

REGION-BASED ALL-IN-FOCUS LIGHT FIELD RENDERING

Goran Petrovic¹, Aneez Kadermohideen Shahulhameed¹, Sveta Zinger¹, Peter H. N. De With^{1,2}

¹Eindhoven University of Technology 5600 MB Eindhoven, The Netherlands

²CycloMedia Technology 4180 BB Waardenburg, The Netherlands

ABSTRACT

Light field rendering is an approach to synthesize virtual views of a scene from a set of original images. When minimizing the number of images for rendering, the light field may become under-sampled, leading to *aliasing* artifacts. To render an under-sampled light field in *high quality* and without aliasing artifacts is a challenge. We present a light field rendering algorithm with *region-based focus* to create all-in-focus virtual views. Our algorithm was compared to: (a) ground truth images and (b) a state-of-the-art technique for rendering under-sampled light fields. Extensive rendering experiments confirm that our algorithm provides visible quality improvement, quantified as about 10% RMSE reduction. However, the subjective improvement is larger and it produces images comparable to the ground truth. This algorithm contributes to practical applications of light field rendering, such as image generation in stereoscopic displays and Free-Viewpoint Video (FVV).

Index Terms— Light field rendering, Free-Viewpoint Video, Aliasing, Image Quality.

1. INTRODUCTION

In an attempt to anticipate future deployment of 3D-video systems [1], the MPEG community has recently singled out two broad application scenarios: *Three-Dimensional Television (3D-TV)* and *Free Viewpoint Video (FVV)*. The 3D-TV applications enable viewers to perceive depth in the displayed scene. Two closely spaced images of the same scene are displayed simultaneously to create the effect of depth. With FVV, a scene can be displayed from different viewpoints in an *interactive* fashion. The user either selects an arbitrary new viewpoint and a viewing direction, or the user's movements are continuously tracked and the displayed content automatically adjusted to the new position.

Light field rendering [2] [3] is used for view synthesis in state-of-the-art 3D-TV [4] and FVV systems [1]. These rendering techniques can be conceptually viewed as sampling and interpolation of the plenoptic function [5]. Each pixel value in an original camera represents a sample of the plenoptic function. To synthesize a view from a virtual camera, the direction and angle of a viewing ray (corresponding to a pixel

in the virtual view) are used to select the nearby samples in the original cameras.

Levoy and Hanrahan [2] first demonstrated the suitability of *densely sampled light fields* to synthesize virtual views. Their algorithm assumes the scene object to be close to the image plane and fixes the image plane at a particular depth when rendering. This *a-priori* decision as to what part of the scene can be rendered *in focus*, e.g., without aliasing is undesirable. The relationship between the sampling density and non-aliased rendering was first theoretically analyzed by Chai *et al.* [5]. To avoid aliasing, they suggest the light field must be sampled such that the maximum disparity between adjacent images does not exceed one pixel. The number of images required to guarantee a maximum disparity of one pixel is impractically high. Therefore, physically recorded light fields are always an under-sampled representation of the complete light field. For this reason, many researchers have investigated the possibility of solving the problem of *rendering without aliasing* in the rendering algorithm itself. We take the same approach in this paper.

Our algorithm renders under-sampled light fields of an object or a scene, while minimizing aliasing. The algorithm assigns pixels in the synthesized image to different depth layers, and renders an all-in-focus virtual image. The algorithm is flexible, as it does not require pre-processing to determine the best depth assignment for a pixel. Instead, it makes this decision dynamically, during rendering, while adapting to the camera and scene geometry. The key to our improvements is the observation that rendering quality deteriorates if the spatial support of light field rendering filters extends over object boundaries. To this end, we incorporate an *segmentation step* to roughly separate the image into regions, and produce an all-in-focus image by combining the depth layers per-region. The use of image segmentation to control the spatial support of light field filters is unique to our approach.

2. RELATED WORK

We discuss the related work in the areas of alias suppression and all-in-focus light field rendering.

To *suppress aliasing*, Levoy and Hanrahan [2] propose to *prefilter* the light field. Prefiltering can be implemented by first over-sampling along the camera-spacing dimensions



Fig. 1. Region-based algorithm for all-in-focus light field rendering

and then applying a discrete low-pass filter. Over-sampling is often impractical as the data sets are large. Moreover, this approach makes an *a-priori* decision as to what parts of the scene will be rendered in focus. Isaksen *et al.* [6] introduce the concept of a *movable focal surface*. They achieve in-focus rendering of objects at different depths by dynamically positioning the focal plane. To reduce the aliasing artifacts, they increase spatial support (aperture) of the reconstruction filter. However, their *wide-aperture filter* also smoothes out high-frequency content and tends to produce blurry renderings in many practical situations.

A number of recent approaches extend the ideas of Isaksen *et al.* [6] and combine renderings for multiple focal planes to produce an *all-in-focus* image, e.g., [7]. Different focal planes are used to build a simple geometric model of the scene, consisting of a small number of depth layers. Most of these techniques are implemented as pre-processing steps and require user input. A notable exception is the algorithm of Takahashi and Naemura [7] that uses spatial consistency of different filters to estimate the depth of different scene elements during rendering. We have recently performed an extensive study to compare most of the above proposals to reduce aliasing when rendering under-sampled light fields [8]. Since our objective is to perform all-in-focus light field rendering automatically and on-the-fly, the method [7] is most closely related to our work.

3. REGION-BASED ALL-IN-FOCUS LIGHT FIELD RENDERING

Our algorithm builds on earlier in all-in-focus rendering using multiple focal planes. However, we are adding a segmentation step in the core of the algorithm to support the creation of the focal planes. Figure 1 shows the basic steps in the algorithm.

For a given viewpoint, images are synthesized by moving the assumed depth of the focal plane used in dynamic-light field rendering [6]. This produces a set of images, each of which are focused on a particular part of the scene. Then, using a focus measure, the areas in focus in each image are identified and combined into the final rendered image in real time. To identify the regions in focus from a synthesized image, we use a focus measure for each pixel.

We briefly review the computation for the focus metric $f_n(x, y)$, which is specified as follows

$$f_n(x, y) = \sum_{-M < l, k < M} \frac{sub_n(x+k, y+l)}{(2M+1)^2}, \quad (1)$$

where $sub_n(x, y)$ is calculated by

$$\begin{aligned} sub_n(x, y) &= |A_n(x, y) - B_n(x, y)|, \\ A_n(x, y) &= \min(Cr_i + Cg_i + Cb_i), \quad i \in w, \quad (2) \\ B_n(x, y) &= \max(Cr_i + Cg_i + Cb_i), \quad i \in w. \quad (3) \end{aligned}$$

The parameters Cr_i, Cg_i, Cb_i correspond to the amplitudes of the red, green and blue channels of the i^{th} ray used in the interpolation of pixel (x, y) . The parameter w denotes the width of the aperture filter. Using variables $A_n(x, y)$ and $B_n(x, y)$, we keep track of the minimum and maximum of the sum of intensity over different color channels, and use it to estimate the smoothness of the color difference among the rays used for interpolation (defined as $sub_n(x, y)$). This difference will be small if the pixel is in focus and large if the pixel synthesized at position (x, y) is out of focus. While $sub_n(x, y)$ can be used as a focus metric, the resulting depth allocation will be noisy. Hence, we implement $sub_n(x, y)$ as a weighted average over a block of pixels using Eqn. 1. The choice of the weighted-average metric is motivated by the observation that the objects in the scene are larger than pixels [7].

However, the assumption made in [7] is that the size and shape of the block used in averaging the pixels within this block are at similar depth. This assumption is false when the block is used across a surface discontinuity. In this situation, including the pixel outliers in the matching will deteriorate the overall matching score. This motivates our contribution to introduce *segmentation*. We have used a version of *normalized cuts* for segmentation [9], as it is robust and the implementation is available. In this technique, the pixels in the image are first represented as a weighted undirected graph $G = (V, E)$ of V vertices and E edges. The nodes of this graph correspond to pixels of the image. Every pair of nodes (i, j) is connected by an edge, and the weight on each edge $s(i, j)$ is a function of the similarity [9] between nodes i and j . The similarity is measured using intensity and intervening contours in the image. The graph $G = (V, E)$ is then segmented into two disjoint complementary parts I_1 and I_2 , by removing the edges connecting these two parts. The degree of similarity between these two parts can be computed as the total weight of the edges that have been removed, denoted as $cut(I_1, I_2) = \sum_{u \in I_1, t \in I_2} s(u, t)$. The optimal partitioning of a graph is the one that minimizes this value. The algorithm uses a fraction of the total edge connections to all the nodes in the graph as cut cost, instead of the total edge weight connecting the two partitions. This fraction leads to the so-called

normalized cut $Ncut$ and is defined by

$$Ncut(I_1, I_2) = \frac{cut(I_1, I_2)}{asso(I_1, V)} + \frac{cut(I_1, I_2)}{asso(I_2, V)}. \quad (4)$$

Let us now present the complete algorithm, which is outlined in Algorithm 1. The algorithm first generates n different wide-aperture renderings [6] by placing the focal planes uniformly between the maximum depth (Z_{max}) and minimum depth (Z_{min}), and one at the optimal depth Z_{opt} [5] from plenoptic sampling. The maximum and minimum depths are assumed to be available from the capturing process [3]. For each of the n rendered images (excluding the image at Z_{opt}), we compute the color-based metric sub_n according to Eqn. 2. Next, we segment the image rendered at Z_{opt} into regions. Subsequently, we perform matching within each of the individual regions to compute f_n in Eqn. 2. The algorithm then assigns each region to a particular focal plane by estimating the minimum f_n over all the pixels that constitute the region. The “all-in-focus” image generation process consists of selecting each region and assigning it to appropriate depth depending on the region-based matching.

Figure 2 illustrates the rendering at depth Z_{opt} , the final rendered image and the comparison of estimated depth layers which illustrates the effectiveness of our region-based focus algorithm. The data set called “Toys” (256 images, 320x240 pixels) was obtained from the MIT Light Field archive ¹.

Algorithm 1: REGION-BASED ALL-IN-FOCUS RENDERING ($l_d, Z_{min}, Z_{max}, (x, y), w, n, \text{threshold}$)

```

foreach layer  $i=1$  to  $n$  do
   $Z \leftarrow Z_{min} + i \times \frac{Z_{max} - Z_{min}}{n}$ ;
   $sub_i \leftarrow$  color difference in width  $w$  for each pixel
  using WideAperture ( $Z, (x, y), w$ );
  Generate  $I_{intermediate}$  using  $Z_{opt}$ ;
  Segment  $I_{intermediate}$ ;
  foreach segment  $seg$  in  $I_{intermediate}$  do
    foreach pixel  $p$  in  $seg$  do
       $f_n \leftarrow$  region-based matching based on
      segmentation;
      calculate sum of  $f_n$  over all pixels in the
      segment;
    if  $sum \leq \text{threshold}$  then
      Assign  $seg$  to  $Z_k$ , to  $k \in [1, n]$ ;
      render all pixels in segment using quadrilinear
      filter
    else
      do a weighted blending from all the  $Z_i$ ;
  return all-in-focus rendering

```

¹Unfortunately, the MIT data set is no longer available for download at the time of this writing.

4. RESULTS

We have implemented our region-based rendering algorithm in Matlab and have used different light field data sets to evaluate its performance. Data set “Jewel” (289 images, 672x420 pixels) was captured with a two-dimensional camera array and is accessible from the Stanford New Light Field Archive [3].

In all our experiments, we compute the focus metric (Eqn. 1) using $M = 8$ (8x8 pixel block) and $w = 12$ (12 rays selected in a diamond pattern around the ray to render).

We evaluate the performance of our algorithm both visually and objectively. For the objective comparison, we apply the *Root Mean Square Error* (RMSE) as a metric to quantify the rendering error, which is defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^I \sum_{j=1}^J (R(i, j) - O(i, j))^2}{I \times J}}. \quad (5)$$

In the above, R and O are the rendered and original image, respectively, and $I \times J$ stands for the image resolution. This image quality metric can be applied directly when comparing the relative performance of two rendering algorithms. However, it does not quantify the absolute quality of the rendered images. Our algorithm creates *virtual views* of the scene, i.e., images not present in the original data set. A plausible way to verify its correctness is to compare the rendered virtual image to the original image from the same viewpoint. To obtain the *ground truth* images, we have divided the entire data set into two subsets by assigning every even row and column from the camera array to the *new input* data set and every odd row and column to the *ground truth* data set.

The results for the *RMSE* comparison are shown in Table 1. The first result column refers to our proposed algorithm and the second to our implementation of the algorithm by Takahashi and Naemura [7]. For the sake of completeness, we also include the score for rendering at the optimal plane Z_{opt} [5]. We have used 5 focal planes with all three algorithms. The columns of Table 1 correspond to virtual images synthesized at different positions in the array (indexed by the array row and column). The synthesized virtual view was compared to the ground-truth image at the same position. We present only a limited number of representative scores, as we have observed a similar trend for other viewpoints. Both the Takahashi’s and our algorithm perform significantly better than the Z_{opt} algorithm which uses a single plane only (as could be expected). Our algorithm shows a small *RMSE* improvement of, on the average, 10% over Takahashi’s. This absolute difference score is partly due to the structure of the test images with a large uniform background. However, a visual comparison of our algorithm to Takahashi’s [7] clearly illustrates the advantages of our algorithm. Figure 3 shows a magnified view on the objects. The square-template matching of [7] is insufficient to reproduce the complicated patterns on

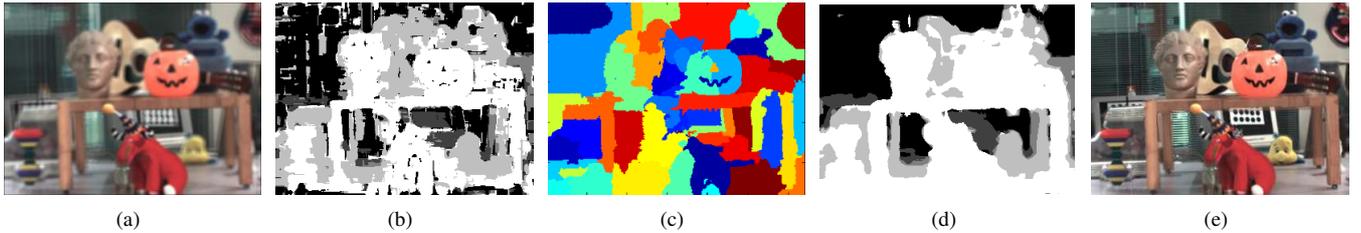


Fig. 2. Algorithm results: (a) Rendering at Z_{opt} , (b) Estimated depth layers without segmentation, (c) Segmentation of the Z_{opt} -rendered image, (d) Estimated depth layers with segmentation, (e) All-in-focus rendering.

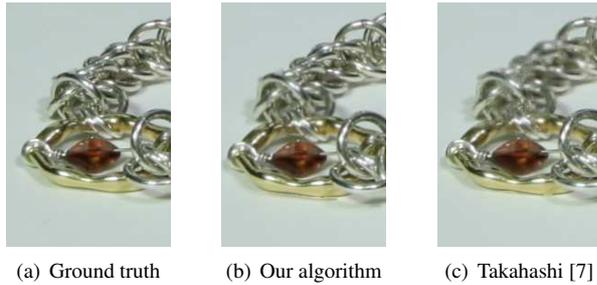


Fig. 3. Comparison with existing algorithms.

the object surfaces and introduces significant rendering blur. Our region-based approach produces sharp renderings in the same area, as evidenced by the ground-truth comparison.

	Proposed	Takahashi's	Z_{opt}
Position(8,8)	1575.0	2048.3	3350.8
Position(10,8)	2061.6	2304.5	2761.7
Position(6,10)	1907.6	2107.8	2674.8
Position(10,10)	2117.8	2247.8	2781.7

Table 1. Error analysis using RMSE on “Jewel” light field.

5. DISCUSSION AND CONCLUSION

Our improvement has been realized at the expense of a higher complexity, caused by adding image segmentation. It is known that image segmentation can be a *computationally intensive* task. The segmentation is carried out while performing the rendering. This is why we have implemented a relatively simple segmentation only in the form of color segmentation. This is an attempt to balance complexity against quality improvement. The *robustness* of our color-based focus metric depends on two assumptions: (1) no occlusions occur, and (2) object surfaces are Lambertian. Nevertheless, this metric appears to be robust, as we experimentally demonstrate by rendering scenes with complex occlusion relationships (such as in “Toys”) and strong specular reflections (like in “Jewels”). We thus offer empirical evidence that our approach is not sensitive to the violation of these assumptions and has a certain degree of robustness.

This paper contributes in two aspects. First, we have provided a rendering algorithm that includes image segmentation to maximize the quality by controlling the light field rendering at object edges and discontinuities in the signal, thereby reducing aliasing artifacts. Second, we performed an experimental analysis of the impact of scene-depth discontinuities on all-in-focus rendering quality. Rendering experiments with two different light field data sets demonstrate that the proposed algorithm improves rendering quality both quantitatively (RMSE reduction of 10% on average) and, more significantly, perceptually. We expect similar performance gains for other data sets, as the algorithm is not constrained by the accuracy of segmentation. Future work will be to adaptively optimize the selection of segmentation thresholds for a given scene, as they are currently set empirically.

6. REFERENCES

- [1] A. Kubota, A. Smolic, M. Magnor, M. Tanimoto, T. Chen, and C. Zhang, “Multiview imaging and 3DTV,” *IEEE Signal Processing Magazine*, vol. 24, pp. 10 – 21, Nov. 2007.
- [2] M. Levoy and P. Hanrahan, “Light field rendering,” in *SIGGRAPH*, 1996.
- [3] B. Wilburn, N. Joshi, and V. Vaish et al., “High performance imaging using large camera arrays,” *ACM Transactions on Graphics*, vol. 24, pp. 765 – 776, 2005.
- [4] W. Matusik and H.-P. Pfister, “3D TV: A scalable system for real-time acquisition, transmission, and autostereoscopic display of dynamic scenes,” in *SIGGRAPH*, 2004.
- [5] J. X. Chai, X. Tong, S. C. Chan, and H. Y. Shum, “Plenoptic sampling,” in *SIGGRAPH*, 2000.
- [6] A. Isaksen, L. McMillan, and S. Gortler, “Dynamically reparameterized light fields,” in *SIGGRAPH*, 2000.
- [7] K. Takahashi and T. Naemura, “Layered light-field rendering with focus measurement,” *Signal Processing: Image Communication*, vol. 21, pp. 519 – 530, July 2006.
- [8] Aneez Kadermohideen Shahulhameed, “Region-based all-focused light field rendering using color-based focus measure,” in *MSc Thesis, Eindhoven University of Technology*, Oct. 2008.
- [9] T. Cour, F. Benezit, and J. Shi, “Spectral segmentation with multiscale graph decomposition,” in *IEEE International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2005.