = \frac{1}{\sqrt{2}}(\bar{b} + \bar{b} \times \bar{n}) \qquad \text{and}$$
$$\bar{v} = \frac{1}{\sqrt{2}}(\bar{b} + \bar{n} \times \bar{b}) \qquad (9)$$

build an orthogonalized tangential system that minimizes the squared angular differences between corresponding basis vectors of (u, v) and $(\bar{u}, \bar{v})$.

Using this orthogonalized system, the rotation $\mathcal{L}_{p^*}$ is given by

$$\mathcal{L}_{p^*}(x) = R_0 (\bar{u}, \bar{v}, \bar{n})^\top x. \qquad (10)$$

Note that, by choosing $(\bar{u}, \bar{v})$ as proposed, $\mathcal{L}_{p^*}$ is an approximation of tangential orientation preservation in the sense of condition 3 of the previous section.

This approximation may introduce a slight rotation of the principal axis of an anisotropic BRDF when the surface is sheared. The rotation may be counterintuitive depending on the direction of the shear. In general, the effect of a shear on a real-world material effectively changes the original BRDF, depending on its microstructure. As the microstructure is not known, we decided to use the original BRDF. Thus, a tangential rotation of certain characteristics of the anisotropic BRDF cannot be avoided.

It may not be surprising that $\mathcal{L}$ turned out to be a rigid transformation. For rigid object transformations and cases where impostor geometry and object surface coincide, rotating the reflectance data according to the inverse object transformation is the appropriate choice. Consequently, this technique is used in many applications, e.g., for image-based BRDF measurements [30], [19], [16], for bump mapping, for BTFs [4], [9], and for various other texturing techniques. All these techniques assume relatively accurate impostor geometry.

We assume arbitrary, in particular nonuniform, skewed deformations of an approximate impostor geometry that is different from the real object surface. The central result of our analysis is that, in these cases, again a rigid transformation of the impostor surface frame meets the requirements of appearance preservation best. Our framework allows the derivation of $\mathcal{L}$ for arbitrary deformations, using the local impostor parameterization. As the resulting transformation is rigid, material properties—including anisotropic BRDFs —stay the same, even for skewed object transformations.

4.3 Shading

Once $\mathcal{L}$ is determined, an impostor point can be shaded as described in Section 3. However, there are some issues that should be considered when lighting a deformed surface reflectance field.

When using an environment map to light an SRF, it needs to be filtered according to the spatial resolution of the reflectance images representing the SRF (see Fig. 1). Using an unfiltered environment map may lead to aliasing artifacts if the map contains details that are finer than the spacing of the light sources used to acquire the SRF.

During rendering of a deformed SRF, $\mathcal{L}_{p^*}$ is applied to all lighting directions incident to p^*. This corresponds to a rotation of the environment map before evaluating the SRF with that environment. Thus, the rotated environment map needs to be refiltered according to the reflectance field sampling. This operation would have to be performed for every surface point, which is impractical as filtering is an expensive operation for a nonuniform reflectance field sampling.

We present an alternative way to light a deformed surface reflectance field that comes without the need for refiltering. Although an SRF is acquired using directional lighting, we can simulate lighting by *point light sources* with only little artifacts.

Lighting an impostor point p^* by a point light source starts with the reflectance query (p^*, l^*, v^*) describing the viewing ray to p^*, and the direction l^* to the point light-source as seen from p^*. Applying the look-up scheme $(p, l, v) = \mathcal{L}(p^*, l^*, v^*)$ transforms the query into acquisition space. $\mathrm{SRF}(p, l, v)$ yields a reflectance coefficient that can be used to shade the impostor point p with a color I_p:

$$I_p = \mathrm{SRF}(p, l, v) A(\|p - p_l\|) I_l, \tag{11}$$

where I_l is the color of the point light source at p_l and $A(d)$ is the light attenuation factor depending on the distance to the light source. Using the impostor point p^* instead of a point on the real object surface introduces an error in the incident lighting direction. But, usually, this error is small compared to the resolution of the SRF.

In contrast to environment mapping, this technique introduces SRF queries for lighting directions that are not present in the reflectance images. We rely on the SRF implementation to properly interpolate novel lighting directions, as discussed in Section 5.2.

The proposed approximation of point light-sources allows for dynamic lighting effects when animating surface reflectance fields. However, as our look-up scheme ignores global illumination effects, these effects may appear wrong on deformed objects.

Point light sources can also be used to implement environment mapping without the need to refilter the map for every impostor point. This can be done by properly subsampling the environment once for a dense set of directions. Then, point light sources are defined at these directions, colored by the corresponding values of the environment. Provided the subsampling is dense enough, lighting the scene with these light sources leads to an appropriate reconstruction of the lighting environment. Note that light direction interpolation implicitly acts as a reconstruction filter. Thus, adaptively filtering the environment map is traded for interpolation. This alternative scheme is easier to implement and fits naturally into the framework of point light sources. For more flexible lighting effects, environment maps and point light sources can be combined.

4.4 Limitations

We presented a simple technique to preserve the appearance of a surface reflectance field under arbitrary deformations However, the look-up scheme contains some approximations affecting the proper reproduction of lighting effects. There are three classes of errors that may occur.

First, our BRDF-based derivation of the look-up function ignores nonlocal effects like interobject reflection, self-shadowing, refraction, or subsurface scattering. This may lead to false shadowing and erroneous refractions. However, modeling these effects requires incorporating the exact object geometry. As an SRF contains only approximate geometry information, these global lighting effects cannot be completely preserved during the deformation.

The BRDF preserving approach itself contains some approximations. As $\mathcal{L}_{p^*}$ is derived from the local warp around the observed impostor point instead of a point on the real object surface, the warped SRF may show some BRDFs that seem to be rotated relative to the surface frame. The effect grows as the local warp's rotational part around p^* is different from the rotation of the observed, real surface point q^*. Note that the error introduced by choosing the impostor normal to determine $\mathcal{L}_{p^*}$ is comparatively low. In fact, the normal is only used to derive the rotational approximation of the local warp. Evaluating the SRF does not incorporate a normal anymore. In particular, this normal has no impact on the reflectance properties of the reproduced BRDFs.

A third error class affects object/impostor parallaxis in regions where the impostor geometry is distant to the object surface: For nonuniform stretches, the object texture may appear to be shifted. This happens if the corresponding viewing rays in object and acquisition space do not intersect the object at corresponding points q and q^*. Fig. 5 shows such an example.

Despite these limitations, the proposed scheme shows significant advantages. Its ability to preserve reflectance properties without explicit knowledge of the real surface normals is crucial for the deformation of surface reflectance fields. Moreover, it makes it suited for many other image-based applications where the exact object normals are not known.

An important property of this look-up scheme is that, in the limit, the BRDF preservation is exact as the SRF geometry converges to the real object geometry. The proposed method directly benefits from improvements of the geometric representation, for example, for geometry acquired with laser range scanning.

5 IMPLEMENTATION

We implemented an animation system for surface reflectance fields similar to the system presented by Matusik et al. [20]. We improved the impostor representation, its rendering, and the reflectance field interpolation. Finally, we present a new shading scheme that speeds up the rendering of warped SRFs.
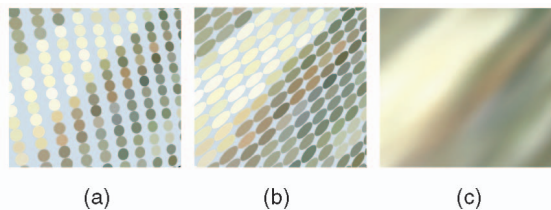
(a)   (b)   (c)

Fig. 8. Close-ups of a surfel geometry. The surfels are placed on a square grid that was anisotropically stretched. (a) and (b) depict surface splats with reduced support to show the surfel distribution. (a) Circular splats cannot follow the deformation. Thus, holes in the surface may appear. (b) Adaptively deformed elliptical splats preserve a closed surface and provide anisotropic texture filtering. (c) Blending the elliptical splats at their original size produces a smooth texturing.

Fig. 9. A surfel is defined by its position p and its tangential system (u, v). u and v span a skewed coordinate system that defines the surfel's elliptical region of influence.

5.1 Surfel Representation

Previous work on surface reflectance fields [20] used a point-based representation of the impostor geometry. In this approach, the impostor was densely sampled by *surfels*. Surfels, as introduced by [26], are points in $\mathbb{R}^3$, augmented by additional attributes, such as normal, radius, and some color properties. In their representation, the normal indicates the surface orientation, whereas the radius can be understood as an approximate, circular region of influence.

Subsequently, [35], [36] presented a framework to render surfels based on Heckbert's elliptical weighted average (EWA) texture filtering, [12]. EWA surface splatting is a forward-mapping algorithm that allows for high quality, aliasing-free rendering of point sampled geometry.

Although featuring high image quality, the surfel renderer as used by [20] is not suited for our purposes. It organizes the surfels in a static layered depth cube (LDC) tree, a fixed, axis aligned spatial data structure, that cannot be warped. Instead, we use a variant of the Pointshop3D renderer [34], which is an advanced version of EWA surface splatting. The Pointshop3D renderer is able to render unorganized point-sets and, thus, can cope with dynamically changing point sets.

Deformations of the impostor geometry may lead to visible holes in the point-based reconstruction. For reasonable deformations, this can be avoided by deforming the splat geometry appropriately. As proposed by [36], we are using elliptical surface splats. Fig. 8 shows the effect on a deformed patch of a surfel geometry. In contrast to circular splats that use a position/normal parametrization, our splats are defined by a position p and two tangential vectors u and v, spanning a tangential coordinate system. u and v are explicitly allowed to be nonorthonormal. The surfel is associated with an elliptical region of influence defined by the tangential system (see Fig. 9).

When warping the impostor geometry, the warp has to be applied on each surfel's tangential system by using an affine approximation of the local distortion. In our implementation, this approximation is determined similarly to (u, v) in Section 4.2. This ensures surface coverage as long as the warp's first derivative does not change excessively (see Fig. 8).

The surfel's tangential vectors u and v provide us directly with the corresponding parameters of our local impostor parameterization. By adding the surfel's original position p_0 and orientation r_0 to the set of surfel attributes,
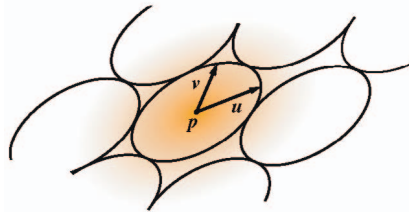
all impostor parameters required for the warped SRF rendering are encoded in the surfels.

5.2 Reflectance Image Interpolation

In general, each surface reflectance field look-up $\mathrm{SRF}(p, l, v)$, must be interpolated from the fixed set of reflectance images. Depending on the viewing ray $p + sv, \ s \in \mathbb{R}$, the query must be handled by interpolating between the observing camera views of the SRF. View interpolation is a common problem in image-based rendering. Like Matusik et al. [20], we use unstructured lumigraph interpolation that was introduced with unstructured lumigraph rendering (ULR) [2].

ULR interpolation has proven to be a very flexible interpolation scheme that can be controlled by an arbitrary penalty function, rating each camera's appropriateness. In our implementation, we chose the same penalty as in [20].

In previous work, the queried lighting direction l was always chosen from the set of acquired illuminations, sampling a stationary environment map. However, since our point light-source implementation produces arbitrary lighting directions, we have to interpolate reflectance queries between the acquired lighting directions as well.

This problem has been addressed by Koudelka et al. [15] who intersected the queried light ray with a triangulated hull of the acquisition lightsource positions. The intersected triangle yielded the three closest light sources that were consequently used to interpolate between three different lighting conditions. However, this does not work for arbitrary positions of the acquisition light sources. We also found that, for slim triangles in the hull, light source interpolation was not smooth enough.

Instead, we are also using ULR to interpolate between light directions. Treating the light sources of the acquisition stage as cameras at infinite distance enables us to use ULR for light source interpolation. In our implementation, this is done by using the same penalty function as for camera interpolation.

Using ULR for camera selection affords consideration of occlusions in the scene. If the observed object point is not visible from a given camera, this camera should not be considered for interpolation. When interpolating between light sources, there are no visibility constraints since shadowing effects are already captured in the acquired reflectance images. Moreover, taking occlusion into account would change the appearance of translucent or refractive materials.

5.3 Reordered Evaluation

Compared to the shading of static objects, substantially more reflectance images are involved during the shading of
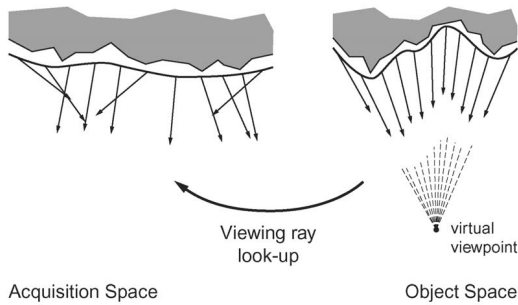
Fig. 10. Viewing ray divergence due to back-projection.

a deformed model. This is because the rendered viewing rays tend to diverge in acquisition space when the impostor is warped (see Fig. 10).

The number of reflectance images involved in the shading process increases with deformations. Since SRF data usually exceeds conventional main memory size, most SRF look-ups must be read from disk. The time needed for disk access exceeds all other computation times by orders of magnitude. This makes SRF look-up performance a critical issue, especially when producing animated SRFs.

We address this problem with two techniques: A smaller cache holds the most recently used blocks of data. Unfortunately, naively shading the geometry shows adverse cache-coherency. Thus, as a second technique, all shading operations are decomposed into minimal *shading operations*, each containing a single SRF look-up. Table 2 shows their detailed structure. The execution of the shading operation is deferred until a larger number of operations has been collected. Sorting them before execution in a cache-optimal way decreases shading times by an order of magnitude. Moreover, due to the the cache coherent shading execution, the cache can be kept very small. In our system, it was sufficient to dimension the cache large enough to hold three reflectance images at the time. To facilitate the deferred shading, we accumulate all surfel colors first before rendering them to frame buffer.

6 RESULTS

For our experiments, we use the two models shown in Fig. 11 that were scanned by Matusik et al. [20], [21]. The doll SRF is well-suited to the analysis of appearance preservation as it contains many different materials. The

TABLE 2
A *shading operation* Represents
a Basic Reflectance Field Look-Up

Entry	Description
$\mathrm{REF}(S)$	Reference to the target surfel.
(i_c, i_l)	Camera index and light source index for the reflectance image look-up.
(x_c, y_c)	Look-up position on the camera plane.
$(\omega_r, \omega_g, \omega_b)$	Color weights.

The shading operation contains the depicted entries that are used to accumulate reflectance values in the surfel colors. A shading operation is executed by looking up the reflectance image (i_c, i_l) at position (x_c, y_c). The resulting RGB reflectance value is weighted by $(\omega_r, \omega_g, \omega_b)$ and added to the color of surfel S.

Model	Viewpoints	Lights	Surfels	PCA Data
Doll	432 (360°)	60	532 935	4.4 GB
Mug	216 (180°)	44	757 760	1.8 GB

Fig. 11. The presented models. Left: The surface reflectance field of a doll. Right: A half-filled beer mug. The table shows the complexity of each data set.

mug SRF was chosen to show the limitations of our approach as it contains refractive effects that cannot be handled correctly by our technique.

The reflectance images of the models were captured using the high dynamic range (HDR) technique of Debevec and Malik [6]. For each viewpoint and lighting condition, four $1,360 \times 1,032$ pictures with different exposure time were taken. Using up to 432 different viewpoints and 60 light directions, this produced a total of $432 \times 60 \times 4 = 103,680$ images. Uncompressed this would correspond to 407 GB of data. All reflectance images were compressed using a PCA compression that achieved a reduction of at least a factor of 10. Fig. 11 lists the compressed size of both models.

With these optimizations, the vast part of the rendering time is spent for the ULR computation. Our current implementation assumes arbitrarily placed cameras and light sources during the acquisition. Using knowledge about the scanner geometry or a spatial data structure to locate the *k*-nearest light sources or cameras would speed up the rendering times significantly. However, this optimization has not been performed, resulting in rendering times of about 10 minutes per frame for the doll data set with five point light sources on a 1GHz Pentium III.

View extrapolation is an important issue when deforming surface reflectance fields as parts of the objects may become visible from directions where no data was acquired before. The extrapolation is implicitly covered by the ULR interpolation. Fig. 12 shows an example.

Fig. 13 shows the quality of appearance preservation at the example of the doll data set. Note the visual preservation of the different materials of the wooden base, the diffuse fabric, and the specular braid of the skirt. All scenes were lit by three colored point light sources (*red*, *green*, and *blue*) from static positions.

Fig. 14 shows various deformations of the beer mug model. Although our approach is not able to preserve refractive effects, the results appear realistic for reasonable deformations.

Our surfel renderer distorts surface splats in object space according to the local warp. This allows for rendering large deformations without visible holes (Fig. 15).
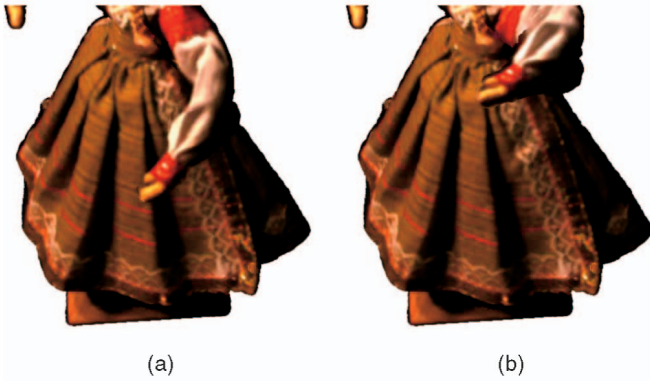
Fig. 12. View extrapolation. (a) Original doll data set. (b) Occluded parts of the skirt become visible as one arm is lifted. The model is lit by two point light sources. (a) is chosen to be an original view to force the views used for view extrapolation to be at maximum distance from the current view.

Section 5.3 discussed the viewing ray divergence coming with the deformation of a surface reflectance field. Fig. 16 shows an example of an SRF warp together with its corresponding camera and light source blending fields. Both blending fields were visualized by assigning random colors to cameras and light sources, respectively. Subsequently, the colors were blended together according to the ULR interpolation coefficients. The blending fields clearly show the viewing ray divergence in distorted regions.

Fig. 17 shows a number of frames of a surface reflectance field animation, including large deformations and varying lighting conditions.

7   CONCLUSIONS

We developed a novel scheme to deform and relight surface reflectance fields. Our technique approximately preserves all material properties of the captured object during deformation. The method uses an enclosing impostor geometry of the rendered object. An appearance preserving data look-up scheme employs a local parameterization of the impostor geometry without relying on proper surface normals. Our method is especially suited for relighting of

Fig. 14. Deformation of refractive objects. Different warps were applied to the mug data set. All glasses are rendered from the same viewpoint and lit by five point light sources (red, green, blue, $2 \times$ white). The effect of the deformations on specular highlights can easily be observed.

real-world models with complex surface properties for which normals are hard to determine.

We presented an extended surfel representation to render deformable surface reflectance fields. Using elliptical splats that follow the local deformation in object space allowed for seamless rendering of deformations while preserving optimal texture quality. At the same time, the surfel representation naturally complements the local impostor parameterization. In principle, the presented look-up scheme can be used for any other image-based approach that uses approximating impostor geometry.
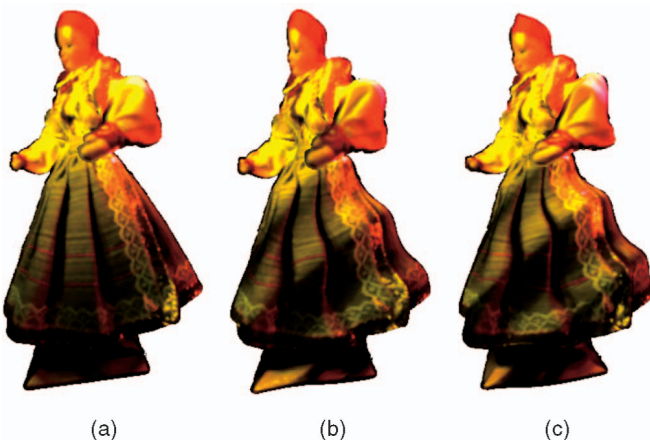

Fig. 13. Preservation of material properties (a) The original doll surface reflectance field. (b), (c) A sinusoidal warp is applied to the data set.
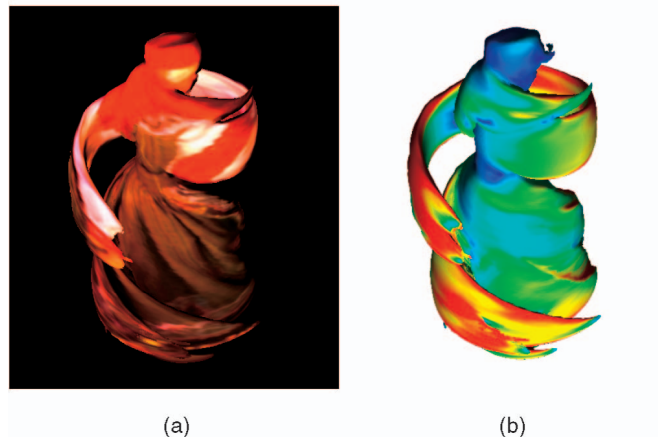

Fig. 15. Even under large deformations, the surfel renderer displays a closed surface. (a) The depicted warp anisotropically stretches parts of the doll data. (b) The stretch anisotropy is color encoded showing the ratio of the principal axes of the elliptical surface splats. Splats in the blue regions remain circular, green areas show a stretch ratio of 1:2, while red denotes a local stretch of more than 1:4.
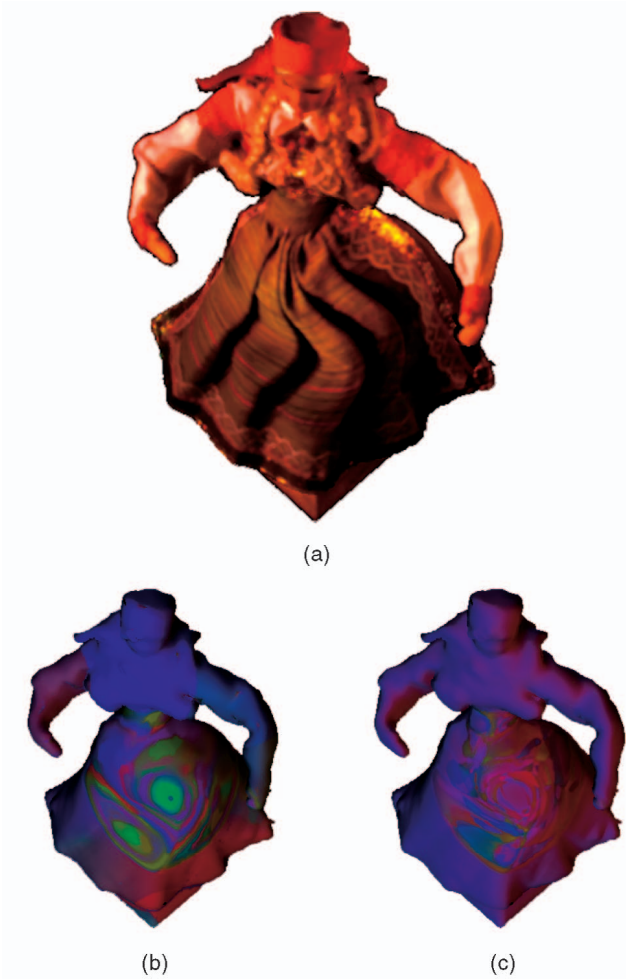
Fig. 16. Viewing ray divergence. (a) The doll's skirt was distorted. (b) The corresponding camera blending field displays the viewing ray divergence in distorted regions. (c) The light source blending field.

8 FUTURE WORK

Simulating global lighting effects during deformation requires knowledge of the object geometry. While surface reflectance fields only provide an approximate geometry in general, this geometry is usually close to the real object surface. Exploiting this fact may lead to approximate methods to analyze and reproduce global illumination on deformable surface reflectance fields.

Another way to preserve global effects would be to collect additional information during the acquisition process of the surface reflectance field. Matusik et al [20] proposed the *opacity hull* to explicitly capture translucency and refraction through an SRF. This information could be used to reconstruct the internal light transport. In particular, opacity hulls may be used to detect and manipulate self-shadowing inside a surface reflectance field.

Shifted texture parallaxis can be alleviated by enhancing the impostor geometry. Stereo algorithms could be used on textured objects to improve the geometric approximation.

REFERENCES

[1] S.L. Altman, "Hamilton, Grassmann, Rodrigues, and the Quaternion Scandal—What Went Wrong with One of the Major Mathematical Discoveries of the Nineteenth Century?" *A Math. Assoc. Am. J.*, Dec. 1989.

[2] C. Buehler, M. Bosse, L. McMillan, S. Gortler, and M. Cohen, "Unstructured Lumigraph Rendering," *Computer Graphics, SIGGRAPH 2001 Proc.*, pp. 425-432, 2001.

[3] W.-C. Chen, J.-Y. Bouguet, M.H. Chu, and R. Grzeszczuk, "Light Field Mapping: Efficient Representation and Hardware Rendering of Surface Light Fields," *ACM Trans. Graphics*, vol. 21, no. 3, pp. 447-456, July 2002.

[4] K.J. Dana, B. van Ginneken, S.K. Nayar, and J.J. Koenderink, "Reflectance and Texture of Real World Surfaces," *ACM Trans. Graphics*, vol. 1, no. 18, pp. 1-34, 1999.
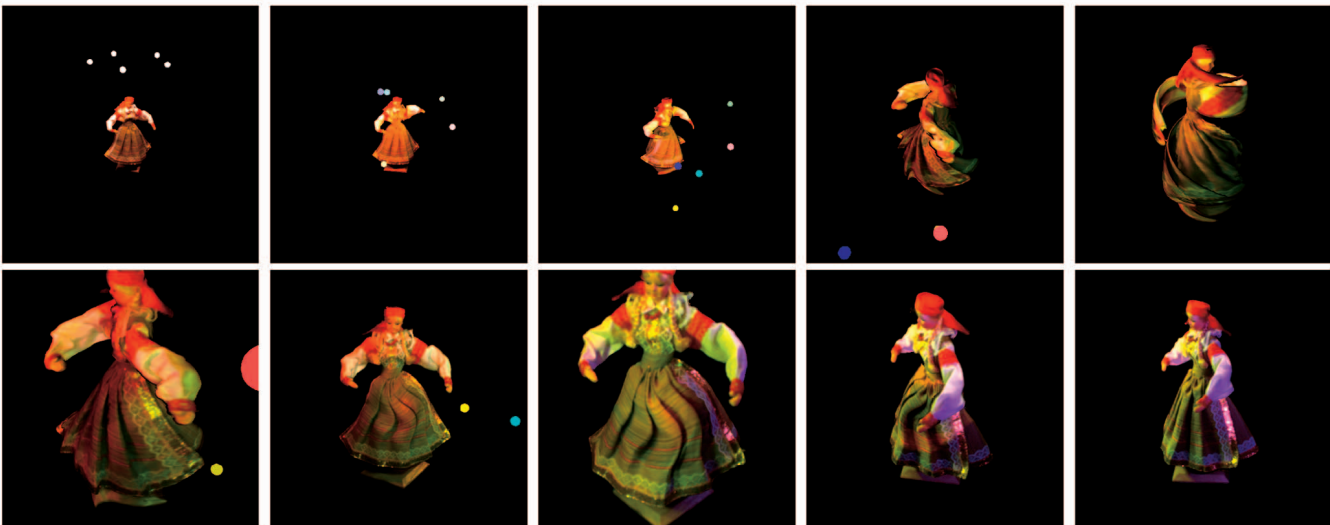

Fig. 17. Selected frames of a surface reflectance field animation. The model is continuously deformed, while viewpoint and lighting conditions change. The model is lit by colored light sources, donoted by spheres of the light color.

[5] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar, "Acquiring the Reflectance Field of a Human Face," *Computer Graphics, SIGGRAPH 2000 Proc.,* pp. 145-156, July 2000.

[6] P. Debevec and J. Malik, "Recovering High Dynamic Range Radiance Maps from Photographs," *Computer Graphics SIGGRAPH '97 Proc.,* pp. 369-378, 1997.

[7] P. Debevec, C. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry- and Image-Based Approach," *Computer Graphics SIGGRAPH '96 Proc.,* pp. 11-20, Aug. 1996.

[8] P. Debevec, Y. Yu, and G. Borshukov, "Efficient View-Dependent Image-Based Rendering with Projective Texture-Mapping," *Proc. Ninth Eurographics Workshop Rendering,* pp. 105-116, June 1998.

[9] R. Furukawa, H. Kawasaki, K. Ikeuchi, and M. Sakauchi, "Appearance Based Object Modeling Using Texture Database: Acquisition, Compression and Rendering," *Proc. 13th Eurographics Workshop Rendering,* June 2002.

[10] S. Gortler, R. Grzeszczuk, R. Szeliski, and M. Cohen, "The Lumigraph," *Computer Graphics SIGGRAPH '96 Proc.,* pp. 43-54, Aug. 1996.

[11] T. Hawkins, J. Cohen, and P. Debevec, "A Photometric Approach to Digitizing Cultural Artifacts," *Proc. Second Int'l Symp. Virtual Reality, Archaeology, and Cultural Heritage,* Nov. 2001.

[12] P. Heckbert, "Fundamentals of Texture Mapping and Image Warping," master's thesis, Dept. of Electrical Eng. and Computer Science, Univ. of California, Berkeley, June 1989.

[13] W. Heidrich and H.-P. Seidel, "Realistic, Hardware-Accelerated Shading and Lighting," *Computer Graphics SIGGRAPH '99 Proc.,* pp. 171-178, Aug. 1999.

[14] J. Kautz and M. McCool, "Interactive Rendering with Arbitrary Brdfs Using Separable Approximations," *Proc. Rendering Techniques '99 (Proc. 10th Eurographics Workshop Rendering),* pp. 281-292, June 1999.

[15] M. Koudelka, P. Belhumeur, S. Magda, and D. Kriegman, "Image-Based Modeling and Rendering of Surfaces with Arbitrary BRDFs," *Proc. Computer Vision and Pattern Recognition,* Dec. 2001.

[16] H. Lensch, J. Kautz, M. Goesele, W. Heidrich, and H.-P. Seidel, "Image-Based Reconstruction of Spatially Varying Materials," *Proc. 12th Eurographics Workshop Rendering,* pp. 104-115, June 2001.

[17] M. Levoy and P. Hanrahan, "Light Field Rendering," *Computer Graphics SIGGRAPH '96 Proc.,* pp. 31-42, Aug. 1996.

[18] T. Malzbender, D. Gelb, and H. Wolters, "Polynomial Texture Maps," *Computer Graphics SIGGRAPH 2001 Proc.,* pp. 519-528, 2001.

[19] S. Marschner, S. Westin, E. Lafortune, and K. Torrance, "Image-Based Measurement of the Bidirectional Reflectance Distribution Function," *Applied Optics,* vol. 39, no. 16, 2000.

[20] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, R. Ziegler, and L. McMillan, "Image-Based 3D Photography Using Opacity Hulls," *ACM Trans. Graphics,* vol. 21, no. 3, pp. 427-437, July 2002.

[21] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan, "Acquisition and Rendering of Transparent and Refractive Objects," *Proc. 13th Eurographics Workshop Rendering,* June 2002.

[22] M. McCool, J. Ang, and A. Ahmad, "Homomorphic Factorization of BRDFs for High-Performance Rendering," *Computer Graphics,* vol. 35, pp. 171-178, 2001.

[23] G. Miller, S. Rubin, and D. Ponceleon, "Lazy Decompression of Surface Light Fields for Precomputed Global Illumination," *Proc. Ninth Eurographics Workshop Rendering,* pp. 281-292, June 1998.

[24] K. Nishino, Y. Sato, and K. Ikeuchi, "Appearance Compression and Synthesis Based on 3D Model for Mixed Reality," *Proc. IEEE Int'l Conf. Computer Vision (ICCV '99),* pp. 38-45, Sept. 1999.

[25] K. Nishino, Y. Sato, and K. Ikeuchi, "Eigen-Texture Method: Appearance Compression Based on 3D Model," *Proc. Computer Vision and Pattern Recognition,* pp. 618-624, June 1999.

[26] H. Pfister, M. Zwicker, J. van Baar, and M. Gross, "Surfels: Surface Elements as Rendering Primitives," *Computer Graphics, SIGGRAPH 2000 Proc.,* pp. 335-342, July 2000.

[27] K. Pulli, M. Cohen, T. Duchamp, H. Hoppe, L. Shapiro, and W. Stuetzle, "View-Based Rendering: Visualizing Real Objects from Scanned Range and Color Data," *Proc. Eurographics Rendering Workshop 1997,* pp. 23-34, June 1997.

[28] Y. Sato, M.D. Wheeler, and K. Ikeuchi, "Object Shape and Reflectance Modeling from Observation," *Computer Graphics, SIGGRAPH '97 Proc.,* pp. 379-387, 1997.

[29] D. Vlasic, H. Pfister, S. Molinov, R. Grzeszczuk, and W. Matusik, "Opacity Light Fields: Interactive Rendering of Surface Light Fields with View-Dependent Opacity," *Proc. Interactive 3D Graphics Symp. 2003,* 2003.

[30] G. Ward, "Measuring and Modeling Anisotropic Reflection," *Computer Graphics,* vol. 26, pp. 265-273, 1992.

[31] D. Wood, D. Azuma, K. Aldinger, B. Curless, T. Duchamp, D. Salesin, and W. Stuetzle, "Surface Light Fields for 3D Photography," *Computer Graphics, SIGGRAPH 2000 Proc.,* pp. 287-296, July 2000.

[32] Y. Yu, P. Debevec, J. Malik, and T. Hawkins, "Inverse Global Illumination: Recovering Reflectance Models of Real Scenes from Photographs," *Computer Graphics, SIGGRAPH '99 Proc.,* pp. 215-224, Aug. 1999.

[33] Z. Zhang, L. Wang, B. Guo, and H.-Y. Shum, "Feature-Based Light Field Morphing," *Computer Graphics, SIGGRAPH 2002 Proc.,* pp. 457-464, July 2002.

[34] M. Zwicker, M. Pauly, O. Knoll, and M. Gross, "Pointshop 3D: An Interactive System for Point-Based Surface Editing," *Computer Graphics, SIGGRAPH 2002 Proc.,* pp. 322-329, July 2002.

[35] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "Surface Splatting," *Computer Graphics, SIGGRAPH 2001 Proc.,* pp. 371-378, July 2001.

[36] M. Zwicker, H. Pfister, J. Van Baar, and M. Gross, "EWA Splatting," *IEEE Trans. Visualization and Computer Graphics,* vol. 8, no. 3, pp. 223-238, July-Sept. 2002.

Tim Weyrich received the MS degree in computer science from the University of Karlsruhe (TU), Germany, in 2001. He is currently a PhD student in the Computer Graphics Laboratory of the Institute of Technology (ETH) in Zürich. His research interests are point-based graphics, 3D reconstruction, and appearance modeling. He is member of ACM SIGGRAPH and GI.

Hanspeter Pfister received the MS degree in electrical engineering from the Swiss Federal Institute of Technology (ETH) Zurich, Switzerland, in 1991. He received the PhD degree in computer science in 1996 from the State University of New York at Stony Brook. He is associate director and a senior research scientist at MERL—Mitsubishi Electric Research Laboratories—in Cambridge, Massachusetts. He is the chief architect of VolumePro, Mitsubishi Electric's real-time volume rendering hardware for PCs. His research interests include computer graphics, scientific visualization, and computer architecture. He has been teaching introductory and advanced graphics courses at the Harvard Extension School since 1999. He is an associate editor of the *IEEE Transactions on Visualization and Computer Graphics* (*TVCG*) and chair of the IEEE Technical Committee on Visualization and Computer Graphics (TCVG). He is a senior member of the IEEE and a member of the ACM, ACM SIGGRAPH, the IEEE Computer Society, and the Eurographics Association.

Markus Gross received the Master of Science degree in electrical and computer engineering and the PhD degree in computer graphics and image analysis, both from the University of Saarbrucken, Germany. He is a professor of computer science and director of the Computer Graphics Laboratory of the Swiss Federal Institute of Technology (ETH) in Zürich. From 1990 to 1994, he worked for the Computer Graphics Center in Darmstadt, where he established and directed the Visual Computing Group. His research interests include point-based graphics, physics-based modeling, multiresolution analysis, and virtual reality. He has been widely publishing and lecturing on computer graphics and scientific visualization and he authored the book *Visual Computing* (Springer, 1994). He is the associate editor of the *IEEE Computer Graphics and Applications* and has served as a member of international program committees of many graphics conferences. He will be chair of the papers committee of ACM SIGGRAPH 2005. He is a senior member of the IEEE.