Vision, Modeling, and Visualization (2016)
M. Hullin, M. Stamminger, and T. Weinkauf (Eds.)

Gloss Editing in Light Fields

Yulia Gryaditskaya[1], Belen Masia[2], Piotr Didyk[1,3], Karol Myszkowski[1] and Hans-Peter Seidel[1]

[1]Max-Planck Institute for Informatics, Saarbrücken, Germany
[2]Universidad de Zaragoza, Zaragoza, Spain [3]Saarland University, MMCI, Saarbrücken, Germany

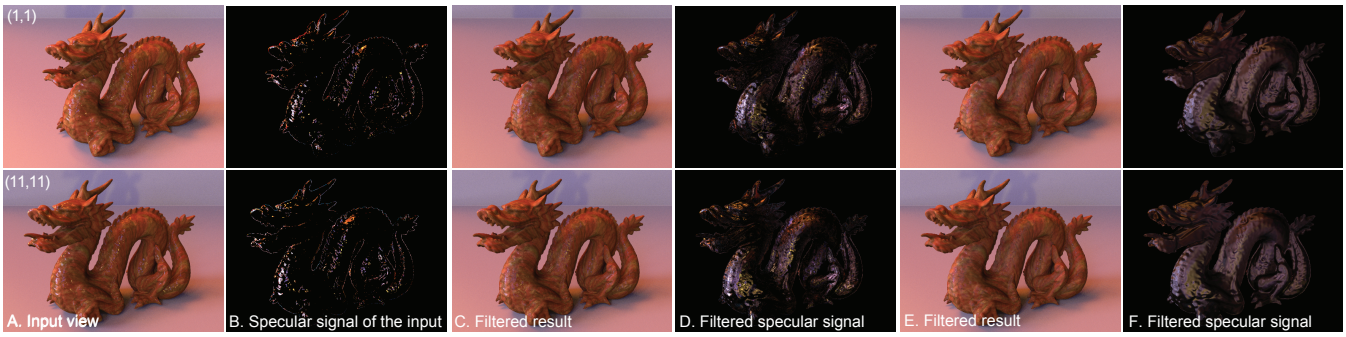

Figure 1: *Our multidimensional filtering approach allows angularly-coherent editing of glossy appearance in a light field (LF) by means of a user-specified roughness parameter* σ_v. In pairs, from left to right: *input LF, and two editing results showing increasing roughness (*$\sigma_v = 0.016$ *and* 0.064*, respectively). Each pair shows the actual LF view, and the corresponding specular component with increased contrast for better visualization. We show here two views of the LF (top and bottom rows); for videos of the full LF please refer to the supplemental material.*

Abstract

With the improvement of both acquisition techniques, and computational and storage capabilities, we are witnessing an increasing presence of multidimensional scene representations. Two-dimensional, conventional images are gradually losing their hegemony, leaving room for novel formats. Among these, light fields are gaining importance, further propelled by the recent reappearance of virtual reality. Content generation is one of the stumbling blocks in this realm, and light fields are one of the main input sources of content. As their use becomes more common, a key challenge is the ability to edit or modify the appearance of the objects in the light field. This paper presents a method for manipulating the appearance of gloss in light fields. In particular, we propose a multidimensional filtering approach in which the specular highlights are filtered in the spatial and angular domains to target a desired increase of the material roughness. The filtering kernel is computed based on surface normals and view direction. Our technique generates angularly-coherent plausible edits in both synthetic and captured light fields.

Categories and Subject Descriptors (according to ACM CCS): I.3.3 [Computer Graphics]: Picture/Image Generation—

1. Introduction

A light field (LF) is a general representation of the light traversing a scene, which provides information about its appearance observed from different directions [LH96]. This potentially enables reproducing all visual cues that can be observed in the real world, including motion parallax and accommodation, which are usually missing in conventional imaging pipelines. Given the completeness of LFs, they have a potential of becoming the ultimate visual data representation for digital imaging. The process of adapting them to current imaging pipelines is facilitated by recent work on LF acquisition [NLB*05, VLD*13], compression [VTMC11], and display [WLHR12], which makes a good prospect for developing an end-to-end LF pipeline in the near future. An important component of such a pipeline is a post-processing stage where artists can edit the content to alter its appearance. In this work, we address the problem of material editing, and more precisely, editing the apparent roughness of a surface, in structured LFs.

In this context, several techniques have been proposed for edit-

DOI: 10.2312/vmv.20161351

ing of 2D content [KRFB06, BBPA15], but the problem remains mostly unexplored for LFs. Although it seems feasible to use existing 2D techniques and apply them separately to different views, such an approach does not exploit the rich information encoded in LFs. As a result, complex interactions between scene geometry, viewing location, and light positions cannot be easily handled. Even though LFs enable reconstruction of such information by means of inverse rendering [HFB*09], the problem is usually difficult, and requires solving a highly underconstrained optimization problem. Fortunately, the accurate solution might not be needed. Since reflectance estimation is an underconstrained problem [DAW01], the human visual system (HVS) does not perform inverse rendering to obtain information about physical properties of materials. Instead, it relies on invariants and image properties such as highlight size or brightness [Fle12]. The HVS is also insensitive to large inconsistencies in illumination [OCS05], which further suggests that an approximate solution for material editing might be sufficient.

Inspired by these findings, our main observation is that an increase in surface roughness leads to changes in the size and brightness of specular highlights, and that such operations can be performed using simple image processing techniques, and in particular filtering. Consequently, we propose a new LF editing approach which modifies the apparent roughness of the surface by filtering the specular component of the LF. Since angular consistency is of high importance for material perception [WFEM10, DFY*11], filtering is performed jointly in the spatial and angular domains, with parameters optimized to closely match ground-truth results. Inspired by perceptual findings, our approach explicitly avoids the difficult problem of light source position estimation, and relies only on the depth map, which can be computed for LFs with a satisfactory fidelity [CLY*14, LCBKY15, ZSL*16, WSR16]. For this purpose, we adapt existing solutions paying special attention to the depth reconstruction in highlights, which are often prone to artifacts (Sec. 3). We demonstrate that even partial scene information, such as depth and surface normals, enables high quality editing of the object appearance (Sec. 4). One of the key benefits of using a filtering approach is that in difficult regions it provides graceful quality degradation without catastrophic failures. As a result, the edits do not introduce visible artifacts and inconsistencies (Sec. 5).

2. Related Work

Methods for material editing can be categorized according to what information they rely on. The most advanced techniques rely on having complete scene information, i.e., geometry, lighting, and reflectance properties of objects. In this category, interactive techniques for both parametric [CP06, NDM06, TGY*09], as well as measured [BAOR06] BRDFs were proposed. Some of them address also spatially-varying BRDFs [LBAD*06]. For any of such BRDF representations, even sparsely marked user edits must be meaningfully propagated to all image regions of similar appearance [PL07, AP08, NSRS13]. Another group of methods for material editing includes techniques that do not rely on an explicit BRDF representation, but work directly with a 3D scene where the user interactively deforms the shape of reflections or their position on the reflecting surface [ROTS09].

Our work is most related to image-based methods in which the difficulty of humans to perceive inconsistencies in material appearance has been exploited for editing materials [KRFB06], or even caustics [GSLM*08]. Such approaches benefit from the fact that the scene is observed from a single viewpoint, and thus, some unnoticeable inaccuracies can be present in depth or illumination estimation. These techniques often require additional user interaction [OCDD01, YTBK11]. Meanwhile, other methods assume that per-pixel normal and depth data are given together with the input image, e.g., for shading manipulation [VBFG12]. More recently, Boyadzhiev et al. [BBPA15] proposed a purely image-based technique to modify material properties by performing a subband decomposition and selectively sifting its coefficients.

The main difference between our technique and the work mentioned above is that our method does not require explicit information about geometry, material, or illumination. Instead, during our filtering step, we exploit the fact that such information is encoded in the LF implicitly. As a preprocessing step, we compute depth information and a specular-diffuse separation that will be used in the filtering step. With respect to the recent work by Boyadzhiev et al. [BBPA15], our technique provides different edits. While they modify high level properties, such as oiliness, glow or blemishes of the skin, we focus on the editing of glossy appearance, and roughness in particular. Furthermore, our filtering allows to propagate edits to views of the LF which had no gloss before. The approach by Boyadzhiev et al. is designed for 2D images, and it cannot support such operations in a trivial manner.

With the increasing interest in LFs, many techniques have been proposed to edit them. These include methods for morphing LFs [ZWGS02, WLL*05], interactive deformation of objects [COSL05], or propagation of local edits such as painting and scissoring [SK02]. Further efforts focused on hole-filling, object reshuffling and resizing, manipulating object depth, and parallax magnification [ZWS*16]. Recently, some work has targeted automatic edit propagation [JMG11] (with similar goals as in [AP08] for BRDFs) to ensure spatial and angular consistency, and different paradigms for efficient LF editing [JMB*14]. Although many new LF editing techniques have been recently proposed, to our knowledge no method exists for gloss appearance editing in LFs.

3. Light Field Preprocessing

To run our gloss editing algorithm we require normal maps of the LF views, as well as a separation of the specular and diffuse components. In this section we describe how depth maps can be estimated, how normals can be computed from these depth maps, and how to obtain the separation for the specular and diffuse components.

3.1. Depth Estimation

There has been a significant progress in methods estimating depth information from LF data [KZP*13, HP14, CLY*14, LCBKY15, TSM*15, ZSL*16, WSR16]. We use the approach by Chen et al. [CLY*14] as it provides good reconstructions at object edges, which is important to our algorithm. After computing depth map, we perform an additional step that improves normal estimation and angular coherence. For completeness, we provide a brief overview of the method by Chen et al., followed by details of our extension.

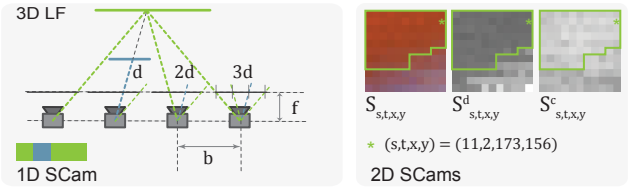

Figure 2: *Illustration of 1D and 2D SCams.* ***Left****: The 3D LF consists of two planar objects (green and blue) at two different depths. A 1D SCam is shown for the scene point where the dashed lines (rays) converge, as the image of the point in the different views. In the case of a 4D LF each SCam would thus be 2D, and built in an analogous manner.* ***Right****: Example SCams S, S^d and S^c (Eq. 3) computed for the point of the 'dragon' scene. The region marked with green outline corresponds to $S^*_{s,t,x,y}$.*

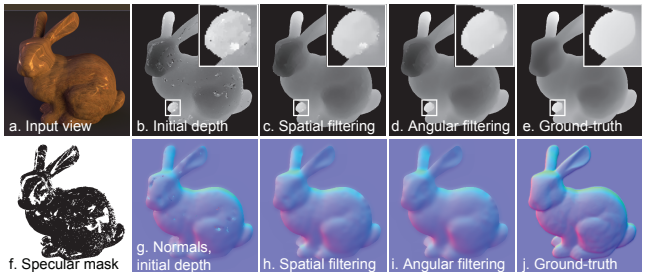

Figure 3: *Illustration of the depth maps for each of the steps in Sec. 3.1. The input LF consists of 15x15 views. The second row shows our estimated mask of the specular regions and the normal maps, computed as in Sec. 3.2. The maps show the values in the masked region, but the masks are not used for the steps in Sec. 3.*

3.1.1. Initial Disparities Estimation

In stereo image pairs, depth is estimated by finding correspondences between the points of two views. The correspondences are defined by the displacement (disparity) of the same object point between the views. The disparity d and depth Z values are linearly related:

$$Z = bf/(b-d), \tag{1}$$

where f is a focal length and b is a baseline, both in pixels units.

In a structured LF L, the displacement between cameras is uniform, and therefore, a single disparity value d provides a correspondence between all instances of the same scene point. Each point in a 4D LF L can be represented by four coordinates (s,t,x,y), using two plane parametrization, where (s,t) define the view and (x,y) the spatial position within this view; dimensions of the LF are thus $N_s \times N_t \times N_x \times N_y$. In the absence of occlusions, the instance of the point $L(s,t,x,y)$ in other views can be obtained from the offset $(\Delta s, \Delta t)$ between of the views:

$$L(s+\Delta s, t+\Delta t, x+\Delta s d, y+\Delta t d) = L(s,t,x,y). \tag{2}$$

In the presence of occlusions it might correspond to a different scene point (Fig. 2). The image $S_{s,t,x,y,d} = \bigcup_{(\Delta s,\Delta t)} L(s+\Delta s, t+\Delta t, x+\Delta s d, y+\Delta t d)$ defined by the relation (2), without consideration of occlusions, is called a surface camera (SCam) [YMG04].

Since a diffuse Lambertian object has the same appearance from all viewing directions, depth estimation algorithms typically try to find, for each point (s,t,x,y), the disparity value d, which gives the most uniform—in terms of color—SCam $S_{s,t,x,y,d}$. This works well in most cases, but, on object edges and in the presence of occlusions, the SCam might be divided into several regions. Thus, one is interested in finding a set of pixels $S^*_{s,t,x,y,d} \subseteq S_{s,t,x,y,d}$, which are the actual instances of the scene point (s,t,x,y) (Fig. 2 right). Chen et al. [CLY*14] determine $S^*_{s,t,x,y,d}$ as the pixels with the most similar color values to a color of reference pixel (s,t,x,y). To this end, a bilateral consistency metric (BCM) for each pixel is computed with respect to a reference pixel based on spatial and color distances. Then, $S^*_{s,t,x,y,d}$ is obtained by thresholding the values of BCM. In the BCM proposed by Chen et al., we set the color variance and spatial variance to $5/255$ and $\lfloor \max(N_s,N_t)/2 \rfloor$. Finally, for each test value $\tilde{d}$ the color *consistency* measure $\mathbf{c}_{\tilde{d}}(s,t,x,y)$ is computed on the region $S^*_{s,t,x,y,\tilde{d}}$, based only on color distances. Then, d is the optimal value of the consistency measure: $\mathbf{c}(s,t,x,y) = \max_{\tilde{d}} \mathbf{c}_{\tilde{d}}(s,t,x,y)$.

3.1.2. Final Disparities Estimation

The depth estimation algorithm described in Section 3.1.1 computes disparity estimates $\mathbf{d}$ for each point of the input LF. The estimation might be noisy and not angularly consistent. It may also contain errors in specular regions (see Fig. 3b). Most solutions which perform regularization or filtering to avoid these problems involve a MRF formulation to solve an energy minimization problem [TSM*15, WSR16, LCBKY15]. However, a discrete labeled disparity map, obtained from the optimization, is not optimal for our subsequent normal computation. We thus avoid this solution and rely on the fact that we have a good detector of inaccurate depth map regions: noisy pixels or pixels belonging to specular regions will have low values of the consistency measure $\mathbf{c}$ discussed in Sec. 3.1.1. The *unreliable* pixels can be defined as the ones with $\mathbf{c} < \theta$, where θ is a threshold value (in our implementation $\theta \in [0.8, 0.9]$) . Once we have detected the regions where depth is inaccurate, we use a strategy similar to [LLK*02] to propagate depth from reliable to unreliable regions within each view of the LF. For this propagation we rely on the idea that the depth of unreliable pixels can be estimated as the average of depth values of reliable pixels in some window around the unreliable pixel. We achieve this with a push-pull algorithm. To this end, we first compute a Gaussian pyramid based on reliable pixels, and then, propagate depth information in a top-down fashion to unreliable pixels from coarser levels. As a result, the depth map is significantly improved (compare Figs. 3b and 3c). This type of propagation can lead to errors on objects edges. Moreover, within-view propagation might lead to angularly inconsistent depth estimates. However, most regularization or filtering solutions are designed to work *within each view separately*, since most depth estimation algorithms seek a depth map only for the central view.

To obtain angular consistency and alleviate the errors on the edges, we perform an angular filtering step, which is performed *within SCams*. We exploit the fact that the scene points which have a specular reflection in certain views will not have them in some others. For points with specular reflection, disparity estimates are obtained through within-view propagation, while the rest of the points have

been obtained as described in (Sec. 3.1.1). Since the latter are more reliable than the former, we want to propagate angularly the disparity estimates with high values of the consistency measure to those, where the disparity has been estimated via within-view propagation.

For each reliable pixel (x,y) within each view (s,t), we compute the SCams from the disparity $\mathbf{d}$ and consistency $\mathbf{c}$ maps (Fig. 2):

$$S^d \equiv S^d_{s,t,x,y,d} = \bigcup_{(\Delta s,\Delta t)} \mathbf{d}(s+\Delta s, t+\Delta t, x+\Delta s d, y+\Delta t d) \quad (3)$$

$$S^c \equiv S^c_{s,t,x,y,d} = \bigcup_{(\Delta s,\Delta t)} \mathbf{c}(s+\Delta s, t+\Delta t, x+\Delta s d, y+\Delta t d), \quad (4)$$

where $d = \mathbf{d}(s,t,x,y)$. Instead of propagating d, we first compute an average $\bar{d}$ in S^d over the reliable pixels with similar disparity values: $\bar{d} = \sum_j w_{ij} S^d(j)$, where i is the index in S^d of the pixel (s,t,x,y). This step is performed to account for small errors in disparities estimation obtained as described in Sec. 3.1.1. During the averaging, the pixels are weighted by:

$$w_{ij} = e^{-\frac{w_d^2}{2\sigma_d^2}} \max\left(0, \frac{S^c(j)-\theta}{1-\theta}\right) \max\left(0, \frac{S^c(i)-\theta}{1-\theta}\right), \quad (5)$$

where w_d is the difference between disparity estimates, σ_d defines how similar two disparity values should be to be considered as two estimates for the same scene point. The other terms ensure we do not take into account pixels with unreliable depth estimates (w_{ij} is zero if consistency value $S^c(j)$ or $S^c(i)$ is less than the threshold θ).

Once the averaged disparity value $\bar{d}$ is obtained we want to propagate it to the other pixels $j \in S^d$:

$$S^d(j) = \bar{d} w^p_{ij} + S^d(j)(1-w^p_{ij}), \quad (6)$$

where we perform linear interpolation between the average $\bar{d}$ and the old disparity value $S^d(j)$ based on how reliable and how similar its value to $S^d(i)$. To do so, we now compute the weight w^p_{ij} for propagation:

$$w^p_{ij} = e^{-\frac{w_d^c}{2\sigma_d^2}\frac{\theta^2}{(2S^d(j))^2}} \max\left(0, \frac{S^c(i)-\theta}{1-\theta}\right), \quad (7)$$

where the exponential term determines to which pixel to propagate, and the second term as before states that the propagation should not be done if $i = (s,t,x,y)$ has low value of consistency measure. The term $\frac{\theta^2}{(2S^d(j))^2}$ influences the value of σ_d, so that the disparity values of unreliable pixels j could differ from $S^d(i)$ more.

3.2. Normals Computation

The technique described in Sec. 3.1 computes a disparity value for each pixel of each view in the LF, $\mathbf{d}$. To compute the corresponding normals, we first convert the disparity value $d \in \mathbf{d}$ to depth value Z using Eq. (1). We further compute horizontal (Gx) and vertical (Gy) gradients using forward differences. The tangent plane for a point is given by vectors $[1,0,Gx(s,t,x,y)]$ and $[0,1,Gy(s,t,x,y)]$, and the normal is computed as the cross product of these two vectors. Finally, we apply a guided filter [HST13] to the computed normal map with the depth map as a guidance image, to obtain a smooth normal map, and normalize vector lengths to one (Fig. 3 g-j).

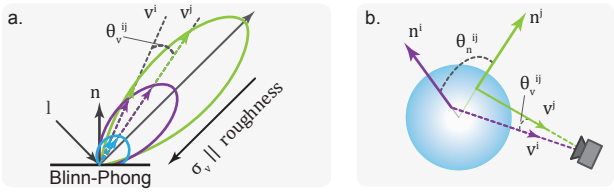

Figure 4: ***a.*** *Reflectance lobes at a certain scene point, with normal n and light direction l, for different roughness values. Vectors v^i and v^j illustrate view directions of the scene point to two views of the LF, i and j.* ***b.*** *Normals and view directions for two scene points within one view.*

3.3. Specular and Diffuse Component Separation

For a review of different approaches for specular separation please refer to [ABC11]. We select an approach which does not require any assumptions on illumination or any type of hallucination for missing information [CKS*05, TSW*15].

The specular separation is performed under the assumption that the point containing the specular reflection in one view will be observed as diffuse in some other. We further assume a Lambertian model for the diffuse component of the surface BRDF. To find all the instances of the point (s,t,x,y) with disparity value $d = \mathbf{d}(s,t,x,y)$ one needs to find the region $S^* \equiv S^*_{s,t,x,y,d}$ (Sec. 3.1.1). Compared to Sec. 3.1.1 we now know the disparity estimates $S^d_{s,t,x,y,d}$ (Eq. 3) and can immediately obtain S^* by taking the pixels with depth estimates similar to d. To an accuracy of noise level, the diffuse component can be estimated as the RGB value of the pixel with the lowest intensity value within S^*. Following Lin et al. [LLK*02], we consider all unreliable pixels, defined in Sec. 3.1.2, as specular.

4. Filtering

We pose the process of editing glossy appearance as a filtering problem, in which the edited value of each pixel in the LF is computed from the values in neighboring pixels according to a *roughness* parameter specified by the user. In a LF, neighboring pixels refer to both angular and spatial neighbors, i.e., filtering should be done in both the spatial and angular dimensions, combined, since the shape and appearance of the highlight is affected by both domains. We describe and justify the filtering approach below. Note that we filter only the specular component of the input LF (or of a masked region of it). The object in LF can be selected for example using an interactive technique [WSG13]. Thus, when we refer to LF values we will be referring only to the specular component of the LF. However, we maintain the notation of the previous section for clarity.

Angular filtering We exploit here that the LF contains instances of a scene point as observed from different view directions. For each point (s,t,x,y) of the input light field L with the disparity $d = \mathbf{d}(s,t,x,y)$, these instances form the set $S^* \equiv S^*_{s,t,x,y,d}$, defined in Sec. 3.1.1. The pixels of S^* share the same depth, normal, and directions to the light sources, while the direction to each camera (or viewpoint) v_j varies.

Our task is to obtain the new intensity value for a point from the input LF, which implicitly encodes information about the light

sources, and the known information about varying view directions. This can be formulated as a convolution of the input specular signal with some view-dependent weights, yielding the edited signal $\tilde{S^*}$. This convolution is done *per SCam of the LF* as follows:

$$\tilde{S^*}(i) = \sum_{j \in \hat{S}^*} \hat{S}^*(j) k^a_{ij}(\sigma_v), \quad \text{for each } i \in S^*, \tag{8}$$

where $\mathbf{k}^a$ is the convolution kernel, $\hat{S}^*$ are the pixels of S^* with values greater than zero. For the computation of the kernel weights k^a_{ij} we consider the following: *(i) two pixels i and j should share more information (i.e., weights will have higher values) if their view directions are more similar, and (ii) the higher the roughness value we want to achieve, the more information will be shared between the pixels with similar view directions*. This is illustrated in Fig. 4a, where we depict reflectance lobes for different roughness values, together with view directions for a certain scene point as captured in two views of the LF, v^i and v^j. For the different roughness values, one can see that, the smaller the angle $\theta^{i,j}_v$ between camera directions ($\theta^{i,j}_v = \arccos(v^i \cdot v^j)$), the more similar the reflectance values of the pixels will be; and that, the higher the roughness, the more similar the reflectance values as well, as explained in (i) and (ii) above. Taking this into account, the weights of the convolution kernel for the computation of pixel i are defined as:

$$k^a_{ij}(\sigma_v) \equiv k^a(v_i, v_j, \sigma_v) = \exp\left(-\frac{\theta^{i,j\,2}_v}{2\sigma_v^2}\right) \omega_{ij}, \tag{9}$$

where σ_v relates to the desired roughness. For the base version of our weights we set ω_{ij} to 1; its actual value is introduced next.

The question arises if there is any other information which can be used without implicit computation of the scene lighting. Let us consider the widely used Cook-Torrance BRDF model:

$$\rho(i,l,v) = \frac{D(l,v)F(l,v)G(l,v)}{4(v \cdot n)(l \cdot n)}, \tag{10}$$

where $D(\cdot)$ is a microfacets distribution function, $F(\cdot)$ is a Fresnel term, $G(\cdot)$ is a geometrical attenuation term, l is the light source direction, and i is some scene point. All the terms, except for the $(v \cdot n)$ term in the denominator, depend on the light source direction. While performing a convolution we can account for this term by multiplying the input signal at j by $(v^j \cdot n^j)$ and dividing the result of the convolution by $(v^i \cdot n^i)$. That corresponds to setting weights ω_{ij} in Eq. (9) to the following:

$$\omega_{ij} \equiv \omega(v_i, v_j) = \frac{(v^j \cdot n^j)}{(v^i \cdot n^i)}. \tag{11}$$

Spatial filtering For different points within each view of the LF, the normals, camera directions and light source directions might differ. However, in a small neighborhood around a given surface point one can assume a constant light source direction. Thus, the similarity between pixels is determined not only as a difference between view directions, but also as a difference between normal vectors. This is also modeled as a convolution, involving view- and normal-dependent kernel weights k^s_{ij} and yielding the edited signal $\tilde{L}_{s,t}$. This convolution is done *per view of the LF* as follows:

$$\tilde{L}_{s,t}(i) = \sum_{j \in \hat{L}_{s,t}} \hat{L}_{s,t}(j) k^s_{ij}(\sigma_v, \sigma_n), \quad \text{for each } i \in L_{s,t}, \tag{12}$$

where $\hat{L}_{s,t}$ are pixels of the view $L_{s,t}$ with values greater than zero. The kernel weights are computed as follows:

$$k^s_{ij}(\sigma_v, \sigma_n) \equiv k^s(v_i, v_j, n_i, n_j, \sigma_v, \sigma_n) = \tag{13}$$
$$= \exp\left(-\frac{(\theta^{i,j}_v)^2}{2(\sigma_v)^2} - \frac{(\theta^{i,j}_n)^2}{2(\sigma_n)^2}\right) \omega_{ij},$$

where $\theta^{i,j}_n = \arccos(n_j \cdot n_i)$, $\theta^{i,j}_v$ is the same as in Eq. (9), but computed over a different set of LF points: $i \in L_{s,t}$ and $j \in \hat{L}_{s,t}$ (see Fig. 4b). Parameters σ_v and σ_n control how rough the final material will appear in the LF: *the higher the roughness value we want to achieve, the more information will need to be shared between the pixels with similar view directions and normal vectors*. This is achieved by increasing σ_v and σ_n.

Combined Filtering One can observe that in the case in which i in Eq. (13) is set to go over S^*, and j to go over $\hat{S}^*$, then $k^s(\sigma_v, \sigma_n) = k^a(\sigma_v)$, since in this case $\theta^{i,j}_n = 0$. This allows us to define the universal kernel weights $k_{ij}(\sigma_v, \sigma_n)$ by Eq. (13), such that

- $k_{ij}(\sigma_v, \sigma_n) = k^a(\sigma_v)$ if $i \in S^*$ and $j \in \hat{S}^*$,
- $k_{ij}(\sigma_v, \sigma_n) = k^s_{ij}(\sigma_v, \sigma_n)$ if $i \in L_{s,t}$ and $j \in \hat{L}_{s,t}$.

In this case the convolution is done *per LF* and takes the form: $\tilde{L}(i) = \sum_{j \in \hat{L}} \hat{L}(j) k_{ij}(\sigma_v, \sigma_n)$, for each $i \in L$, where $\hat{L}$ are pixels of L with values greater than zero, and i denotes each of the pixels in light field L.

4.1. Filtering parameters and validation

In order to validate the multidimensional filtering, as well as to study the relation between σ_n and σ_v, we search for optimal values of these parameters by formulating an optimization problem.

Optimal filtering parameters We minimize the luminance differences between a target material with roughness r_2, and an input material with roughness r_1 filtered to obtain the target one. To avoid convergence to local minima, this optimization is performed in two steps: We first solve for σ_v and then for σ_n.

When solving for σ_v, we only perform *angular* filtering. Further, we select SCams $\mathcal{S}$ that contain the peak of the specular reflection. This ensures that our optimization results will not be biased by the SCams with partial information about lighting. We demonstrate in the next section that such SCams result in missing energy in the filtered LF. Consequently, we minimize the luminance differences between the specular component of the target SCam, $S^*_{r_2}$, and that of the filtered SCam, $\tilde{S}^*_{r_1}$, for all SCams in subset $\mathcal{S}$:

$$\min_{\varepsilon, \sigma_v} \sum_{S^*_{r_2} \in \mathcal{S}} \sum_{i \in \tilde{S}^*_{r_2}} \left(S^*_{r_2}(i) - \varepsilon \sum_{j \in \hat{S}^*_{r_1}} k^a_{ij}(\sigma_v) \hat{S}^*_{r_1}(j) \right)^2, \tag{14}$$

where ε is a global normalization factor. Once we have σ_v and ε, we obtain σ_n by minimizing the differences over the specular regions of the full LF:

$$\min_{\sigma_n} \sum_{i \in L^*_{r_2}} \left(L_{r_2}(i) - \varepsilon \sum_{j \in \hat{L}_{r_1}} \hat{L}_{r_1}(j) k_{ij}(\sigma_n, \sigma_v) \right)^2, \tag{15}$$

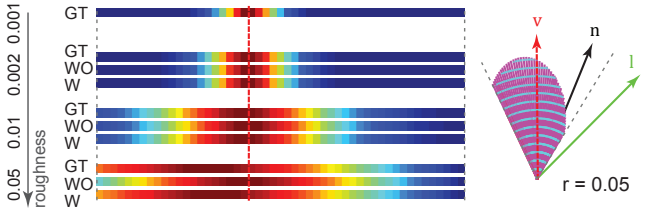

Figure 5: *Evaluating the contribution of the weighting factor ω_{ij} on a 1D SCam of a LF with $N_s = 50$. **Left:** An input LF with a roughness $r = 0.001$, is filtered with different parameters to obtain different target roughnesses (0.002, 0.01 and 0.05). For each target roughness, the three 1D SCams show the target (ground truth) SCam (GT), the filtered SCam without ω_{ij} (WO), and the filtered SCam with ω_{ij} (W). Note that ω_{ij} is required to capture the asymmetry of the ground truth SCam. **Right:** Illustration of light reflected at the points sampled by the 1D SCam (for the specific case of $r = 0.05$), for a scene point with normal n and light direction l (violet for actual values and cyan for filtered values).*

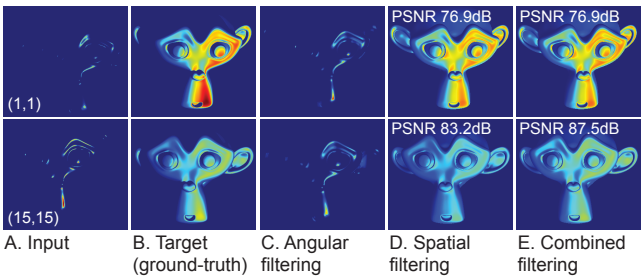

Figure 6: *Evaluation of angular, spatial and combined filtering on a 4D LF (15x15 views; the figure shows the heatmaps for the luminance channel of the views (s,t)=(1,1)/(15,15)). The input LF (A) has a roughness value r = 0.001, and the target roughness is 0.1(B). Values of σ_v and σ_n are obtained as described in Sec. 4.1. The light fields (B,D,E) are normalised to match the mean luminance of the view (s,t) = (1,1).*

where k_{ij} is given by Eq. (13). We solve Eqs. (14,15) using the implementation of the trust region-reflective algorithm from the MATLAB Optimization Toolbox.

We solved these optimization problems for several target roughness values to uniformly cover the range of possible roughness values and observed that consistently we obtained $\sigma_n \approx \sigma_v/2$. That is in accordance with the following intuition: If we consider the case where the normal, light source direction and view direction are lying on the same plane, a change of the normal in this plane by some angle γ will lead to the half angle changing by γ. At the same time, a change of a view direction by γ leads to a change in the half angle by $\gamma/2$. To obtain all results in Sec. 5, we fix the ratio between these parameters, so that $\sigma_n = \sigma_v/2$.

Validation To assess the contribution of our weighting factor ω_{ij} (defined in Eq. 11), in Fig. 5 we show the results of filtering for different target roughness values, with and without this factor. In the figure we depict a 1D SCam of the LF. Note that without this weighting factor the filtering can not achieve the asymmetry which is inherent to large roughness values.

Once we have shown that the weighting factor ω_{ij} is required to properly depict the appearance, we evaluate the contribution of the different types of filtering, i.e., considering only angular, spatial, or combined filtering. In Fig. 6 we show the results obtained with each type. If angular filtering alone is performed, for it to be correct, an SCam should cover the full specular lobe. Otherwise, filtering will lead to a lack of energy compared to the ground truth (see Fig. 6C vs. B). Spatial filtering within a single image allows to correctly modify the appearance of the specular component within a certain neighborhood. However, the closer the light source is to the surface, the less reliable the spatial propagation becomes for distant points. Thus, the spatial filtering might not be sufficient and may lead to inconsistent appearance of angular-dependent effects (see Fig. 6 columns D vs. B). Accounting for information both angularly and spatially ensures a more correct distribution of energy over the LF (Fig. 6 column E).

5. Results and limitations

We demonstrate our approach on three sample scenes. The full preprocessing pipeline is shown in Fig. 7: Depth is estimated for each view (Fig. 7E), followed by normals computation (Fig. 7F) and separation of specular and diffuse components (Fig. 7C and D). In Figs. 8 and 1 we show filtering results for three different output roughness values for each of input scenes. For each result we show both the filtered specular layer and the composited result; when merging both layers, the filtered layer is scaled so that the mean values of the input and filtered specular layers match. Please refer to the supplemental material for videos showing the full light fields of final and intermediate steps for all scenes. Our technique yields angularly-coherent, plausible edits of the appearance of glossy objects. In Fig. 9 and supplemental videos, we show side-by-side comparisons with ground-truth, as well as error maps computed using the SSIM [WBSS04] between the filtered and ground-truth results. Note that the major differences are caused by imprecision of depth reconstruction and not by the filtering. Additionally, varying the weight of the specular layer while compositing one can achieve more/less metallic appearance of objects (Fig. 10 and supplemental).

We also show that simple 2D images manipulation is insufficient for editing a material's roughness. The recent work of Boyadzhiev et al. [BBPA15] enables single-image modification of some material properties. In terms of gloss appearance it can make the material more or less shiny; we show results of both operations in Fig. 11 (a and b) for our *dragon* scene. With this technique, not only the highlight is modified, but the texture is also affected (see, e.g., Fig. 11b and supplemental). Further, while on a single image the results of both techniques may not be strikingly different, the difference becomes clear when observing the whole LF (please see supplemental videos). Finally, in Fig. 11c we plot the errors of this technique, under these assumption that the decomposition allows to process the highlight separately from the texture. The figure shows that when highlights get brighter or dimmer, the shape of the lobe does not change correctly. While our technique allows to model that behaviour correctly (Sec. 4.1 and Fig. 5).

The major limitation of our approach comes from depth estima-

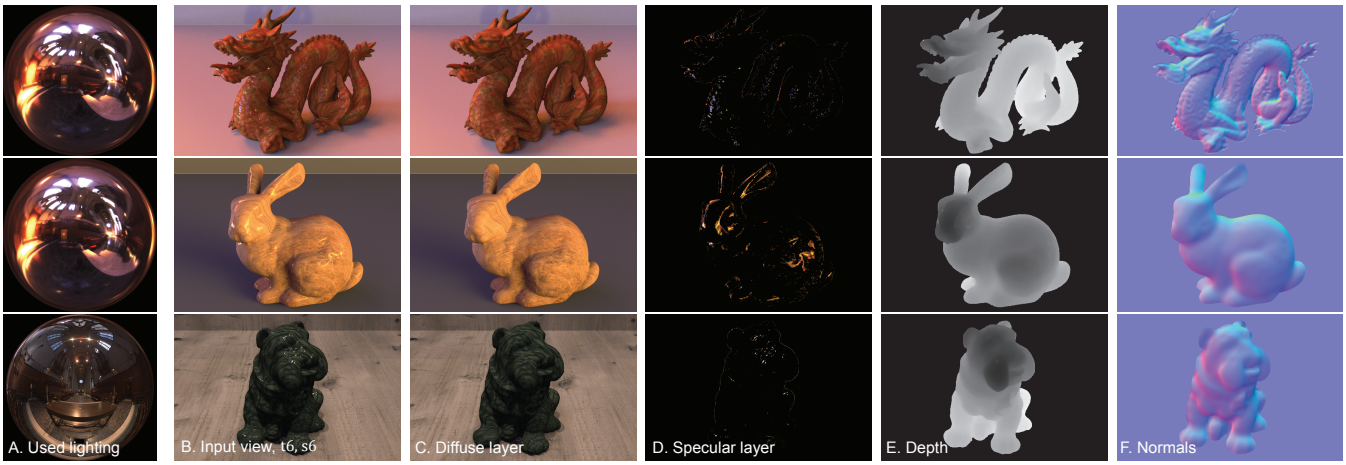

Figure 7: *Steps of our preprocessing pipeline for three sample scenes: bunny, dragon and dog. All three input LFs have 11 by 11 views. For the bunny scene the disparity range is set to [-6:-3.3], for the dragon scene to [-6:-2.5] and for the dog scene to [-5.8:-2.5].*

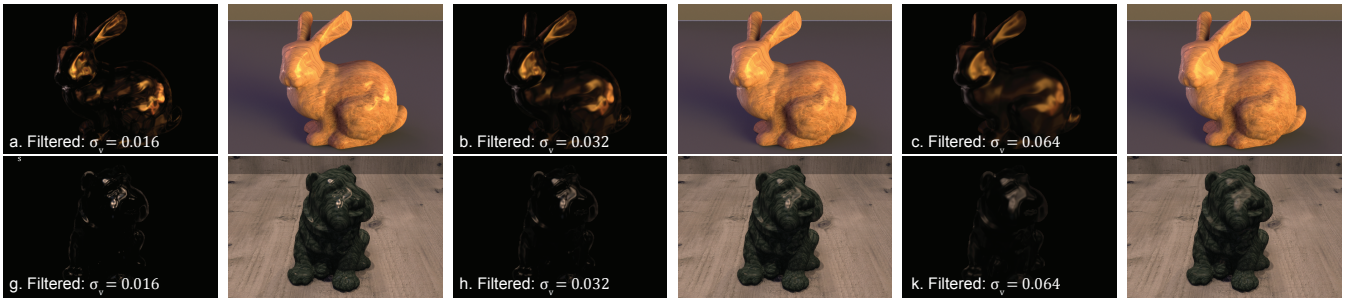

Figure 8: *Several filtering results for the scenes in Fig. 7, with varying degrees of output roughness. For each result we show both the filtered specular layer and the composited result.*

Figure 11: ***a*** *and* ***b.*** *Results of gloss editing using [BBPA15], making the material less and more shiny, respectively (the parameters used are specified in the figure, for their meaning please refer to [BBPA15]).* ***c.*** *Illustration of these operations on an input reflectance lobe (purple), as it is made more shiny (green) or less shiny (blue). Solid lines correspond to the ground truth, and dashed ones to the results obtained by scaling the lobe.*

tion and specular separation algorithms. Both, non-textured objects or objects with very large specular highlights, will result in poor depth estimation. Possible solutions involve integrating depth cameras, hallucinating missing information, or using additional hardware for, e.g., specular separation. So far our algorithm is targeted for roughness increase only and is applicable only to opaque objects. As mentioned in Sec.4 the editing requires preliminary masking the object or the material. We use the ground-truth masks obtained during rendering. Alternatively, the masks can be obtained automatically, using segmentation techniques [MFT*16].

6. Conclusion

We have presented a new technique for modifying glossy appearance, and in particular apparent roughness, of materials in LFs. It is based on the observation that plausible material edits do not require a complete scene reconstruction which is usually expensive and prone to artifacts. Instead, it is sufficient to consistently propagate existing specular information in the spatial and angular domains. For this purpose, we have developed a novel filtering approach that is applied to the specular component of the light field. The filtering parameters have been optimized to provide results closely matching the ground-truth results, and the technique has been evaluated on several scenes as well as compared to ground-truth renderings. Currently, our technique is only capable of increasing surface roughness. The opposite operation is an interesting avenue for future work. Despite our assumptions, the method provides results that closely match the ground-truth results. In the future, perceptual experiments could be used to further evaluate our results and investigate if the technique can be further simplified.

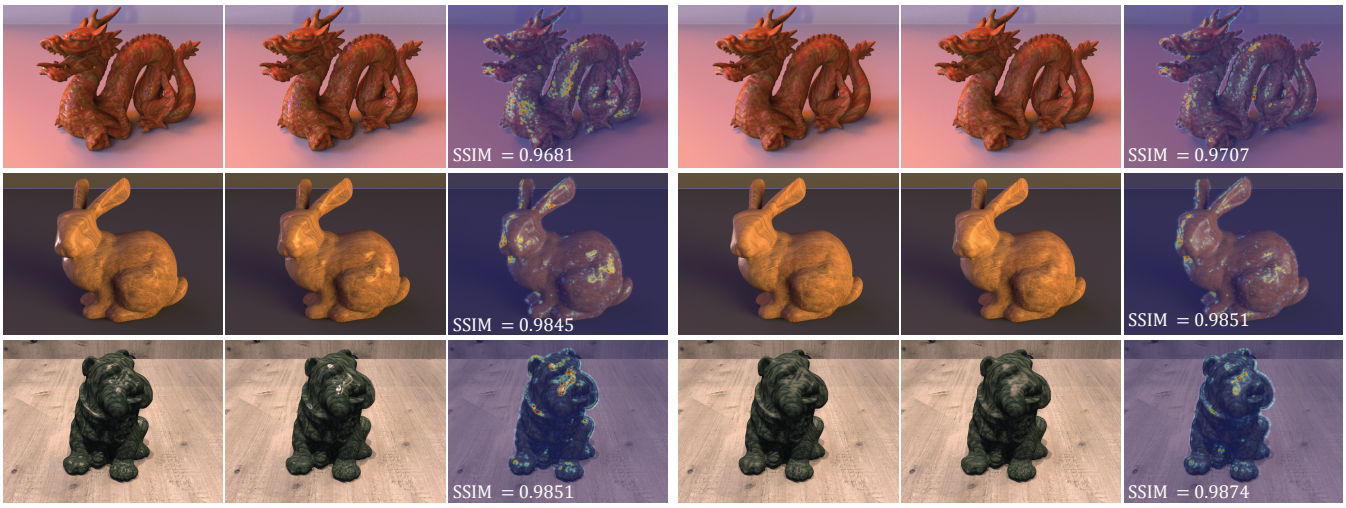

Figure 9: *Comparison of the filtered results (b,e) with ground-truth results for different roughness values (a,d). In this case no optimization of parameters was performed. The ground-truth results were selected manually by visual similarity to the filtered one. Figures (c,f) show the overlay of the filtering results ($\sigma_v = 0.016/\sigma_v = 0.064$) with the dissimilarity maps between the filtered and ground-truth results, computed using SSIM [WBSS04].*

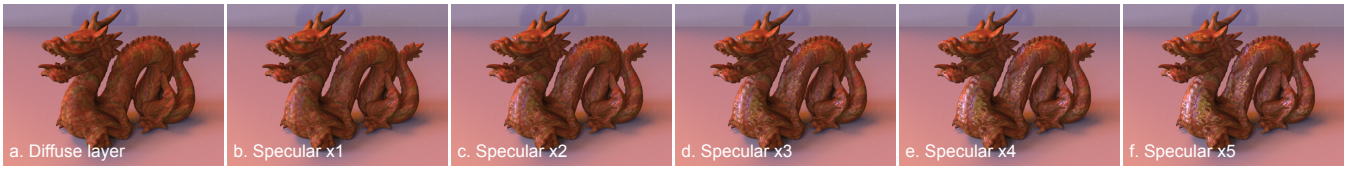

Figure 10: *Edited results obtained merging the diffuse layer with a linearly scaled filtered specular layer ($\sigma_v = 0.032$).*

Acknowledgements

This work was partly supported by the Fraunhofer and the Max Planck cooperation program within the framework of the German pact for research and innovation (PFI).

References

[ABC11] ARTUSI A., BANTERLE F., CHETVERIKOV D.: A survey of specularity removal methods. In *Comp. Graph. Forum* (2011), vol. 30, Wiley Online Library, pp. 2208–2230. 4

[AP08] AN X., PELLACINI F.: Appprop: all-pairs appearance-space edit propagation. *ACM Trans. Graph. (Proc. SIGGRAPH) 27* (2008). 2

[BAOR06] BEN-ARTZI A., OVERBECK R., RAMAMOORTHI R.: Real-time BRDF editing in complex lighting. *ACM Trans. Graph. 25*, 3 (2006), 945–954. 2

[BBPA15] BOYADZHIEV I., BALA K., PARIS S., ADELSON E.: Band-sifting decomposition for image-based material editing. *ACM Trans. Graph. 34*, 5 (2015), 163. 2, 6, 7

[CKS*05] CRIMINISI A., KANG S. B., SWAMINATHAN R., SZELISKI R., ANANDAN P.: Extracting layers and analyzing their specular properties using epipolar-plane-image analysis. *Comput. Vis. Image. Und. 97*, 1 (2005), 51–85. 4

[CLY*14] CHEN C., LIN H., YU Z., KANG S. B., YU J.: Light field stereo matching using bilateral statistics of surface cameras. In *Proc. IEEE CVPR* (2014), pp. 1518–1525. 2, 3

[COSL05] CHEN B., OFEK E., SHUM H.-Y., LEVOY M.: Interactive deformation of light fields. In *Proc. I3D* (2005), pp. 139–146. 2

[CP06] COLBERT M., PATTANAIK S.: BRDF-shop: Creating physically correct bidirectional reflectance distribution functions. *IEEE Comput. Graph. Appl.* (2006), 30–36. 2

[DAW01] DROR R. O., ADELSON E. H., WILLSKY A. S.: Recognition of surface reflectance properties from a single image under unknown real-world illumination. In *Proc. of IEEE Workshop on Identifying Objects Across Variations in Lighting* (2001), pp. 1–8. 2

[DFY*11] DOERSCHNER K., FLEMING R. W., YILMAZ O., SCHRATER P. R., HARTUNG B., KERSTEN D.: Visual motion and the perception of surface material. *Current Biology 21*, 23 (2011), 2010–2016. 2

[Fle12] FLEMING R.: Human perception: Visual heuristics in the perception of glossiness. *Current Biology 22*, 20 (2012), R865–R866. 2

[GSLM*08] GUTIERREZ D., SERON F. J., LOPEZ-MORENO J., SANCHEZ M. P., FANDOS J., REINHARD E.: Depicting procedural caustics in single images. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 27*, 5 (2008), 120:1–9. 2

[HFB*09] HABER T., FUCHS C., BEKAERT P., SEIDEL H.-P., GOESELE M., LENSCH H.: Relighting objects from image collections. In *Proc. CVPR* (2009), IEEE, pp. 627–634. 2

[HP14] HEBER S., POCK T.: Shape from light field meets robust pca. In *Proc. ECCV*. Springer, 2014, pp. 751–767. 2

[HST13] HE K., SUN J., TANG X.: Guided image filtering. *IEEE TPAMI 35*, 6 (2013), 1397–1409. 4

[JMB*14] JARABO A., MASIA B., BOUSSEAU A., PELLACINI F., GUTIERREZ D.: How do people edit light fields? *ACM Trans. Graph. (Proc. SIGGRAPH) 33*, 4 (2014). 2

[JMG11] JARABO A., MASIA B., GUTIERREZ D.: Efficient propagation of light field edits. In *Proc. SIACG* (2011), pp. 75–80. 2

[KRFB06] KHAN E. A., REINHARD E., FLEMING R. W., BÜLTHOFF H. H.: Image-based material editing. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 654–663. 2

[KZP*13] KIM C., ZIMMER H., PRITCH Y., SORKINE-HORNUNG A., GROSS M. H.: Scene reconstruction from high spatio-angular resolution light fields. *ACM Trans. Graph. (Proc. SIGGRAPH) 32*, 4 (2013). 2

[LBAD*06] LAWRENCE J., BEN-ARTZI A., DECORO C., MATUSIK W., PFISTER H., RAMAMOORTHI R., RUSINKIEWICZ S.: Inverse shade trees for non-parametric material representation and editing. *ACM Trans. Graph. (Proc. SIGGRAPH) 25*, 3 (2006), 735–745. 2

[LCBKY15] LIN H., CHEN C., BING KANG S., YU J.: Depth recovery from light field using focal stack symmetry. In *IEEE ICCV* (2015), pp. 3451–3459. 2, 3

[LH96] LEVOY M., HANRAHAN P.: Light field rendering. In *Proc. SIGGRAPH* (1996), ACM, pp. 31–42. 1

[LLK*02] LIN S., LI Y., KANG S. B., TONG X., SHUM H.-Y.: Diffuse-specular separation and depth recovery from image sequences. In *Proc. ECCV*. Springer, 2002, pp. 210–224. 3, 4

[MFT*16] MIHARA H., FUNATOMI T., TANAKA K., KUBO H., NAGAHARA H., MUKAIGAWA Y.: 4D light field segmentation with spatial and angular consistencies. 7

[NDM06] NGAN A., DURAND F., MATUSIK W.: Image-driven navigation of analytical BRDF models. In *Proc. EGSR* (2006). 2

[NLB*05] NG R., LEVOY M., BRÉDIF M., DUVAL G., HOROWITZ M., HANRAHAN P.: Light field photography with a hand-held plenoptic camera. *Comp. Science TR 2*, 11 (2005), 1–11. 1

[NSRS13] NGUYEN C. H., SCHERZER D., RITSCHEL T., SEIDEL H.-P.: Material editing in complex scenes by surface light field manipulation and reflectance optimization. vol. 32. 2

[OCDD01] OH B. M., CHEN M., DORSEY J., DURAND F.: Image-based modeling and photo editing. In *Proc. SIGGRAPH* (2001), pp. 433–442. 2

[OCS05] OSTROVSKY Y., CAVANAGH P., SINHA P.: Perceiving illumination inconsistencies in scenes. *Perception 34*, 11 (2005). 2

[PL07] PELLACINI F., LAWRENCE J.: Appwand: Editing measured materials using appearance-driven optimization. *ACM Trans. Graph. (Proc. SIGGRAPH) 26*, 3 (2007). 2

[ROTS09] RITSCHEL T., OKABE M., THORMÄHLEN T., SEIDEL H.-P.: Interactive reflection editing. *ACM Trans. Graph. 28*, 5 (2009). 2

[SK02] SEITZ S. M., KUTULAKOS K. N.: Plenoptic image editing. *INT J COMPUT VISION 48*, 2 (2002), 115–129. 2

[TGY*09] TALTON J. O., GIBSON D., YANG L., HANRAHAN P., KOLTUN V.: Exploratory modeling with collaborative design spaces. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 28*, 5 (2009), 167:1–10. 2

[TSM*15] TAO M. W., SRINIVASAN P. P., MALIK J., RUSINKIEWICZ S., RAMAMOORTHI R.: Depth from shading, defocus, and correspondence using light-field angular coherence. In *CVPR* (2015), pp. 1940–1948. 2, 3

[TSW*15] TAO M., SU J.-C., WANG T.-C., MALIK J., RAMAMOORTHI R.: Depth estimation and specular removal for glossy surfaces using point and line consistency with light-field cameras. *IEEE TPAMI*, 1 (2015). 4

[VBFG12] VERGNE R., BARLA P., FLEMING R. W., GRANIER X.: Surface flows for image-based shading design. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012), 94:1–94:9. 2

[VLD*13] VENKATARAMAN K., LELESCU D., DUPARRÉ J., MCMAHON A., MOLINA G., CHATTERJEE P., MULLIS R., NAYAR S.: Picam: An ultra-thin high performance monolithic camera array. *ACM Trans. Graph. (Proc. SIGGRAPH Asia) 32*, 6 (2013), 166:1–166:13. 1

[VTMC11] VETRO A., TOURAPIS A., MULLER K., CHEN T.: 3D-TV content storage and transmission. *IEEE Trans. on Broadcasting 57*, 2 (2011), 384–394. 1

[WBSS04] WANG Z., BOVIK A. C., SHEIKH H. R., SIMONCELLI E. P.: Image quality assessment: from error visibility to structural similarity. *IEEE TIP 13*, 4 (2004), 600–612. 6, 8

[WFEM10] WENDT G., FAUL F., EKROLL V., MAUSFELD R.: Disparity, motion, and color information improve gloss constancy performance. *J. Vis. 10*, 9 (2010), 7. 2

[WLHR12] WETZSTEIN G., LANMAN D., HIRSCH M., RASKAR R.: Tensor displays: Compressive light field synthesis using multilayer displays with directional backlighting. *ACM Trans. Graph. (Proc. SIGGRAPH) 31*, 4 (2012), 1–11. 1

[WLL*05] WANG L., LIN S., LEE S., GUO B., SHUM H.-Y.: Light field morphing using 2D features. *IEEE TVCG 11*, 1 (2005), 25–34. 2

[WSG13] WANNER S., STRAEHLE C., GOLDLUECKE B.: Globally consistent multi-label assignment on the ray space of 4D light fields. In *IEEE CVPR* (2013), pp. 1011–1018. 4

[WSR16] WANG T.-C., SRIKANTH M., RAMAMOORTHI R.: Depth from semi-calibrated stereo and defocus. In *Proc. of the IEEE CVPR* (2016), pp. 3717–3726. 2, 3

[YMG04] YU J., MCMILLAN L., GORTLER S.: Surface camera (SCam) light field rendering. *IJIG 4*, 04 (2004), 605–625. 3

[YTBK11] YEUNG S.-K., TANG C.-K., BROWN M. S., KANG S. B.: Matting and compositing of transparent and refractive objects. *ACM Trans. Graph. 30*, 1 (2011), 2:1–2:13. 2

[ZSL*16] ZHANG S., SHENG H., LI C., ZHANG J., XIONG Z.: Robust depth estimation for light field via spinning parallelogram operator. *Comput. Vis. Image. Und. 145* (2016), 148 – 159. 2

[ZWGS02] ZHANG Z., WANG L., GUO B., SHUM H.-Y.: Feature-based light field morphing. *ACM Trans. Graph. (Proc. SIGGRAPH) 21* (2002), 457–464. 2

[ZWS*16] ZHANG F. L., WANG J., SHECHTMAN E., ZHOU Z. Y., SHI J. X., HU S. M.: Plenopatch: Patch-based plenoptic image manipulation. *IEEE TVCG PP*, 99 (2016), 1–1. 2