Simplification of Meshes with Digitized Radiance

Kenneth Vanhoey · Basile Sauvage · Pierre Kraemer · Frédéric Larue ·
Jean-Michel Dischler

Abstract View-dependent surface color of virtual objects can be represented by outgoing radiance of the surface. In this paper we tackle the processing of outgoing radiance stored as a vertex attribute of triangle meshes. Data resulting from an acquisition process can be very large and computationally intensive to render. We show that when reducing the global memory footprint of such acquired objects, smartly reducing the spatial resolution is an effective strategy for overall appearance preservation. Whereas state-of-the-art simplification processes only consider scalar or vectorial attributes, we conversely consider radiance functions defined on the surface for which we derive a metric. For this purpose, several tools are introduced like coherent radiance function interpolation, gradient computation, and distance measurements. Both synthetic and acquired examples illustrate the benefit and the relevance of this radiance-aware simplification process.

Keywords Digitized artifacts · Surface light field · Radiance · Mesh simplification · Rendering

1 Introduction

In the scope of digitization of cultural heritage, the demand for high-fidelity visualization is increasing. Compared to usual colored surfaces (using, *e.g.*, textures, vertex colors), surface light fields (SLF) improve appearance modeling: the color depends not only on the position on the surface (spatial dimension) but also on the viewing direction. This allows capture of, *e.g.*, specular highlights induced by glossy objects. Applications include virtual museums, archiving, and off-site study.

Acquiring SLF is regularly made easier and more reliable by the development of acquisition devices, treatments, and reconstruction algorithms [14,24]. The memory load however still grows in proportion to the spatial resolution multiplied by the directional one. Memory limits for storage, transmission and rendering may be pushed in various ways, using data compression, streaming, data reduction, and level-of-detail approaches.

In this project, our acquisition process [24] generates dense surface meshes typically composed of hundreds of thousands to millions of vertices per object (spatial dimension). Compact representations of hemispherical radiance functions (RF) stored on these vertices still require dozens of coefficients per vertex (directional dimension). Our main contribution is a new method for the reduction of this dense data that exploits a radiance-aware metric for geometric mesh simplification, as illustrated in Fig. 1. To our knowledge, no other metric exists for such functional data, thus we compare our results to state-of-the-art color-aware simplification. We show that reducing the spatial resolution (our strategy) may better preserve the visual quality compared to reducing the directional resolution.

Alternatively, one can store radiance in texture maps instead of on vertices [2,25]. However texture mapping, filtering, and mip-mapping raise several issues. First, a parameterization has to be defined, which, for complex objects, implies to cut the texture into pieces (charts). To avoid visible seams at chart boundaries and permit mip-mapping, texel redundancy and complex data structures are often used [21]. Second, it is difficult to adapt the resolution of the texture according to spatial variations unless they have been taken into account when building the parameterization. Conversely, mesh

K. Vanhoey · B. Sauvage · P. Kraemer · F. Larue · J.-M. Dischler
ICube, Université de Strasbourg, CNRS, France

K. Vanhoey
Inria, France

| Full resolution (921k vertices) | 6.2% (58k) | 3.1% (29k) | 1.6% (14k) | + wireframe |
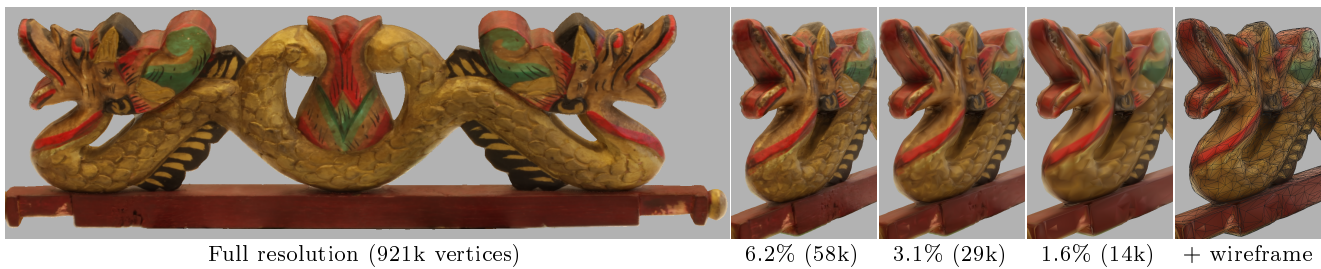
Fig. 1 Double dragon model with per-vertex radiance attribute. Mesh simplification adapts to geometric, color, and specular features. RF are encoded with polynomials of bi-degree 4 and respectively weigh $401MB$, $13MB$, $6MB$ and $3MB$.

simplification is designed to adapt the spatial resolution, assuming that radiance is stored on each vertex.

Some previous techniques [2,25] address the problem of compressing directional information independently of spatial information. Our approach is complementary since it addresses spatial simplification of the data: further directional compression remains compatible.

The key concept of our approach is to combine a well-known mesh-driven simplification algorithm (*i.e.*, iterative edge-collapse [11]) with a new metric defined on radiance, that measures what will be actually rendered. Therefore, RF are attached to the vertices of an over-dense surface mesh after acquisition and pre-processing. Our contributions start by first determining a set of tools to do calculations with RF on the surface, including specular highlight-aware interpolation, gradient computation and distance measurement (section 3). Second, we propose a method that simplifies dense data while ensuring visual similarity w.r.t. to the original. Therefore we define a new metric on RF that allows to evaluate the cost of an edge collapse operation (section 4). Level-of-detail rendering is then made possible by the use of progressive meshes: the data is represented at multiple scales such that it can be adapted at runtime to the actual rendering constraints. Third, because simplified meshes can exhibit large triangles that may cover many pixels in screen-space, we propose an improved rendering method in order to preserve the light field highlights even in those cases (section 5). Compared to directional reduction, the results show that radiance-aware spatial simplification well preserves acquired objects appearance. Compared to color-based metric, our method is also more accurate in the preservation of the directional features of the objects (section 6).

2 Related work

This work is related with two areas: i) view-dependent colors, namely light fields, and ii) mesh-driven level-of-detail methods, in particular mesh simplification.

2.1 Surface light fields

Light fields [10,17] define the color of a scene as a function of a 4D space covering position and the viewing direction. *Surface* light fields map these data on the surface as a way to discard background data and avoid projection and parallax errors, thus being more precise. It can be expressed by i) a combination of eigen-vectors by using factorization methods [5,20], or ii) independent localized 2D hemispherical RF [2,25]. For keeping local control in our algorithm, we use the latter.

Numerous function bases can be used to represent 2D RF in a data-driven acquisition context. Non-linear ones (*e.g.*, [16]) are precise but may be difficult to obtain or process [26]. They are rather used for representing BRDF, *i.e.*, 4D hemispherical functions. For 2D, linear combinations of basis functions are widely-used for their simplicity and flexibility. Commonplace are spherical harmonics or wavelets [18,23], polynomials [19], or lumispheres [25]. Choosing the appropriate one is left to the user: our algorithms are independent of it. This choice however influences both implementation complexity and computation time.

2.2 Mesh simplification

Surface mesh simplification has been given a lot of attention. We are interested in methods considering geometry and aspect-describing attributes (*e.g.*, color, texture) [3,4,7–9,11,12]. Methods exploiting the unitary edge collapse operation (Fig. 3 and Fig. 5), as opposed to vertex removal or triangle collapse, are standard both for ease of implementation and wide range of applications. Edges are iteratively collapsed in increasing order of damage they cause. Hereby, we define a new measure of this damage. Some metrics allow to define an optimal embedding (*i.e.*, that minimizes the metric) for the vertex resulting from a collapse. Half-edge collapse instead defines the resulting vertex as one of its two predecessors. Our new metric can be used with both variants.

Metric on geometry. The *quadric error metric* (QEM) is the most widely-used geometric error metric [7]. It efficiently estimates the sum of squared distances w.r.t. the planes defined by the triangles of the initial (dense) mesh surrounding the vertices preceding the collapse. A "memoryless" variant [12] computes the QEM w.r.t. the mesh immediately preceding the current collapse instead of the initial mesh. Our new metric is defined as a combination of a memoryless QEM with a (also memoryless) radiance measure.

Metric on scalar or vectorial attributes. Classical attributes (*e.g.*, normals, colors) are real-valued vectors of some dimension m. A first strategy directly extends the QEM to $\mathbb{R}^{3+m}$ by mixing up geometry and other attributes [8]. As considering such a mix is quite arbitrary, the more recent alternatives use the QEM as a geometry metric and add their specific error to it. In [12], a separate QEM is computed in attribute space $\mathbb{R}^m$, and in [15], a metric is specifically designed for vertex colors. Both combine this with the QEM for geometry. None of these techniques is directly extensible to functional attributes.

Metric on radiance attributes. To our knowledge, functional attributes have not been considered before. The RF we are dealing with encode a color depending on the viewing direction. A constant RF (diffuse color) is thus equivalent to color attributes. We compare our radiance metric on such data in Fig. 9.

Progressive meshes [11] reverse edge collapses by vertex splits. They represent triangle meshes at different resolutions. In the present paper we show progressive meshes with radiance attribute.

3 Calculations on radiance functions

Surface light fields $L(\mathbf{p}, \omega)$ define for every point $\mathbf{p}$ on the surface the outgoing radiance. This is a function that associates a color to any viewing direction ω. It represents the light emitted from $\mathbf{p}$ into direction ω. The domain for ω is the hemisphere of visible directions, *i.e.*, centered on $\mathbf{n}$, the normal to the surface at $\mathbf{p}$.

In our setting, the surface is a triangle mesh defined by its connectivity and its positions $\mathbf{p}_i$ and normals $\mathbf{n}_i$ at vertices v_i. The embedding is linear: a point with barycentric coordinates $(\alpha_1, \alpha_2, \alpha_3)$ in a triangle $\mathbf{t} = (v_1, v_2, v_3)$ is embedded at $\mathbf{p} = \alpha_1 \mathbf{p}_1 + \alpha_2 \mathbf{p}_2 + \alpha_3 \mathbf{p}_3$. The normal is also assumed to be linearly interpolated (up to normalization), which is a common approximation for rendering. A radiance function $L(\mathbf{p}_i, \omega)$ is defined at each vertex.

In this section, we define tools to calculate on radiance functions over the surface. They will be used to design a new metric for simplification (section 4) and to define improved rendering formulas (section 5). First, we define formulas for RF reflected around the local surface normal (section 3.1), as they exhibit higher spatial coherence [25]. Second, we derive triangle interpolation (section 3.2), which defines L at any $\mathbf{p}$ on the surface (spatial continuity). Third, we derive triangle extrapolation (section 3.4), which defines RF off the initial surface, which is useful since edge collapses change the geometry. The latter exploits a gradient of the RF on the surface (spatial derivative) which we first present in section 3.3. Finally we propose distance measurements between two RF (section 3.5) so as to serve our metric.

3.1 Reflected representation

Let $L(\mathbf{p}, \omega)$ be a RF defined on the hemisphere oriented with normal $\mathbf{n}$. We apply the reflection $R_{\mathbf{n}}$ with respect to $\mathbf{n}$ in order to get $\widetilde{L}(\mathbf{p}, \omega) = L(\mathbf{p}, R_{\mathbf{n}}\omega)$ which we call *reflected radiance function* (r-RF). This idea improves the spatial coherence for a large class of common materials. It has indeed been used efficiently for compact BRDF representation [22], image-based rendering by pre-filtered environment mapping [1], and compression of SLF [25]. We apply it for coherent interpolation and improved rendering quality.

The motivation is that the reflected $\widetilde{L}$ tend to change less than L when $\mathbf{p}$ varies, *i.e.*, over the surface. Fig. 2 illustrates this on a simple example: suppose two functions $L(\mathbf{p}_1, \cdot)$ and $L(\mathbf{p}_2, \cdot)$ (middle) were reconstructed from a similar material (say, a Phong-like reflectance model) and a point light source. They then exhibit specular peaks around the ideal reflection direction. Since the surface normals $\mathbf{n}_1$ and $\mathbf{n}_2$ diverge, $L(\mathbf{p}_1, \cdot)$ and $L(\mathbf{p}_2, \cdot)$ differ a lot. In contrast, $\widetilde{L}(\mathbf{p}_1, \cdot)$ and $\widetilde{L}(\mathbf{p}_2, \cdot)$ (bottom) are similar because the peaks are aligned with the lighting direction.

This holds for complex lighting environments (*e.g.*, many lights) resulting in complex outgoing radiance functions, provided that the common lighting environment is at infinite distance. Inter-reflections and auto-occlusions violate this condition in general, but it is respected locally when comparing nearby points on the surface: reflection *locally* increases the coherence. This is very important in our context because edge collapse (section 4) is a *local* process for simplifying preferably homogeneous regions, and rendering (section 5) *locally* interpolates RF in triangles. As a result, a region with homogeneous material and similar local environment results in homogeneous r-RF over the region.
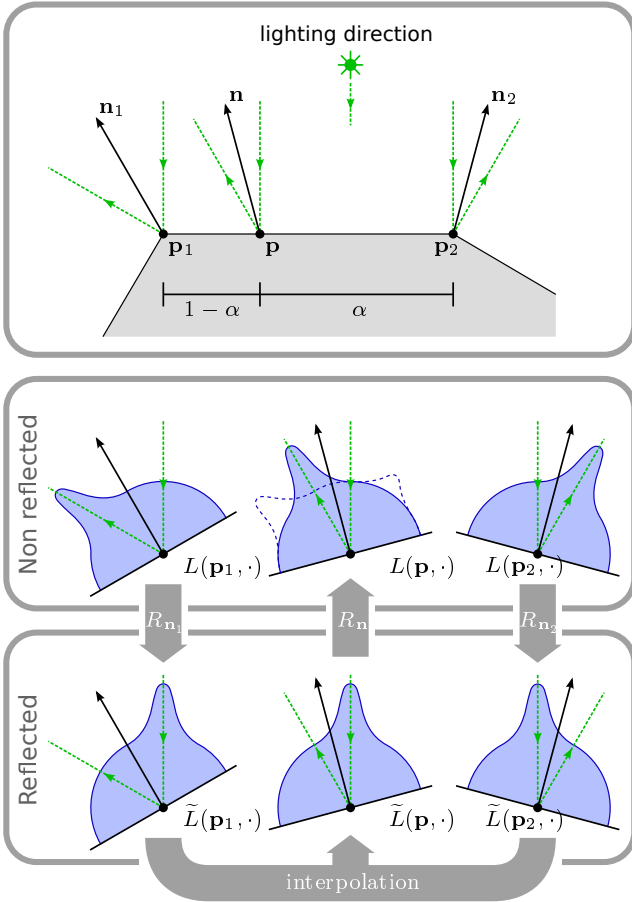
Fig. 2 Interpolation at point $\mathbf{p}$ of two radiance functions $L(\mathbf{p}_1,\cdot)$ and $L(\mathbf{p}_2,\cdot)$. Functions at vertices $\mathbf{p}_1$ and $\mathbf{p}_2$ are derived from a point light source at infinite distance and a Phong-like reflectance model. i) Each function is reflected w.r.t. its own normal. ii) Linear interpolation is performed (bottom). iii) The result is reflected w.r.t. the interpolated normal $\mathbf{n}$. Naive interpolation on non-reflected functions (dashed line) would be much less coherent.

3.2 Interpolation

Interpolation consists in deriving a RF at any position $\mathbf{p}$ in a triangle $\mathbf{t}$ from known RF at vertices $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$. As presented in Fig. 2, we define the r-RF at $\mathbf{p} = \alpha_1\mathbf{p}_1 + \alpha_2\mathbf{p}_2 + \alpha_3\mathbf{p}_3$ through linear interpolation:

$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \alpha_1\widetilde{L}(\mathbf{p}_1,\omega) + \alpha_2\widetilde{L}(\mathbf{p}_2,\omega) + \alpha_3\widetilde{L}(\mathbf{p}_3,\omega) \quad (1)$$

which formally defines a RF interpolation by

$$\begin{aligned} L_{\mathbf{t}}(\mathbf{p},\omega) = {} & \alpha_1 L(\mathbf{p}_1, R_{\mathbf{n}_1}R_{\mathbf{n}}\omega) \\ & + \alpha_2 L(\mathbf{p}_2, R_{\mathbf{n}_2}R_{\mathbf{n}}\omega) + \alpha_3 L(\mathbf{p}_3, R_{\mathbf{n}_3}R_{\mathbf{n}}\omega) \end{aligned} \quad (2)$$

This procedure tends to preserve specular peaks because it essentially interpolates functions of ω while naive interpolation (Fig. 2, middle, dashed line) would interpolate colors at fixed ω.

3.3 Gradient

If one fixes the viewing direction ω then equation (1) amounts to simple linear interpolation of colors over the triangle $\mathbf{t}$. Thus for fixed ω, the gradient w.r.t. $\mathbf{p}$ of $\widetilde{L}_{\mathbf{t}}$ is constant over $\mathbf{t}$. Differentiation on the surface [6] leads to

$$\nabla_{\mathbf{p}}\widetilde{L}_{\mathbf{t}}(\omega) = J(J^T J)^{-1}\begin{bmatrix} \widetilde{L}(\mathbf{p}_2,\omega) - \widetilde{L}(\mathbf{p}_1,\omega) \\ \widetilde{L}(\mathbf{p}_3,\omega) - \widetilde{L}(\mathbf{p}_1,\omega) \end{bmatrix} \quad (3)$$

where $J = \begin{bmatrix} \mathbf{p}_2 - \mathbf{p}_1, \mathbf{p}_3 - \mathbf{p}_1 \end{bmatrix}$. Note that $\nabla_{\mathbf{p}}\widetilde{L}_{\mathbf{t}}$ is a function defined over the hemisphere. It should be a Jacobian matrix but we consider the color channels separately (see section 7), so we compute one vector per channel.

3.4 Extrapolation

By using the gradient, equation (1) can be rewritten as

$$\widetilde{L}_{\mathbf{t}}(\mathbf{p},\omega) = \widetilde{L}(\mathbf{p}_1,\omega) + \nabla_{\mathbf{p}}\widetilde{L}_{\mathbf{t}}(\omega) \cdot (\mathbf{p} - \mathbf{p}_1) \quad (4)$$

Although the gradient lies in the triangle's plane and was designed to compute *interpolation*, this formula actually defines its extension to compute it for $\mathbf{p}$ that does not lie in the triangle, not even in the plane. Conversely to the straightforward equation (1), formulation (4) provides *extrapolation* by extending $\widetilde{L}_{\mathbf{t}}$ to 3D with constant gradient on the whole plane and a (spatially) constant (directional) color in the normal direction. We exploit this in section 4.

3.5 Distance measurements

We define the distance between radiance functions by

$$d\left(L(\mathbf{p}_1,\cdot), L(\mathbf{p}_2,\cdot)\right) = \left\| \widetilde{L}(\mathbf{p}_1,\cdot) - \widetilde{L}(\mathbf{p}_2,\cdot) \right\|_{L_2(\Omega)} \quad (5)$$

where

$$\|L(\mathbf{p},\cdot)\|_{L_2(\Omega)} = \left(\frac{1}{2\pi}\int_{\omega\in\Omega} L(\mathbf{p},\omega)^2 d\omega\right)^{1/2} \quad (6)$$

is the L_2-norm for square integrable functions on the hemisphere Ω. The choice of Ω will be discussed in section 4.1.

This measure, which integrates r-RF rather than RF, has several advantages:

- It is consistent with the above interpolation procedure: the average of $L(\mathbf{p}_1,\cdot)$ and $L(\mathbf{p}_2,\cdot)$ minimizes the sum of squared distances to $L(\mathbf{p}_1,\cdot)$ and $L(\mathbf{p}_2,\cdot)$.
- Distance between diffuse RF (*i.e.*, constant functions) reduces to distance between simple colors.
- $\|L(\mathbf{p},\cdot)\|_{L_2(\Omega)} = \|\widetilde{L}(\mathbf{p},\cdot)\|_{L_2(\Omega)}$ since they are defined on the same hemisphere.

4 Mesh simplification

A mesh is defined by its connectivity and its embedding, which is generally a set of attributes attached to the vertices. Mesh simplification based on edge collapse [11] essentially consists in two stages:

1. build a priority queue of all collapsible edges;
2. until the mesh is simplified enough, collapse the first edge and update the queue.

This algorithm is grounded on a priority criterion (for the edges) and an embedding strategy (for the vertex resulting from a collapse), which must both take into account all the attributes. We first present our radiance error metric in the context of half-edge collapse and then extend it to general edge collapse.

4.1 Radiance error metric for half-edge collapse

Half-edge collapse is represented in Fig. 3: the edge $v_0 v_1$ is collapsed onto v_1. We define the radiance error caused by the collapse as a perceived difference power of light emitted by the surface. Indeed, since L is a luminous flux (lumen) per unit solid angle and per unit surface area, the error E defined as follows has *squared lumen* for theoretic dimension.

$$E = \sum_{\mathbf{t}} Area(\mathbf{t}) \ d^2\big(L(\mathbf{p}_0, \omega), \ L_{\mathbf{t}}(\mathbf{p}_0, \omega)\big) \qquad (7)$$

where the distance is defined by equation (5), extrapolation is defined by equation (4), and the sum runs over the post-collapse modified triangles (Fig. 3 right).

This equation can be understood as follows. Each final triangle $\mathbf{t} = (v_1, v_2, v_3)$ contributes to the error in proportion to its area multiplied by the change of RF caused by the collapse. The change of RF is measured by the squared distance between the RF at position $\mathbf{p}_0$ before and after collapse. The RF at $\mathbf{p}_0$ is L before collapse and is extrapolated from the triangle $\mathbf{t}$ by $L_{\mathbf{t}}$ after collapse. Since the distance is measured at point $\mathbf{p}_0$, the hemisphere for integration is chosen to be Ω_0.
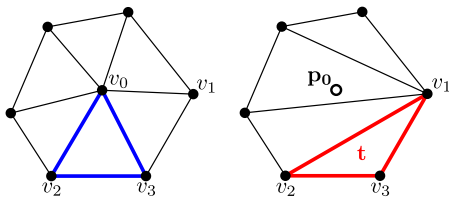
Fig. 3 Half-edge collapse of $v_0 v_1$ onto v_1. The point $\mathbf{p}_0$ (position of the vertex v_0 before collapse) is generally off the surface after collapse.

4.2 Coherence with diffuse color

To understand the relevance of E in terms of color, consider purely diffuse materials: equation (5) still applies and the distance reduces to an error between colors. As a consequence, E is the squared color error times the impacted area. Fig. 4 illustrates typical situations: the triangle t in red contributes to the error if its extrapolated color $L_{\mathbf{t}}$ (background) differs from the original color, *i.e.*, before the collapse. This is the case of $\mathbf{p}''_0$, as opposed to $\mathbf{p}_0$ and $\mathbf{p}'_0$ which perfectly match $L_{\mathbf{t}}$. E is a good measure of *visible* color changes because the gradient links color and geometry: the contribution at $\mathbf{p}''_0$ is positive although its original color is the same as $\mathbf{p}_1$, while the contribution at $\mathbf{p}'_0$ vanishes although its original color differs from that of $\mathbf{p}_1$.

Our metric can then be compared to previous errors for color attributes [8,12,15]. It combines all the following advantages:

- Neither local parameterization nor projection is needed.
- It can be extended to functional attributes like radiance.
- The combination with common quadratic geometric errors is invariant under scaling (see section 4.4).

In the results section (Fig. 9), we show competitive results compared to a state-of-the-art metric on colors.

4.3 Embedding and metric for edge collapse

Compared to half-edge collapse the more general edge collapse case requires an embedding strategy for all attributes ($\mathbf{p}$, $\mathbf{n}$ and L) of the vertex resulting from the collapse. We used the optimal position $\mathbf{p}$ as proposed in [7]. However, in order to define the normal $\mathbf{n}$, we
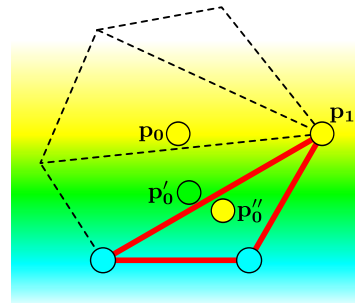
Fig. 4 Illustration of the contribution of a single triangle $\mathbf{t}$ (red) to the error of the collapse of $v_0 v_1$ onto v_1 (equation (7)). The shaded background represents its extrapolated color $L_{\mathbf{t}}$ while dots $\mathbf{p}_0$, $\mathbf{p}'_0$ and $\mathbf{p}''_0$ represent 3 possible positions and original colors (*i.e.*, at v_0 before collapse, see Fig. 3). The color error between a dot and the background is measured by d^2 in equation (7).
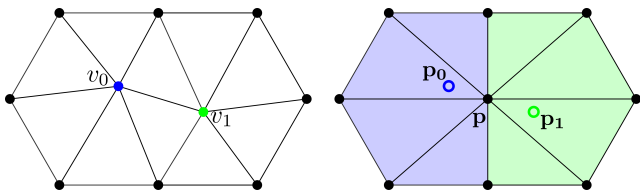
Fig. 5 Edge collapse.

don't use the extended QEM [8] because mixing normals and geometry is somehow arbitrary. Instead we project $\mathbf{p}$ onto the edge $\mathbf{p}_0\mathbf{p}_1$ and then linearly interpolate between $\mathbf{n}_1$ and $\mathbf{n}_2$ accordingly. The reflected radiance $\widetilde{L}(\mathbf{p}, \omega)$ is interpolated in the same way, by restricting equation (1) to one dimension along the edge. Unlike half-edge collapse, an explicit expression of $\widetilde{L}(\mathbf{p}, \omega)$ is required here. Thus a rotation-invariant spherical basis is needed (see section 6).

Finally, the RF error is computed by summing equation (7) applied once for $\mathbf{p}_0$ over the triangles depicted in blue in Fig. 5 and once for $\mathbf{p}_1$ over the triangles depicted in green.

4.4 Combination with geometric error

To be effective, simplification must consider all the attributes. We therefore linearly combine E with the memoryless variant of the standard QEM [12] because our radiance metric is also memoryless. Like the QEM, our metric has quadratic growth in the mesh size, so the balance between both does not depend on the geometric scaling. Still, the balance depends on the mesh density and on the color space encoding. In our examples, the radiance error is very low because color changes are gradual and specular effects impact only part of the hemisphere. So we empirically weighted 1% geometry and 99% radiance on all our examples.

4.5 Progressive meshes

Considering we have at our disposal an edge collapse metric and an embedding strategy, one can build progressive meshes with radiance attributes. A key application here is level-of-detail rendering. It consists in adapting the resolution to the rendering conditions (viewpoint, memory and time resources, screen-space resolution). The issue is to dynamically adapt the resolution in such a way that collapses and splits are not perceptible. Static pictures are therefore not suited for illustrating this, especially since radiance is visible when the viewpoint changes. The accompanying video (Online Resource 2) shows that this application is effective.

5 Rendering

Eventually, the object is rendered. In section 4, we determined a metric that considers interpolation of r-RF within each triangle. In this section, we define how to render accordingly and visually show why this interpolation is important.

A straightforward technique for shading triangles with RF defined on their vertices consists in evaluating the RF on these vertices within the *vertex shader* and then interpolating resulting colors at fragment level, *i.e.*, within the triangle. This approach works well when triangle meshes are extremely dense w.r.t. the viewpoint, *i.e.*, a triangle covers only a few pixels in screen-space. Conversely, when considering simplified meshes, triangles can be large in screen-space. This technique then tends to sweep out specular highlights, as shown in Fig. 6. One can notice that this is equivalent to per fragment naive interpolation of RF (Fig. 2, dashed line). It generates artifacts analogous to those of Gouraud-shading with vanishing highlights inside the triangles.

Rendering quality can significantly benefit from the RF interpolation proposed in section 3.2. In practice, the reflected functions $\widetilde{L}$ are stored and transferred to the GPU. The r-RF evaluation is now done in the *frag-*
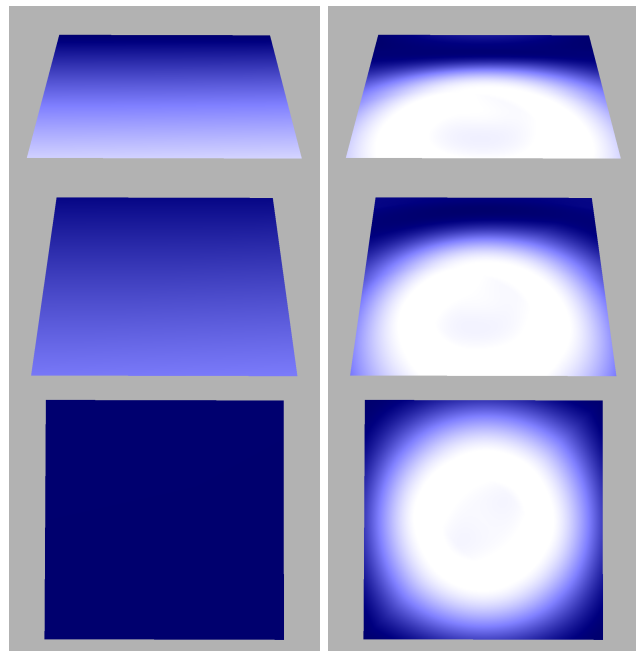
Fig. 6 Toy example generalizing Fig. 2: four radiance functions are stored on a four-vertex plane. Their normals diverge and the functions represent a diffuse + specular behavior (*i.e.*, a Phong-like BRDF). Left (naive interpolation): the specular highlight fades out at some angles. Right (equation (8)): the specular highlight is well interpolated.

ment shader as follows:

$$L_{\mathbf{t}}(\mathbf{p}, \omega) = \alpha_1 \widetilde{L}(\mathbf{p}_1, R_{\mathbf{n}}\omega)$$
$$+ \; \alpha_2 \widetilde{L}(\mathbf{p}_2, R_{\mathbf{n}}\omega) \; + \; \alpha_3 \widetilde{L}(\mathbf{p}_3, R_{\mathbf{n}}\omega) \quad (8)$$

which is actually equation (2) with r-RF on the right-hand side. The improvement of this r-RF interpolation over naive RF interpolation is analogous to the well-known improvement of Phong shading over Gouraud shading, but applied to stored RF. Fig. 6 shows a quad (2 triangles) with divergent normals: equation (8) prevents the specular peak to fade out inside the quad.

To evaluate equation 8 at fragment level, implementation needs to be adapted. The following data is required for each fragment:

- the interpolated position $\mathbf{p}$ and normal $\mathbf{n}$;
- barycentric coordinates α_1, α_2 and α_3;
- all three r-RF of its triangle ($\widetilde{L}$ at $\mathbf{p}_1$, $\mathbf{p}_2$ and $\mathbf{p}_3$).

To obtain the three r-RF per fragment, they are duplicated on the provoking vertex of the triangle in the *geometry shader* and exported as flat output attributes. We actually duplicate only references to the r-RF which are stored in arrays.

Note that switching from per-vertex to per-fragment evaluation impacts performance. It moves the complexity from $\mathcal{O}$(visible vertices) to $\mathcal{O}$(object-covered screen-space pixels). This implies that rendering speed increases when visualizing the object from a distance. Conversely, rendering speed decreases when visualizing a simplified mesh, *i.e.*, when triangles cover many pixels. It is however in the latter case that visual quality is improved by per-fragment evaluation. So eventually, the user can see the choice of per-vertex versus per-fragment evaluation as a trade-off between, respectively, rendering speed and quality.

6 Results

Basis functions. We experimented with two linear bases. First, a space of polynomials [19] that represent hemispherical functions by orthogonal projection on the tangent plane. These functions must be expressed in a local frame and the space is invariant under rotation around the normal. For all figures but Fig. 7 we used a polynomial basis (PB) of bi-degree $d = 4$ which require $\frac{(d+1)(d+2)}{2} = 15$ coefficients per channel and a rotation (from global to local frame). We also experimented with spherical harmonics [18] which represent functions on the entire sphere. They can be expressed either in a local or a global frame and the space is invariant under any rotation. Although functions are encoded on the entire sphere, they contain relevant information only for

607MB 152MB 149MB

0 [_____] Max

188MB 47MB 44MB
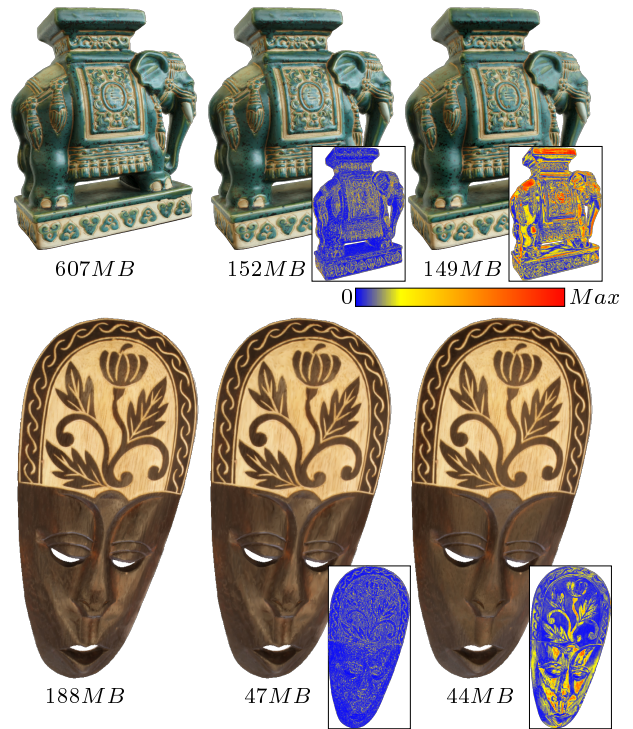
Fig. 7 The original (left) is reduced by mesh simplification (middle) and degree reduction (right). Memory reduction is similar. All objects use spherical harmonics in global frame. Elephant, respectively: $d = 6$ & $1M$ vertices, $d = 6$ & $250K$ vertices, $d = 2$ & $1M$ vertices. Mask, respectively: $d = 8$ & $193K$ vertices, $d = 8$ & $48K$ vertices, $d = 3$ & $192K$ vertices.

the visible hemisphere, so it is compliant with the computations of previous subsections. We show spherical harmonics (SH) of degree $d = 2, 3, 6$ and 8 expressed in global frame (Fig. 7), which require $(d+1)^2 = 9, 16, 49$ and 81 coefficients per color channel, respectively.

We emphasize that our algorithms do not depend on the basis chosen, neither on the basis degree, nor on the frame. We argue that the choice should be left to the final user according to the application needs. Simplification results only slightly differ according to the basis (see Online Resource 1). Equations (1) to (8) apply whatever the basis and the frame. However, for hemispherical functions, (1) to (4) can be computed for a fixed ω only because it requires rotations around arbitrary vectors. On the contrary, rotation-invariant spherical functions allow for an explicit expression of the results in the basis. In other words, equations on functions translate into equations on coefficients. A global frame is not mandatory: it makes most computations simpler, except reflection which is trivial in a local frame.

Spatial versus directional data reduction. Fig. 7 shows that, to save memory, it may be beneficial to simplify
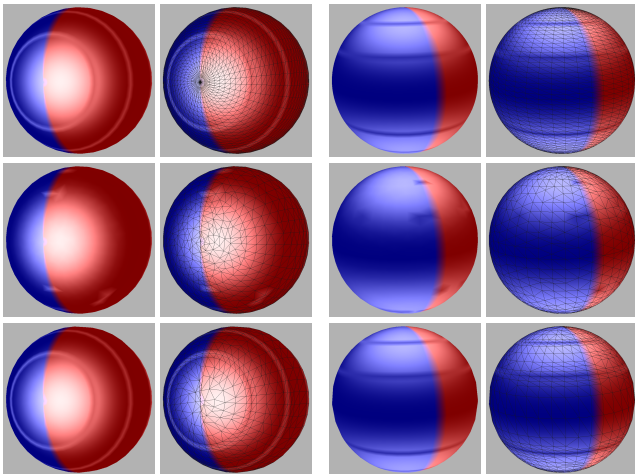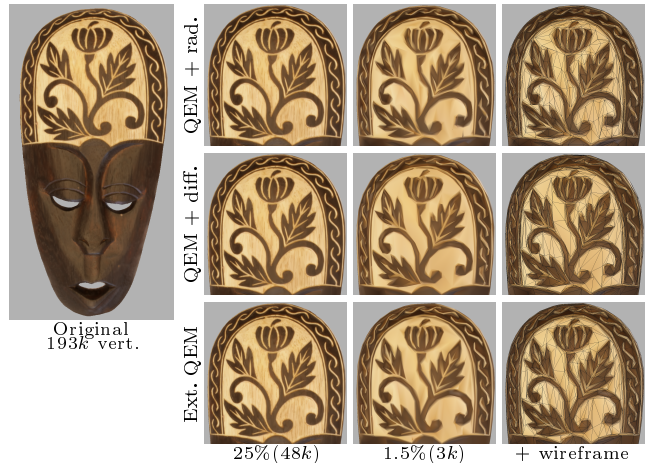
Fig. 8 Original data (top row) exhibits diffuse strips on specular material. Compared to radiance error (bottom), color error (middle) fails to preserve specular features. Simplification to 50% is processed through half-edge collapse. The color error is extended QEM [8] applied to the RF's average color.

Fig. 9 The mask forehead exhibits few geometry or reflection but strong color features. Our metric on radiance (top) and its application to diffuse colors only (middle) amount to similar results, and they compete with QEM extended to colors (bottom) [8], even for severe simplification. Note how well the triangles adapt to the features (right column). RF are encoded with $PB(d = 4)$ and respectively consume $42MB$, $10MB$ and $0.7MB$ of memory.

the mesh (spatial resolution) rather than to reduce the degree of the basis (directional resolution). On such glossy objects, preserving specular highlights is essential for perceiving the material and the geometry, as becomes clear on animated objects in the accompanying video (Online Resource 2). The colored errors emphasize that simplification tends to remove fine details scattered on the surface, while degree reduction tends to produce large errors on the highlights.

Comparison to color metrics. With existing techniques, the best one can do is to use color metrics applied on per-vertex diffuse (*i.e.*, non-directional) color. Fig. 8 illustrates the need for improvement over the latter on a synthetic example. Slight simplification is performed on a sphere alternating diffuse and specular materials (top). It shows that straightforward usage of a color metric is unable to detect and preserve specular features, while our radiance metric does (bottom).

To show a fair comparison of our metric w.r.t. state of the art methods we compare it with color metrics when applied on the diffuse color component of our objects. In this case, our metric reduces to a color distance (see section 4.2). The lower pair of Fig. 9 illustrates this on the mask forehead. Simplification at 25% still preserves fine wood veins; at 1.5%, only pronounced features are preserved. Our metric applied on the full radiance (top row) does not deteriorate the results compared to color metrics. Both our radiance metric (top) and its application on diffuse color (middle) compete with the extended QEM (bottom) on this material having predominant non-directional color features. Note

how the resulting triangles follow the color features at high simplification ratios (right column).

The mask face in contrast has a quite uniform color and few geometry but high specular effects, as shown by different viewpoints in Fig. 10. In such case the radiance metric (middle) improves over the metric on diffuse colors (bottom): the triangles on the eyebrow and the eyelid are much better adapted to the features.

Finally, we drove tests on another large and complex acquired dataset (double dragon, $921k$ vertices, Fig. 1) which combines geometry, color, and directional features. It resists to severe simplification and the different types of features have been preserved. A good balance between them is ensured by the geometry-radiance combination described in section 4.4.

Edge vs. half-edge collapse. In section 4, we discussed two embedding strategies: edge collapse and half-edge collapse. Fig. 7 shows edge-collapse whereas Fig. 9 and Fig. 10 show half-edge collapse. There are no significant differences between results: half-edge collapse generates more elongated triangles while edge collapse may mix colors. This did not result in issues on our examples so one or the other can be used safely.

7 Technical details

Color channels. Besides RGB, we tested CIELUV and CIELAB color spaces because they are perceptually uniform. Since we observed only minor differences (see

Fig. 10 Simplification of the mask model (original on top row) using our radiance metric (middle) and its restriction to diffuse color (bottom). Different viewpoints and close-ups show important light reflections on the face. The radiance metric better preserves important visual features (*e.g.*, eyebrow, eyelid). See caption of Fig. 9 for model details.

Online Resource 3), we worked in RGB endowed with the 2-norm. This norm has the advantage that the 3 color channels can be treated separately in equation (7).

Numerical integration of errors. The computation of errors (equation (7)) requires to integrate a mix of functions over a hemisphere. Deriving a closed form would be tedious, if even possible. When using polynomials, this is due to local frame alignments and domain intersections. When using spherical harmonics, it is due to the arbitrary orientation of the integration domain. Therefore we perform numerical integration using Lebedev quadrature on the sphere.

Time complexity. Simplifications typically take a few hours for our acquired data. The complexity is $\Theta(N(\log N + BP))$ where N is the number of vertices, B the number of basis functions, and P the number of integration points. The numerical integration (BP) actually dominates the priority queue update ($\log N$). Thus the algorithm is suitable and scalable for an offline process.

Data acquisition and reconstruction. The double dragon (Fig. 1), the mask (Fig. 9 and Fig. 10) and the elephant (Fig. 7) are acquisitions of real objects. Geometry is acquired with a 3D-scanner operating with structured light. It generates point clouds which are approximated by a triangle mesh using the Poisson surface reconstruction algorithm [13]. Radiance functions are fitted

on photographs that are taken with a hand-held high-resolution camera and projected onto the mesh [24]. The radiance functions attached to the plane and the spheres (Fig. 6 and 8 and accompanying video in Online Resource 2) are fitted on virtually acquired photographs of a synthetic scene characterized by point light sources and BRDF.

Exceeding hemispheres. RF may have to be evaluated outside of their hemisphere of definition. This happens i) during simplification when comparing RF with different normals, or ii) during visualization when a vertex is visible from under its tangent plane. To solve this problem, colors of the hemisphere's border are prolonged on the opposite hemisphere. This avoids popping artifacts during rendering (except at the opposite pole, which is never visible in practice).

8 Conclusion

In this paper, we treated the problem of the memory load of digitized surface light fields represented by radiance functions stored as attributes on mesh vertices. This load can be problematic for visualization and streaming. We first derived interpolation and gradient formulas as well as a distance measure for RF. Then, we defined a first RF-aware metric that we exploit in a mesh simplification algorithm. Our metric has proven to be a true added value w.r.t. existing techniques (*i.e.*, adaptations from metrics on colors) in terms of quality when directional features are present. When the RF are diffuse, our algorithm reduces to a color metric and competes with the state of the art.

As a result, the user now has a new choice to reduce the memory load: besides reducing the directional resolution (lowering the degree of RF functions), we provided an algorithm for reducing the spatial resolution while preserving geometric and directional features. We have shown that this may be preferable for higher overall realism. We point out that compression strategies for the set of RF [2,25] can be applied as a subsequent complementary stage in the processing pipeline.

As a complementary contribution, we defined how to improve rendering quality for local RF by determining smart interpolation of reflected RF. We have also shown that progressive meshes constitutes an effective application for level-of-detail rendering.

Based on our interpolation and distance measure, we think of dealing with RF stored in textures, including filtering and mip-mapping. One severe issue is the management of parameterization boundaries. Another problem for mip-mapping is the explicit averaging of several RF if hemispherical functions are chosen.

A more accurate view-dependent filtering also requires accounting for geometry-related masking effects.

We would also like to push the comparison further. Indeed, under a rendering perspective, simplification can be compared to compression techniques. However, whereas the memory load can be measured objectively, the visual quality is more subjective. To this end we consider that a perceptual study would be helpful.

Finally, another prospect is advanced signal processing on the mesh. Indeed, interpolation and gradient computations essentially provide spatial continuity for SLF. Investigating further differentiation on the surface could lead to processing through differential equations.

References

1. Cabral, B., Olano, M., Nemec, P.: Reflection space image based rendering. In: Proceedings of SIGGRAPH '99, pp. 165–170. ACM Press (1999). DOI 10.1145/311535.311553

2. Chen, W.C., Bouguet, J.Y., Chu, M.H., Grzeszczuk, R.: Light field mapping: Efficient representation and hardware rendering of surface light fields. In: Proceedings of the 29th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH '02, pp. 447–456. ACM, New York, NY, USA (2002). DOI 10.1145/566570.566601

3. Choi, H., Kim, H., Lee, K.: An improved mesh simplification method using additional attributes with optimal positioning. The International Journal of Advanced Manufacturing Technology **50**(1-4), 235–252 (2010). DOI 10.1007/s00170-009-2484-y

4. Cohen, J., Olano, M., Manocha, D.: Appearance-preserving simplification. In: Proceedings of SIGGRAPH '98, pp. 115–122. ACM Press (1998). DOI 10.1145/280814.280832

5. Coombe, G., Hantak, C., Lastra, A., Grzeszczuk, R.: Online construction of surface light fields. In: Proceedings of the Sixteenth Eurographics conference on Rendering, EGSR'05, pp. 83–90 (2005). DOI 10.2312/EGWR/EGSR05/083-090

6. Do Carmo, M.P.: Differential Geometry of Curves and Surfaces. Prentice-Hall (1976)

7. Garland, M., Heckbert, P.S.: Surface simplification using quadric error metrics. In: Proceedings of SIGGRAPH '97, pp. 209–216. ACM Press (1997). DOI 10.1145/258734.258849

8. Garland, M., Heckbert, P.S.: Simplifying surfaces with color and texture using quadric error metrics. In: Proceedings of the conference on Visualization '98, VIS '98, pp. 263–269. IEEE Computer Society Press, Los Alamitos, CA, USA (1998)

9. González, C., Castelló, P., Chover, M.: A texture-based metric extension for simplification methods. In: Proceedings of the Second International Conference on Computer Graphics Theory and Applications (GRAPP), pp. 69–76 (2007)

10. Gortler, S.J., Grzeszczuk, R., Szeliski, R., Cohen, M.F.: The lumigraph. In: Proceedings of SIGGRAPH '96, pp. 43–54. ACM Press (1996). DOI 10.1145/237170.237200

11. Hoppe, H.: Progressive meshes. In: Proceedings of SIGGRAPH '96, pp. 99–108. ACM Press (1996). DOI 10.1145/237170.237216

12. Hoppe, H.: New quadric metric for simplifying meshes with appearance attributes. In: Proceedings of the 10th IEEE Visualization 1999 Conference (VIS '99), VISUALIZATION '99, pp. 59–66. IEEE Computer Society, Washington, DC, USA (1999)

13. Kazhdan, M., Bolitho, M., Hoppe, H.: Poisson surface reconstruction. In: Proceedings of the fourth Eurographics symposium on Geometry processing, SGP '06, pp. 61–70 (2006)

14. Kim, C., Zimmer, H., Pritch, Y., Sorkine-Hornung, A., Gross, M.: Scene reconstruction from high spatio-angular resolution light fields. ACM Trans. Graph. **32**(4), 73:1–73:12 (2013). DOI 10.1145/2461912.2461926

15. Kim, H.S., Choi, H.K., Lee, K.H.: Mesh simplification with vertex color. In: Proceedings of the 5th international conference on Advances in geometric modeling and processing, GMP'08, pp. 258–271. Springer-Verlag, Berlin, Heidelberg (2008)

16. Lafortune, E.P.F., Foo, S.C., Torrance, K.E., Greenberg, D.P.: Non-linear approximation of reflectance functions. In: Proceedings of SIGGRAPH '97, pp. 117–126. ACM Press (1997). DOI 10.1145/258734.258801

17. Levoy, M., Hanrahan, P.: Light field rendering. In: Proceedings of SIGGRAPH '96, pp. 31–42. ACM Press (1996). DOI 10.1145/237170.237199

18. MacRobert, T.M.: Spherical harmonics: an elementary treatise on harmonic functions, with applications. Dover Publications (1948)

19. Malzbender, T., Gelb, D., Wolters, H.: Polynomial texture maps. In: Proceedings of SIGGRAPH '01, pp. 519–528. ACM Press (2001). DOI 10.1145/383259.383320

20. Nishino, K., Sato, Y., Ikeuchi, K.: Eigen-texture method: Appearance compression and synthesis based on a 3D model. IEEE Trans. Pattern Anal. Mach. Intell. **23**(11), 1257–1265 (2001). DOI 10.1109/34.969116

21. Ray, N., Nivoliers, V., Lefebvre, S., Lévy, B.: Invisible seams. In: Proceedings of the 21st Eurographics Conference on Rendering, EGSR'10, pp. 1489–1496. Eurographics Association, Aire-la-Ville, Switzerland, Switzerland (2010). DOI 10.1111/j.1467-8659.2010.01746.x

22. Rusinkiewicz, S.: A new change of variables for efficient BRDF representation. In: G. Drettakis, N. Max (eds.) Rendering Techniques '98, Eurographics, pp. 11–22. Springer Vienna (1998). DOI 10.1007/978-3-7091-6453-2_2

23. Schröder, P., Sweldens, W.: Spherical wavelets: efficiently representing functions on the sphere. In: Proceedings of SIGGRAPH '95, pp. 161–172. ACM Press (1995). DOI 10.1145/218380.218439

24. Vanhoey, K., Sauvage, B., Génevaux, O., Larue, F., Dischler, J.M.: Robust Fitting on Poorly Sampled Data for Surface Light Field Rendering and Image Relighting. Computer Graphics Forum **32**(6), 101–112 (2013). DOI 10.1111/cgf.12073

25. Wood, D.N., Azuma, D.I., Aldinger, K., Curless, B., Duchamp, T., Salesin, D.H., Stuetzle, W.: Surface light fields for 3D photography. In: Proceedings of SIGGRAPH '00, pp. 287–296. ACM Press (2000). DOI 10.1145/344779.344925

26. Wu, H., Dorsey, J., Rushmeier, H.: A Sparse Parametric Mixture Model for BTF Compression, Editing and Rendering. Computer Graphics Forum **30**(2), 465–473 (2011). DOI 10.1111/j.1467-8659.2011.01890.x