Light Source Interpolation for Sparsely Sampled Reflectance Fields

Billy Chen, Hendrik P. A. Lensch

Stanford University
Computer Graphics Lab, 353 Serra Mall, 94305 Stanford, CA, USA
Email: {billyc,lensch}@graphics.stanford.edu

Abstract

In this paper we address the problem of relighting with sparsely sampled reflectance fields. We present a technique that approximates the correct result of relighting from intermediate light source positions. The acquisition of reflectance fields is a time consuming process, and typically the sampling resolution in the light source positions is rather limited. As a consequence, smoothly moving highlights and shadows due to relighting with a moving light source are hard to generate. Using light source interpolation, densely sampled reflectance fields can be simulated, enabling relighting with area light sources and smooth animation of highlights and shadows. Using light source interpolation we can relight with arbitrarily sampled 4D incident light fields from complex or near-by light sources.

1 Introduction

Light fields and lumigraphs [9, 7] allow for synthesizing arbitrary views from previously recorded scenes or objects. When capturing a *relightable* representation, which is typically a slice of the 8D reflectance field, one can even relight the object. Relightable representations are essential for compositing the object into virtual environments since the reproduction of the appearance under novel lighting conditions is required in order to generate a correct image which accounts for all relevant shading and global illumination effects.

Current devices for acquiring reflectance fields [4, 10, 14, 8, 11, 6, 16], with some exceptions [15, 8], provide a rather limited sampling of the light source positions, due to limited acquisition time or the physical spacing of the sources.

The coarse sampling at discrete locations results in artifacts in the relit images. Some of these artifacts include banded shadows (instead of soft shadows) when illuminated by an area light source, or abrupt jumps in light dependent effects (e.g. shadows and sharp highlights) when moving the light source along a smooth path. Given a coarsely sampled reflectance field these problems can only be overcome by computing relit images that resemble the appearance of the scene as if illuminated from a light source position in between the originally captured locations. We call this process *light source interpolation*.

Clearly, simple blending between images relit from the recorded light source positions would not be sufficient. Shadows and highlights will crossfade from one location to the other instead of moving smoothly. Our approach for light source interpolation flows illumination effects that undergo large movement (e.g. shadows) and blends those regions where differences are rather small.

In this paper we focus on light source interpolation for reflectance fields which allow for relighting with 4D incident light fields [11, 6, 16]. Using these reflectance fields, a spatially varying illumination pattern can be projected into the scene from any of the recorded light sources. Using interpolation we are able to project from intermediate projector positions. In particular, we present a pipeline for performing light source interpolation for sparsely sampled reflectance fields for relighting with 4D incident light fields. The pipeline performs transformations on both the incident and exitant light fields. These transformations are tightly coupled; computation in one domain guides the transformation in the other. We will first review related work before presenting our processing pipeline for light source interpolation.

2 Previous Work

A light stage [4, 10] samples the reflectance field of objects for a single view point, as illuminated by a set of point light sources with fixed spacing. In order to record highly specular or transparent objects the spacing between the light source has to be reduced, i.e. the resolution of the reflectance field has to be increased drastically [14, 8]. Relighting based on these data sets is limited to distant light sources, e.g. environment maps.

Reflectance fields for spatially varying illumination or 4D incident light fields are captured using a projector instead of a point light source [11, 6, 16]. Using a projector, individual light rays can be addressed. In all three approaches the resolution of the projector positions is rather limited and the captured data sets would profit from correct light source interpolation. In this paper, we use reflectance fields acquired using the technique by Sen et al. [16], which provides sufficient resolution to project illumination patterns of up to 1280×1024 pixels into the scene.

Closely related to the problem of light source interpolation is view interpolation within reflectance fields, i.e. to generate images from camera positions not recorded in the original reflectance field. Matusik et al. [15] presented a view interpolation method especially designed for refracting objects. A similar problem has to be solved when deforming the geometry on which a surface reflectance field is defined [20].

When the reflectance field is defined on geometry, one can perform scattered data interpolation to compute intermediate lighting positions, as shown in [21]. Instead of directly interpolating the reflectance field, our technique performs transformations on the incident and exitant light fields.

To compute decent light source interpolations we make use of *intrinsic* images [2]. Intrinsic images can be used to analyze fixed-view sequences of a scene under arbitrary varying illumination. The image sequence is decomposed into a single reflectance image and a set of time-varying illumination images. Weiss [19] applies a maximum-likelihood framework to perform the decomposition. Instead of interpolating the light source illumination directly from relit images, we interpolate on the illumination images since this is what changes when moving the light source. The use of intrinsic images is however limited by the fact that in the computed illumination images, geometric features and the textures of surfaces with varying specular properties are often still visible. Illumination images often contain high frequencies from features which have nothing to do with the real incident illumination. Allowing reflectance images to vary over time and filtering and thresholding in the gradient domain reduces some of these artifacts, and much smoother illumination images are generated [13]. In this paper we also separate images into reflectance and illumination images to ascertain changes in the effective illumination.

Intrinsic images have been used before to perform light source interpolation in reflectance fields of diffuse scenes by Matsushita et al. [12], mainly to determine shadow regions. In their technique an explicit 3D model is reconstructed from multiple view points. Using the 3D model, the movement of shadows for intermediate light source positions can be simulated. By transferring the shadow motion information to the shadows found in the image data the appearance of shadows in intermediate frames is computed quite realistically. Since this 3D reconstruction is difficult and error prone for uncooperative surfaces we resort to optical flow for moving shadows.

3 Overview

Our processing pipeline takes as input, reflectance fields captured from a single camera position. The spatially varying illumination (also known as a 4D incident light field) is parameterized with respect to a set of projectors. Let us first introduce the representation of the reflectance field for spatially varying illumination, which corresponds to the one used in [16].

Given a reflectance field captured using N different projector positions, the part of the reflectance field that describes the transport between the camera and a single projector j can be written efficiently as the transport matrix T_j for $j \in 0, \ldots, N-1$. Let C_j denote the camera image that is observed when projecting the illumination pattern P_j (a 2D image) from projector j into the scene. C_j can be computed given T_j and P_j:

$$C_j = T_j P_j \tag{1}$$

The transport matrix stores for every projector pixel p_{xy} its effect on all camera pixels c_{xy}. As-

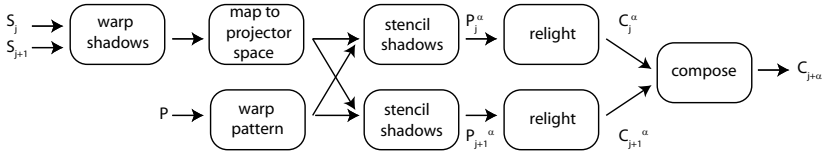

Figure 1: Processing pipeline for light source interpolation between projector locations j and $j+1$. The original shadow masks of the acquired projectors S_j and S_{j+1} are warped to the intermediate frame $j+\alpha$ and mapped from camera to projector space. At the same time, the input pattern P is warped into P_j^α and P_{j+1}^α such that the warped pixels of the original pattern will be projected to the same surface location although projected from j and $j+1$ instead of $j+\alpha$. The transformed shadow is stenciled out in the warped projector patterns. By relighting with each input pattern individually, two camera images C_j^α and C_{j+1}^α are computed, which are then composited into the final result $C_{j+\alpha}$.

sume the projector resolution is $p \times q$ and camera resolution is $m \times n$, then the dimensions of the transport matrix are $mn \times pq$. The number of non-zero elements in the matrices used in this paper is of the order of billions, and a hierarchical representation as suggested in [16] is mandatory. For the remainder of this paper it is however sufficient to think of T as a huge but simple matrix.

Let us concentrate on light source interpolation in the one-dimensional case where we would like to compute an image of the scene as if illuminated from a projector that is located between two sampled projector positions j and $j+1$. Given the interpolation parameter $\alpha \in [0,1]$, the virtual projector should have moved the fraction α of the distance from j into the direction of $j+1$. The final image we aim to compute is $C_{j+\alpha}$.

In theory, one could compute $C_{j+\alpha}$ by first synthesizing $T_{j+\alpha}$ from T_j and T_{j+1} followed by evaluating Equation 1: $C_{j+\alpha} = T_{j+\alpha}P$. Since the effort of assembling an entirely new transport matrix is enormous and the way of computing $T_{j+\alpha}$ is unclear, in this paper we solve the problem without directly computing the intermediate transport matrix. Instead we apply warping techniques both in the projector space and the camera space. That is, we warp the incident illumination and the relit images. Our processing pipeline is depicted in Figure 1:

1. **Shadow Warping** (camera space). In a preprocessing step we compute intrinsic images of the scene using floodlit illumination in each projector. By decomposing the resulting illumination images into a moving and an almost static layer, shadow regions in the original images are detected in the moving layer. This gives us a shadow mask S_j for each projector location j. Computing optical flow on the shadow masks yields the shape of the shadow for the intermediate frame $S_{j+\alpha}$ (see Section 4). This is depicted by the "warp shadows" box in Figure 1. The decomposition into two layers is necessary since the flow field between the illumination images can only be reliably estimated for regions with strong differences (i.e. in the moving layer).
2. **Pattern Warping** (projector space). In a preprocessing step we also compute the flow vectors that describe where a pixel in projector j moves to in projector $j+1$ in order to illuminate the same scene point. If we had placed cameras in the positions of projectors j and $j+1$ this flow field corresponds exactly to the disparity map. Given the illumination pattern P for projector $j+\alpha$ we apply the aforementioned flow field, weighted by α, to generate two distinct patterns P_j^α and P_{j+1}^α for the nearest original projectors. Intuitively, after warping the patterns the incident illumination mimics the incident illumination from $j+\alpha$, although projected from j and $j+1$ (see Section 5). This is depicted by the "warp pattern" box in Figure 1.
3. **Shadow Stenciling** (projector space). We simulate the interpolated shadow $S_{j+\alpha}$ in the final image by masking out the appropriate pixels in the illumination patterns P_j^α and P_{j+1}^α. The affected pixels are determined by transforming the interpolated shadow mask $S_{j+\alpha}$ into the projector space for both projectors (Section 5.2). This is depicted by the "stencil shadows" box in Figure 1.

4. **Relighting.** The relighting is performed using both original transport matrices: $C_j^\alpha = T_j P_j^\alpha$ and $C_{j+1}^\alpha = T_{j+1} P_{j+1}^\alpha$. The resulting images C_j^α and C_{j+1}^α are very similar except for lighting variations due to the different incident light source positions. This process is depicted by the "relight" box in Figure 1.
5. **Compositing** (camera space). The final image is obtained by compositing C_j^α and C_{j+1}^α based on the warped shadow mask: $C_{j+\alpha} = compose(\alpha, S_{j+\alpha}; C_j^\alpha, C_{j+1}^\alpha)$. For regions where the illumination image shows drastic differences, those regions in the final image are obtained by selecting from either the first or the second image. For regions where the change is only moderate, we blend between the two images based on α. The compositing is explained in Section 6, and depicted as the "compose" box in Figure 1.

4 Interpolation Based on Intrinsic Images

Intrinsic images can be used to predict how to alter images C_j taken under two different lighting conditions to generate an intermediate illumination. The input images are separated into a common reflectance image $R(x, y)$ and an illumination image $L_j(x, y)$ corresponding to the actual lighting using $C_j(x, y) = R(x, y) \cdot L_j(x, y)$. The reflectance image is typically computed by applying a median filter directly on the images [2] or by applying it in the gradient domain [19, 13] over multiple input frames. We extend the first approach to better detect differences in the illumination images L_j.

To get from L_j and L_{j+1} to $L_{j+\alpha}$ one could simply blend between them, resulting in cross-faded shadows, or one could try to interpolate flow to really move darkened or brightened regions.

Unfortunately, applying a standard optical flow algorithm [3] to illumination images typically does not result in a meaningful flow field. The illumination images may contain both residual surface texture or the geometry, which would contaminate the computed flow (Figure 2e). We have tried to use the technique proposed by Matsushita et al. [13] in order to remove some of the texturing effects with limited success.

The solution we propose is to separate the illumination image into two layers: one layer containing pixels to be flowed, and the other to be blended. When thresholding on the absolute difference between L_j and L_{j+1}, pixels below the threshold can be blended with no noticeable artifacts. In this case blending would work just fine. This is true for most diffuse reflection, glossy reflectance with broad highlights, or sharp specular highlights which move by only one or two pixels.

After removing all pixels below the threshold it is now easier to compute the optical flow for the remaining regions which differ to a greater extent. In Figure 2, we demonstrate the increased quality of computing the flow on the thresholded illumination over computing it directly from illumination images.

Of course all standard problems of estimating optical flow such as flow in the presence of occluders [17] or the aperture problem still remain, and in such cases our optical flow technique would also fail. Precise movement of shadows can be obtained by incorporating an explicit 3D model of the scene [12]. In the future we plan to derive the 3D model from the transport matrix and would like to further investigate how the movement of shadows can be predicted directly from the transport matrix.

In our pipeline, if the shadow moves behind an occluder then no correspondence is found and thus the interpolated shadow will not move. In this case we resort to blending between frames. By incorporating forward and backward flow and blending between the results of both we handle the case of a zero flow field automatically and consistently.

We can now warp the shadow regions in the illumination image to obtain an intermediate illumination image. As we will explain in Section 5.2 we process the morphed shadow regions even further to deal with the effect of interreflections.

5 Interpolation in Projector Space

This section describes the steps for deforming the input pattern to mimic the incident illumination of the intermediate projector. To do this we apply warping and shadow stenciling.

5.1 Warping in Projector Space

Given an input pattern for the intermediate projector $j + \alpha$ we would like to compute input patterns for the original projectors j and $j + 1$ that best reproduce the appearance as if actually lit from location

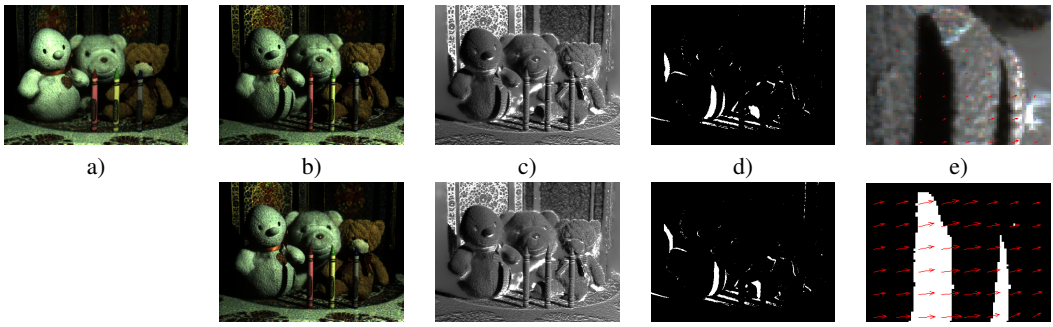

Figure 2: Applying optical flow to intrinsic images. Images captured under two different illumination conditions (b) and (f) are decomposed into a common reflectance image (a) and illumination images (c) and (g). Note how the shadow has moved slightly between the two images (b) and (f). A zoomed-in version of the optical flow determined between the original illumination images is shown in (e), overlaid over the illumination image. The determined flow vectors have almost zero length indicating that the algorithm would not move the shadow, which is incorrect. After thresholding the illumination images, producing (d) and (h), a more accurate flow field (i) can be obtained, which would flow the shadows in the right direction with the correct velocity.

$j + \alpha$. We want to warp the patterns so that they appear aligned on the surface of the scene.

Let $\vec{p}_j$ denote the coordinates of a pixel in projector j. The pixel at $\vec{p}_j$ projected from j will directly illuminate the surface visible at camera pixel location $\vec{c}_j$. Recall, a column of the transport matrix T_j corresponds to the image resulting from illuminating the scene by turning on a single projector pixel. Selecting the column corresponding to $\vec{p}_j$, we can locate $\vec{c}_j$ by searching in this column for the brightest response. Thus, we can establish the mapping $\mathcal{M}_j(\vec{p}_j) = \vec{c_j}$ from projector to camera space, as illustrated in Figure 3.

Given $\mathcal{M}_j$ we know the inverse mapping, $\mathcal{M}_j^{-1}$, from camera space to projector space as well. The mapping will initially be sparse due to sampling, shadows, or very dark surfaces for which the entries in the column are too dim to make out a significant peak. We obtain a dense mapping by applying anisotropic diffusion filtering to the sparse $\mathcal{M}_j$ and $\mathcal{M}_j^{-1}$ mappings. For pixels in shadow regions the information obtained from anisotropic diffusion might be wrong, but those pixels have no effect on the final relit image. The mappings for two neighboring projectors are depicted in Figure 6.

Given the mapping $\mathcal{M}_j$ and a precise calibration of the camera and the projector we could actually compute range images, explicitly reconstructing the 3D shape of the scene. Note how Figure 6, which

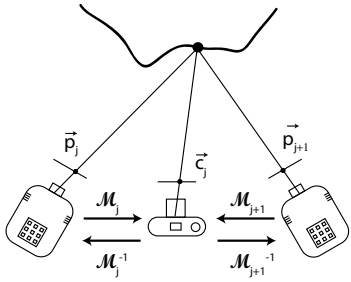

Figure 3: Illustration of the mapping $\mathcal{M}$ and its inverse $\mathcal{M}^{-1}$, mapping from the projector to the camera space and from the camera to the projector space, respectively. The flow between two projector patterns is given by $ProjFlow_{(j \to j+1)}(\vec{p}_j) = \mathcal{M}_{j+1}^{-1}(\mathcal{M}_j(\vec{p}_j))$

is a color-coded visualization of $\mathcal{M}_j^{-1}$, resembles a range image. However, for our analysis, the mapping $\mathcal{M}_j$ is sufficient.

In order to obtain the flow from one projector pattern to the next we simply determine the mapping $ProjFlow_{(j \to j+1)}(\vec{p}_j) = \mathcal{M}_{j+1}^{-1}(\mathcal{M}_j(\vec{p}_j))$ (see Figure 3) by projecting the point $\vec{p}_j$ from projector j into the the camera space using $\mathcal{M}_j$ and then mapping it back to the projector space of $j+1$ using the inverse mapping $\mathcal{M}_{j+1}^{-1}$.

Given this flow field we create two warped ver-

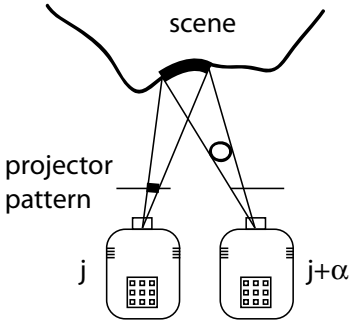

Figure 4: The projector pattern P_j is modified to simulate the effect of a cast shadow from projector $j + \alpha$. This is accomplished by stenciling out the pixels that would otherwise illuminate the shadow region.

sions of the original projector pattern P, one for each projector, yielding P_j^α and P_{j+1}^α. The result of applying the flow field to the input pattern is demonstrated in Figure 5 (bottom row (n) and (p)). Note how the warped checkerboard pattern projected from projector j precisely matches the result of projecting the original pattern from $j + 1$.

5.2 Shadow Stenciling

In the next step, pixels corresponding to shadow regions for the interpolated light source position are stenciled out in the warped projector patterns. This effectively projects a synthetic shadow which would not have been cast by the original projectors (see Figure 4). If we were to directly stencil the shadows in the camera image, we would remove direct illumination effects, but not the multiple interreflections caused by illuminating the shadowed points. By stenciling out the incident illumination for these pixels we eliminate these interreflections. In addition, our approach automatically accounts for multiple interreflections from *non*-shadowed points which might actually brighten the shadow region.

The shadow stenciling is implemented as follows: First we compute the warped shadow image $S_{j+\alpha}$. Using $\mathcal{M}_j^{-1}$ and $\mathcal{M}_{j+1}^{-1}$, the shadow image is transformed into the projector space. The transformed shadow is then multiplied with the warped illumination patterns of the previous section to compute the final patterns P_j^α and P_{j+1}^α. When applying the stenciled patterns, a synthetic shadow is cast into the scene beside the shadow that is present in the original images. Figure 5 (third column) demonstrates the process of stenciling the warped patterns.

6 Compositing

After warping and stenciling the illumination patterns for projectors j and $j + 1$, we multiply them against their transport matrices to obtain two images $C_j^\alpha = T_j P_j^\alpha$ and $C_{j+1}^\alpha = T_{j+1} P_{j+1}^\alpha$ (see Figure 5(g) for an example). These images are already pretty close to the solution (Figure 5(h)) except that they still contain the original shadow and that moving highlights have not yet been warped. In this section we concentrate on compositing the shadows and in Section 8 we adddress how to move the highlights.

The final image is obtained by a per-pixel blend with a spatially varying blending factor β:

$$\begin{aligned} C_\alpha(x,y) &= (1-\beta(x,y))C_j^\alpha(x,y) \\ &\quad +\beta(x,y)C_{j+1}^\alpha(x,y) \end{aligned} \tag{2}$$

Since both images contain the synthesized shadow $S_{j+\alpha}$ we do not need to consider it during compositing. Initially we determine $\beta(x, y)$ based on the shadow masks S_j and S_{j+1} in the following way:

$$\beta = \begin{cases} 0 & : \quad S_{j+1} = 1 \wedge S_j \neq S_{j+1} \\ \alpha & : \quad S_j = S_{j+1} \\ 1 & : \quad S_j = 1 \wedge S_j \neq S_{j+1} \end{cases} \tag{3}$$

For brevity we have dropped the (x, y)-parameters. If $S_j = 1$, it means that the pixel will be in shadow in image C_j^α. In this case we select the pixel from C_{j+1}^α since in that image the pixel might be illuminated. In the same way, if $S_{j+1} = 1$ we select from C_j^α. If the pixel is illuminated from both projectors or is in shadow from both projectors, we simply blend between the appropriate pixels in C_j^α and C_{j+1}^α weighted by α.

At the border between regions selected from a single image and blending regions there should be a smooth transition. This can be accomplished by blurring the β-image and then applying compositing Equation 2.

In Figure 8 we compare the final result of warping, stenciling and compositing with what would

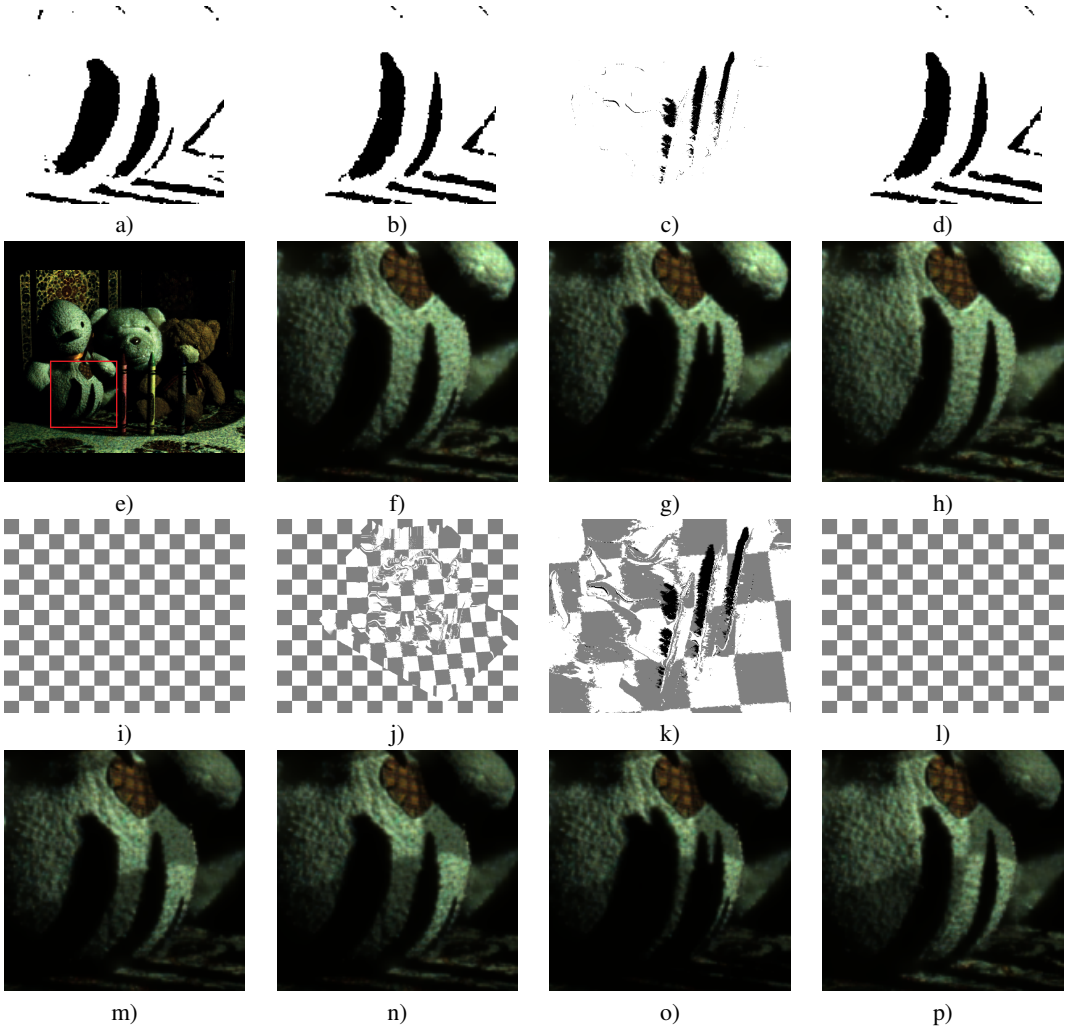

Figure 5: Shadow warping, pattern warping and shadow stenciling. First row: The shadow mask (a) of the left projector is warped, shown in (b), to match the shadow mask of the right projector shown in (d). (b) and (d) therefore appear the same. The warped shadow mask is then transformed into the projector space as shown in (c). Second row: The scene is lit with floodlit illumination and the camera image is shown in (e). The zoomed-in versions (f) and (h) show the original shadows cast from the left and right projector, respectively. When projecting the stenciled illumination pattern (c) from the left projector a synthetic shadow is projected. The resulting image (g) now contains both the original and the synthesized shadow. Third row: A checkerboard input pattern (i) and (l) is projected into the scene from the left and right projectors, resulting in camera images (m) and (p). Warping the input pattern (j) and projecting it from the left projector results in (n), which simulates the effect of projecting the original pattern from the right projector, as shown in (p). Note that the patterns are perfectly aligned in the camera space. Stenciling out the warped and transformed shadow mask (c) in the warped input pattern (j) produces the zoomed-in image shown at (k). Projecting this pattern results in the image (o) which again contains both the original and the synthetic shadows.

be obtained by blending the illuminated images directly.

7 Relighting Results

Being able to smoothly interpolate between two projector locations, we can simulate a point light moving along a path, which we demonstrate in the accompanying video sequence [1]. Since the reflectance field allows for spatially varying illumination, even relighting from a moving projector is simulated. Because of the warping in projector space the observed movement of the pattern is smooth.

When integrating over a set of interpolated images we can relight the scene virtually by an area light source (see Figure 9). Incorporating a moderate number of interpolated light source positions, we can generate the appearance of soft shadows as one expects from area light sources. Summing over a number of original projector locations would instead produce banded shadows (shadow aliasing).

The presented framework allows for relighting with arbitrary incident light fields of high resolution in the spatial and angular domain. These light fields can be captured from real world light sources [5] or environments [18]. We present the result of relighting the scene with a synthetic light field in Figure 7. The incident light field simulates a checkerboard pattern projected from a projector with a very large curved aperture leading to a small region of focus in the middle of the scene where the checkerboard pattern can still be observed while it is blurred in the periphery.

8 Conclusion

The presented pipeline is, to our knowledge, the first attempt of light source interpolation for sparse reflectance fields for illumination with 4D incident light fields. We warp the incident light fields to match the sampled projector locations, we propose a two layer approach to determining flow in illumination images, warp shadows and stencil them out in the projector patterns. Our technique allows for creating smooth animations for moving light sources and correct shadows for area light sources. Although not demonstrated in the paper, specular highlights can also be flowed using this framework by setting an appropriate threshold when splitting the illumination image into a static and moving layer.

The presented methods currently build on a number of simplifying assumptions: First, the peak in each column of the transport matrix actually corresponds to the first bounce of light from the projector. Thus, it is not clear how the algorithm would behave in the presence of specular interreflections, e.g. mirrors or caustics. Second, we assume that the movement of the shadows can be simulated by using linear flow vectors. This requires a moderately dense sampling of the original reflectance field. In the future we plan to come up with a more general solution that works for a larger class of realistic scenes. It might be possible to extract much more information from the transport matrix that can be used to guide the estimation of the flow between all elements in two neighboring matrices, leading to a 4D flow estimation algorithm.

Acknowledgments

This work has been partially funded an NSF contract IIS-0219856-001, a DARPA contract NBCH-1030009, and by the Max Planck Center for Visual Computing and Communication (BMBF-FKZ 01IMC01).

References

[1] http://graphics.stanford.edu/papers/light_source_interpolation.

[2] H.G. Barrow and J.M. Tenenbaum. Recovering intrinsic scene characteristics from images. In A. Hanson and E. Risernan, editors, *Computer Vision Systems*. Academic Press, 1978.

[3] M. J. Black and P. Anandan. The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields. *Computer Vision and Image Understanding, CVIU*, 63(1):75–104, January 1996.

[4] P. Debevec, T. Hawkins, C. Tchou, H.-P. Duiker, W. Sarokin, and M. Sagar. Acquiring the Reflectance Field of a Human Face. In *Proc. SIGGRAPH*, pages 145–156, July 2000.

[5] M. Goesele, X. Granier, W. Heidrich, and H.-P. Seidel. Accurate light source acquisition and rendering. *ACM Trans. Graphics (Proc. SIGGRAPH 2003)*, 22(3):621–630, 2003.

[6] M. Goesele, H. P. A. Lensch, J.Lang, C. Fuchs, and H.-P. Seidel. Disco: Acquisition of translucent objects. *ACM Trans. Graphics (Proc. SIGGRAPH 2004)*, 23(3):835–844, 2004.

[7] S. Gortler, R. Grzeszczuk, R. Szelinski, and M. Cohen. The Lumigraph. In *Proc. SIGGRAPH*, pages 43–54, August 1996.

[8] T. Hawkins, P. Einarsson, and P. Debevec. A dual light stage. In *Eurographics Symposium on Rendering*, 2005.

[9] M. Levoy and P. Hanrahan. Light Field Rendering. In *Proc. SIGGRAPH*, pages 31–42, August 1996.

[10] V. Masselus, P. Dutré, and F. Anrys. The Free-form Light Stage. In S. Debevec, P. Gibson, editor, *Proc. EuroGraphics Workshop on Rendering*, pages 257–265, Pisa, Italy, June 26–28 2002.

[11] V. Masselus, P. Peers, P. Dutré;, and Y. D. Willems. Relighting with 4D incident light fields. *ACM Trans. Graph. (Proc. SIGGRAPH 2003)*, 22(3):613–620, 2003.

[12] Y. Matsushita, S.-B. Kang, S. Lin, H.-Y. Shum, and X. Tong. Lighting and shadow interpolation using intrinsic lumigraphs. *Journal of Image and Graphics (IJIG)*, 4(4):585–604, Oct. 2004.

[13] Y. Matsushita, K. Nishino, K. Ikeuchi, and M. Sakauchi. Illumination normalization with time-dependent intrinsic images for video surveillance. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3–10, 2003.

[14] W. Matusik, H. Pfister, A. Ngan, P. Beardsley, and L. McMillan. Image-Based 3D Photography Using Opacity Hulls. In *Proceedings SIGGRAPH*, pages 427–437, July 2002.

[15] W. Matusik, H. Pfister, R. Ziegler, A. Ngan, and L. McMillan. Acquisition and Rendering of Transparent and Refractive Objects. In *Thirteenth Eurographics Workshop on Rendering*, pages 277–288, June 2002.

[16] P. Sen, B. Chen, G. Garg, S. R. Marschner, M. Horowitz, M. Levoy, and H. P. A. Lensch. Dual photography. *ACM Trans. Graphics (Proc. SIGGRAPH 2005)*, 2005.

[17] L. Torresani and A. Hertzmann. Automatic non-rigid 3D modeling from video. In *Europeaon conference on computer vision (ECCV) (2)*, pages 299–312, 2004.

[18] J. Unger, A. Wenger, T. Hawkins, A. Gardner, and P. Debevec. Capturing and rendering with incident light fields. In *Eurographics Symposium on Rendering: 14th Eurographics Workshop on Rendering*, pages 141–149, June 2003.

[19] Y. Weiss. Deriving intrinsic images from image sequences. In *Eighth IEEE International Conference on Computer Vision*, volume 2, pages 68–75, 2001.

[20] T. Weyrich, H. Pfister, and M. Gross. Rendering deformable surface reflectance fields. *IEEE Trans. on Visualization and Computer Graphics*, 11(1):48–58, January 2005.

[21] T. Zickler, S. Enrique, R. Ramamoorthi, and P. Belhumeur. Reflectance Sharing. In *Eurographics Symposium on Rendering*, 2005.

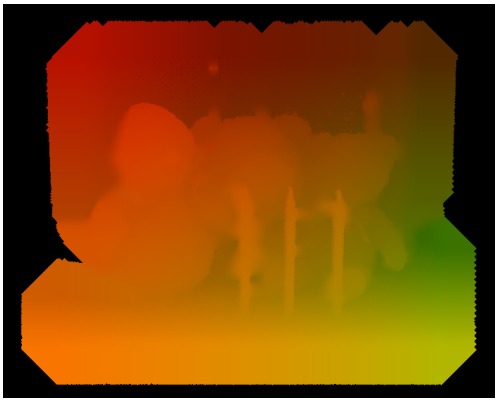

Figure 6: Visualization of the mapping function $\mathcal{M}_j^{-1}$ from the camera to the projector space. The color value represents the coordinate of the projector pixel illuminating this scene point (the red channel for x, the green channel for y). Note its similarity to a range image.

Figure 7: Illumination with a 4D incident light field. Using the presented interpolation framework, arbitrarily high resolution light fields can be projected into the scene. By projecting the same checkerboard pattern from a small but densely sampled range of positions we simulate a projector with an extremely wide aperture having only a small focus region in the middle of the scene. The original projectors were approximately 8.72 cm apart.

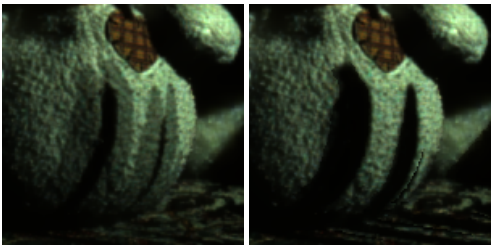

Figure 8: Comparison between naïve blending (left) and compositing after shadow warping and stenciling (right) half-way between two key frames ($\alpha = 0.5$). The left image shows cross-fading between two shadows; we obtain a single shadow using our pipeline.

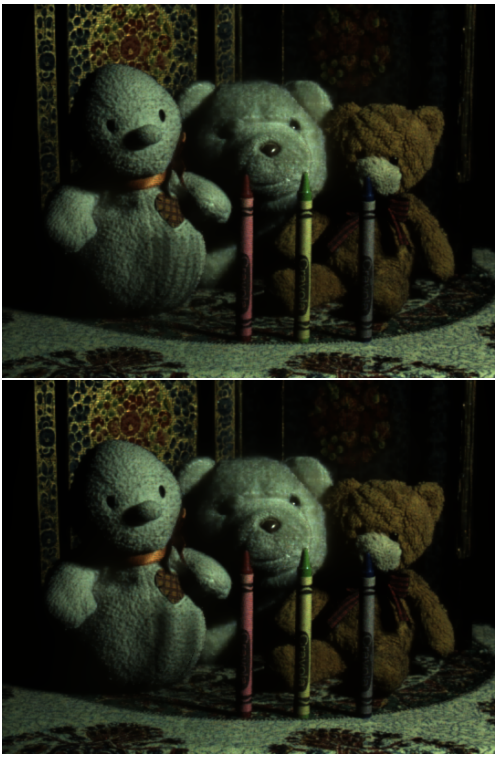

Figure 9: Simulation of an area light source. Adding three of the original input images results in severe banding in the shadow (top). By integrating over 15 interpolated frames corresponding to the same extend of the light source we can generate a soft shadow without banding (bottom).