Patch Optimization for Surface Light Field Reconstruction

Wei LI†(a), Huajun GONG†, Chunlin SHEN†, Nonmembers, and Yi WU†, Member

SUMMARY Surface light field advances conventional light field rendering techniques by utilizing geometry information. Using surface light field, real-world objects with complex appearance could be faithfully represented. This capability could play an important role in many VR/AR applications. However, an accurate geometric model is needed for surface light field sampling and processing, which limits its wide usage since many objects of interests are difficult to reconstruct with their usually very complex appearances. We propose a novel two-step optimization framework to reduce the dependency of accurate geometry. The key insight is to treat surface light field sampling as a multi-view multi-texture optimization problem. Our approach can deal with both model inaccuracy and image to model misalignment, making it possible to create high-fidelity surface light field models without using high-precision special hardware.

key words: surface light field, appearance reconstruction, photo-realistic rendering

1. Introduction

Modeling and photo-realistic rendering the real-world objects receive significant attention in computer vision and graphics communities. The major problem is how to accurately and effectively estimate the complex appearance. To solve such inverse problem, one line of research focuses on fitting the mathematical reflection models, such as the bidirectional reflectance distribution function (BRDF), to describe the reflectance properties of the real world objects. However, those mathematical models are either too simple to describe the real world objects or too computation expensive. On the other hand, some researchers focus on solving this inverse problem in a simple and automatic way by recording emitted lights in all views, which is also named the light field technique [1]. Specifically, by sampling from a dense set of viewpoints, the light field can be reconstructed and describe the objects’ appearance effect in all views using a two-plane parameterization. As the light field based methods do not rely on any reflectance model as well as geometry, they are very suitable to photo-realistically render the objects with various reflectance properties and complex geometry.

Nevertheless, the light field related approaches have some drawbacks in terms of data complexity. A huge amount of data is required in order to achieve high fidelity rendering. Thus, some hybrid solutions combining acquired photos with some geometric information such as Lumigraph [2] and Surface Light Field (SLF) [3]–[5] have been proposed. As a seminal work, Miller et al. [3] parameterized the 4D light field function over the object surface using the geometry information. In this new parameterization, every ray of light is indexed by two parameters (u, v) that define a position on the surface along with two others (s, t) that define the light direction emitted from that position. Benefit from geometry information, surface light field do not only reduce the storage but also produce sharper rendering results than the two-plane light field parameterization.

As geometry measurement plays an essential role in parameterizing surface light field function, incorrect or inaccurate geometry would lead to errors in the parameterization. Regarding this problem, Lambert et al. [6] introduced a sampling criterion in order to optimize the smoothness of outgoing radiance in the angular domain. This criterion replaces the actual surface with a parametrization of surface light field function, which could be thought of refining the geometry.

Instead of refining the geometry directly, we propose a method of optimizing the surface light field samples to achieve robust and high fidelity modeling and rendering, which is first outlined by Wei et al. [7]. Concretely, we sample input views of the surface primitive (we use the triangle on the surface mesh) into the fixed size patches. As we notice that inaccurate geometry makes those sampled patches misaligned along with varying views, the insight of our paper is that we could optimize those sampled patches to remove the influence of inaccurate geometry. Thus, before pushing those patches into the conventional surface light field decomposition and compression procedure [3]–[5], we re-align those sampled patches using a two-step strategy. First, we select the patches over all surface primitives (triangle) which can best represent the diffuse property. Second, for each primitive, we separately align all other patches to the diffuse one. We implement our approach on modern GPU and render pipeline based on Chen’s work [5]. Experiments show that our method could generate better result compared to the baseline [5].

2. Approach

2.1 Surface Light Field Partition

The surface light field was first introduced in [3] to sample light filed on a parametric surface directly. The sur-
face light field can be presented as a four-dimensional function $L(u, v, s, t)$, where $(u, v)$ define location on the surface and $(s, t)$ represent view directions. To implement surface light field with modern rendering hardware, we disassemble $L(u, v, s, t)$ across small surface primitives and build the approximations for each part independently according to [5]. Concretely, we sample and construct a set of small vertex light field $L^v(u, v, s, t)$ within the area covered by one ring triangle neighbor of vertex $x$ to approximate $L(u, v, s, t)$.

In the implementation, the vertex light field function $L^v(u, v, s, t)$ can be represented as a matrix $L^v[u, v, s, t] \in \mathbb{R}^{m \times n}$ by discretizing over surface patches and solid view angles. The columns $n$ of this matrix indicate the sampled camera views, while the rows $m$ encode surface locations. Additionally, according to the theory of Dichromatic Reflection Model [8], we would also like to separate the diffuse component $D^v[u, v, s, t]$ from the other view-dependent lighting effects, which is named the residual component $G^v[u, v, s, t]$. Then, $G^v[u, v, s, t]$ can be compressed by further decomposition into a sum of a small number products of lower-dimensional functions, which can be thought of decoupling the variation in the surface texture from the variation in lighting, according to [5] as follows:

$$L^v[u, v, s, t] = D^v[u, v, s, t] + G^v[u, v, s, t]$$

$$\approx D^v[u, v] + S^v[u, v]V^v[s, t]$$

Here, $S^v[u, v]$ and $V^v[s, t]$ are named the surface map matrix and the view map matrix of vertex $x$, respectively.

2.2 Patch Generation

We assume that we have already captured the target object with a surface mesh $M$ (shown in Fig. 1(a)), which consist of $n_f$ triangles $F = \{f_1, \ldots, f_{n_f}\}$, $n_v$ vertices $X = \{x_1, \ldots, x_{n_v}\}$ and $N$ registered input images $I = \{I_1, \ldots, I_N\}$ (shown in Fig. 1(a)), which are registered using calibration board in our experiment.

As the vertex light field matrix $L^v$ is constructed by concatenating the sampled data of all triangles within its one ring neighbor. Thus, we first generate sampled patches for each triangle in all visible view separately, which is named $L^f$. Note that, $L^f$ is used to decompose and compress the light field data, however, our two-step optimization in Sect. 3 is carried out on each face $L^f$.

As not all input images are visible to face $f$, we first carry out a visibility check to find the subset of input images $\hat{I}^f$ which can be used to construct $L^f$. For images in $\hat{I}^f$, we back-project the face using corresponding camera parameters onto image space to generate a set of small image patches $P^f = \{P^f_1, \ldots, P^f_{n_p}\}$. Note that, we enforce all patches to the same size by bilinear interpolating sampled color. The size is simply chosen from the maximum projection size.

In Fig. 1(e), we visualize the patches $P^f$ of the specific triangle, which is highlighted in red in the top row of Fig. 1. We can see that the line patterns suffer from misalignment and blur caused by wrong sampling from the inaccu-
Equation (2) can be solved by alternative searching $t_i$ and computing proper $D^f$. However, it is too computation expensive and unnecessary. Thus, we simplify the problem using a two-step strategy, which is visualized in Fig. 2. First, we notice that a proper diffuse patch $D^f$ can be directly selected from input samples. So we formulate this selection as an energy optimization problem. Then, we search $t_i$ separately for each view to align $P^f_i$ to the selected diffuse patch $D^f$.

3.1 Diffuse Color Patch Selection

The straightforward way to extract diffuse color $D^f$ of triangle $f$ from sampled patches $P^f$ is to compute mean or median of $P^f$. This is obviously problematic when patches are badly misaligned or with large luminance changes. We solve this problem by seeking a best one in all sampled patches $P^f$. We formulate such patch selection for all triangles $P = \{P^f_1, P^f_2, \ldots, P^f_n\}$ as an energy function:

$$E(P) = \sum_{f \in F} \left( E_i(P^f_i) + E_q(P^f_i) + \sum_{j \in N_f} E_s(P^f_i, P^f_j) \right)$$

Here, $N_f$ is the adjacent faces of face $f$. The first part of Eq. (3) is luminance term $E_i$, which is used to seek the one captured in good luminance condition without specular light. To achieve this objective, we compute the luminance mean $l^f_i$ and variance $\alpha^f_i$ for each patch $P^f_i$. We drop 5% samples with lowest mean luminance, as it is likely to be captured without enough light. Then we extract most possible luminance mean $\bar{l}$ and variance $\bar{\alpha}$ using the mode of $\{l^f_i\}$ and $\{\alpha^f_i\}$. We define the energy of luminance term as $E_l(P^f_i) = (l^f_i - \bar{l})^2 + (\alpha^f_i - \bar{\alpha})^2$. As specular highlight may cause obvious difference $\alpha^f_i$ comparing to $\bar{\alpha}$, $E_l$ would favor the illuminated patches without highlight.

The second part of Eq. (3) is the quality term $E_q$, which is used to indicate the sample quality of target patch. We simply use the footprint on the image to define $E_q(P^f_i) = \pi(P^f_i)$, here $\pi(P^f_i)$ denote the projection size of $P^f$ on image $i$.

The last part of Eq. (3) is the smooth term $E_s$, which is used to favor adjacent triangles selecting patches with similar color on the share edge. $E_s$ is defined as $E_s(P^f_i, P^f_j) = \sum_p d_{edge} ||\eta(P^f_i, p) - \eta(P^f_j, p)||^2$. Here, point $p$ comes from the share edge of $f$ and $f'$. While $\eta(P^f_i, p)$ is the function to get RGB information of point $p$ in patch $P^f_i$. The Eq. (3) forms a Markov Random Field (MRF) problem, we solve it using Tree Re-weighted Message Passing (TRMP) algorithm [10].

3.2 Residual Color Patch Repairing and Decomposition

With optimized diffuse patch $D^f$ for triangle $f$, we carry out an individual alignment on every patch in $P^f$ to form a better light field matrix. We rewrite the Eq. (2) by treating $D^f$ as known and adding norm of $t_0 + t_i$ to punish unnecessary movement, $\max_0, \mathcal{M}(D^f, \phi(P^f_i, t_i))/\|(t_0 + t_i)^2\|$. Here, $t_0$ is used to avoid zero shift and adjust the weight of $t_i$. We solve the problem by greedy search $t_i$ within the distance limit $t_{max}$. We set $t_0$ to (15,15) and $t_{max}$ to 3 pixels in our experiments empirically. Note that, after found each best $t_i$, we also record $C^f_i = \mathcal{M}(D^f, \phi(P^f_i, t_i))$ as the final similarity score for optimized $P^f_i$. We notice that shift could re-align most patches, but there are still some bad sampled patches cannot be repaired. In this case, we just trim those patches, which can be picked out using $C^f_i$.

After aligning all patches, we construct optimized the vertex light field matrix $L^s$ and then derive corresponding residual matrix $G^s$. Note that, we also re-sample the camera views of raw $G^s$ to uniform cover a hemisphere using the Hemispherical Harmonics [11] method. We decompose the re-sampled residual color $G^s$ into $k$ terms surface map and view map using SVD as $G^s = S^s V^s$ to compress the light field data, where $S^s$ is a $m \times k$ matrix of left singular vectors product the diagonal matrix of singular values sorted in decreasing order, $V^s$ is a $k \times n$ matrix of right singular vectors. Here, $k \ll n$ and is set to 4 in our implementation. Since we are interested in the first $k$ approximation terms. Instead of performing a full SVD, we use Power iteration method to iteratively calculate first $k$ terms.

After decomposition, all the surface and view maps are tiled and stored in the textures and can be further compressed using standard texture compression methods. In rendering state, given a novel camera position, the rendering is done by computing the view direction values $(s,t)$ for each surface point $(u,v)$ and then indexing the surface light field matrix using $(u,v,s,t)$ to extract the color value.

4. Experiments

We demonstrate our approach by modeling objects with different materials. Note that all experiments are performed on
We first apply our system to a real object, a 15cm tall bear statue with high specularity. This statue consists of two parts, the silver paint on the body, the reflective black glass with sharp text on the base. The model especially the base part cannot be accurately reconstructed using the structured light scanner in our experiment. Note the geometry error shown in Fig. 1. We compare the rendering results with and without our optimization using the same camera view of one input image, which is shown in Fig. 3 (a)∼(d). Without optimization, obvious artifacts exist on the base of the statue. With optimization, our method could efficiently remove those artifacts. We also show the comparison of different decomposition terms in Fig. 3 (e)∼(g). Higher decomposition terms can always produce more realistic rendering results, but also require more memory storage.

We also use a synthetic 10cm tall can model of Coca-Cola to evaluate our method. The input images are rendered under global illumination using the Autodesk Maya. While the rendering camera positions are uniformly sampled on a sphere, which is with 1m radius and center on the model. And all camera directions are set to look at the model’s barycenter. In this experiment, we add Gaussian noise $t \sim N(0, 5mm)$ and $R \sim N(0, 1^{\circ})$ to the camera position and rotation to simulate image registration errors. As shown in Fig. 4, our method can avoid blurring and produce better rendering result with sharp text and captured specularity. We show the diffuse texture in close view with a green border in Fig. 4. The one using mean color suffers from blurring, while the diffuse texture after our patch selection optimization could keep the sharp text.

To have a more objective evaluation, we use normalized cross-correlation (NCC) to metric the similarity between the rendered images and original input images, which are shown in Table 1. Here, NCC is defined as $\frac{\sum I_A \cdot I_B}{\sqrt{\sum I_A \cdot I_A \times \sum I_B \cdot I_B}}$, where $I_A$ and $I_B$ are vectorized data from target and source image for comparison respectively. Note that, the NCC value is a percentage and the value 100% is returned when we compare an image with itself. The input and running information is also shown in the table. Note that the total time is the sum of resampling, optimization, compression and texture generating time. From this table, we can see our optimization could increase the fidelity of rendering result over conventional method in those two dataset with imperfect input data.

### 5. Conclusions

The surface light field has the ability to photo-realistic visualize the real world objects with complex appearance. However, sampling light field over surface mesh requires accurate geometry and perfect image to model registration, which always makes the scanning system expensive and not...
robust. Thus, we propose a novel optimization framework to reduce the dependency of scan accuracy. We divide the sampled patches into diffuse and residual parts, which are later optimized with our two-step strategy. Based on our patch optimization scheme, the surface light field of real-world objects can be easily and robustly reconstructed even without high-precision hardware.

Acknowledgments

This work was supported in part by the National High-tech R&D Program (863 Program) of China under Grant 2015AA015905. And this work was presented in part at IEEE Conference on Multimedia Information Processing and Retrieval (MIPR), Miami, USA, April 2018.

References


