Adaptive Layer Extraction for Image Based Rendering

Jesse Berent, Pier Luigi Dragotti, Mike Brookes

Communications and Signal Processing Group
Electrical and Electronic Engineering Department
Imperial College, Exhibition Road, London SW7 2AZ, United Kingdom
{jesse.berent04, p.dragotti, mike.brookes}@imperial.ac.uk

***Abstract*—Image based rendering is a promising way to produce arbitrary views of a scene using images instead of object models. However, depth variations and occlusions cause blurring in the rendered images. The solution is to use some geometrical information in order to steer the interpolation filters according to the depth. The level of detail of this geometry is often predetermined. In this paper, we present a method for extracting depth layers in the presence of occlusions for image based rendering. Moreover, we show how the layer extraction can be made to estimate depth layers in an adaptive manner, based on the spectral analysis of the plenoptic function. The rendering system therefore automatically adapts the number of depth layers based on the scene and the spacing of the sample cameras.**

I. Introduction

Image based rendering is a method of generating arbitrary views of a scene that differs from the traditional computer graphics approach. Instead of rendering views of 3D scenes by projecting objects and their textures, new views are rendered by interpolating available nearby images. That is, the scene is not represented by its objects but by the light rays that are captured by the cameras. This is the case for example in the popular Light Field (LF) [1] and Lumigraph [2], [3] representations. New views are obtained simply by interpolating from the sampled light rays. The advantage of such a method is that little or no geometry of the scene is required, as opposed to a full geometric model which can be very difficult to obtain from natural images. Moreover, the rendering algorithms produce convincing photorealistic results since the interpolated viewpoints are obtained through combinations of real images. The main drawback of such a representation is the fact that a huge amount of data (typically hundreds or thousands of images [1]) need to be captured, stored and transmitted.

Image Based Rendering is a sampling and interpolation problem. It is therefore interesting to study the problem in a traditional sampling and interpolation framework. That is, to estimate the spectrum of the signal at hand and determine the sampling frequency necessary for a reconstruction free from aliasing. All the visual information can be characterized with a single seven dimensional function called the plenoptic function [4]. In [5] as well as [6], the authors show that the spectrum of the plenoptic function is approximately band-limited by the maximum and minimum depths in the scene and has a distinctive bow-tie shape. From this spectrum, the authors are able to deduce the number of samples (i.e. images) necessary for an aliasing-free rendering. They also show that the interpolation filter can be steered with an angle that depends on the depth of the scene in order to reduce aliasing. However, in many cases, the sampling period of the cameras is too large (i.e. not enough images) and this simple interpolation is not sufficient. In this case, the scene must be split into different depth layers such that each layer has a tighter bow-tie and can be individually rendered free of aliasing. Therefore, there is a clear tradeoff between the number of images, the number of layers and the depth variation.

Layers have been used for many applications in multi-view images. Several layered representations have been proposed such as the layered-depth-images [7]. They have been used successfully in free-viewpoint video [8] as well. However, these methods are designed to produce an accurate depth map of the scene. New views of the scene are rendered through warping of the layers. This is very sensitive to errors in the depth reconstruction.

Other layered representations are designed for image based rendering such as the coherent layers in Pop-up light field [9] and plenoptic layers [10] (a.k.a plenoptic manifolds) and are based on approximate geometry rather than exact depth. In [5], the authors show that a certain number of layers is optimal for a given scene and number of cameras. Therefore extracting more layers is superfluous. Some scenes do not require advanced layer extraction methods. In fact, the layer extraction should be tailored to the scene and the samples of the Light Field in an adaptive manner. That is, there is a relation between the complexity of the scene (depth variation, occlusion, non-lambertian) and the layer extraction. A simple scene with small depth variation only requires very few depth layers which can be extracted very quickly, e.g. testing for two different depths only. A scene with large depth variations requires many different rendering depths and therefore the layer extraction must test for more depths. Following this analysis, the authors in [11] and [12] reconstruct an approximate depth map based on interpolating images with different constant depth filters

MMSP'09, October 5-7, 2009, Rio de Janeiro, Brazil.
978-1-4244-4464-9/09/$25.00

and fusing in-focus regions. In [9], the user manually extracts layers until satisfied with the rendered result.

The rendering system presented in this paper contains novelty with respect to [5], [11], [12] and [9] in several ways. First, we do not assume known geometry as in [5]. Second, we do not require user interaction as in [9]. Third, the depth estimation in [11] is block-based which may cause reconstruction artifacts in the boundaries of layers and does not take into account occlusions. Finally, [12] does not take into account occlusions and relies on the user for the number of layers. In contrast, our depth estimation and interpolation both take occlusions into account and the number of layers is adaptively estimated.

The paper is organized as follows: In Section II, we discuss the structure of the Light Field and look at its spectrum. Section III derives a patch-based layer extraction using a simple matching criterion. In Section IV, we show how the method can be tailored to the scene observed by automatically adapting the number of layers. Section V illustrates some results and we conclude in Section VI.

II. Light field structure and sampling

As mentioned above, the rendering of new views from a set of sample images is a sampling and interpolation problem. It is therefore interesting to look at the spectrum of the data at hand (i.e. the plenoptic function). This problem has been studied by Chai et al. in [5] for the scenario of multiple views along a baseline as illustrated in Figure 1(a). In this context, the plenoptic function is parameterized with $I(x, y, t)$ where (x, y) are the image coordinates and t is the position of the camera along the baseline. Using the pinhole camera model, it can be shown that a point $\mathbf{p} = (x, y, 0)$ with depth $z_m = f/d_m$ in the image at $t = 0$ will be projected onto the image in t_k as:

$$\mathbf{p}_{m,k} = (x_{\mathbf{p}} - d_m t_k, y_{\mathbf{p}}, t_k), \tag{1}$$

where d_m is the disparity gradient and f is the focal length of the camera. This relation enables one to obtain some insights on the structure and the spectrum of the multiview data. Indeed, points in space are mapped onto lines in the LF and lines with a larger slope will always occlude lines with a smaller ones as illustrated in Figure 1(b). This relation also enables one to show that the spectrum of the plenoptic function is approximately bound by a bow-tie delimited by the maximum depth $z_{max} = f/d_{min}$ and minimum depth $z_{min} = f/d_{max}$ as depicted in Figure 1(c). Given this spectrum, it can be shown that the optimal interpolation filter is steered to the mean disparity gradient. Using this interpolation, the minimum sampling rate $\Delta t = t_{k+1} - t_k$ to avoid aliasing in the t-axis is given by [5] $\Delta t = \frac{1}{Bfh}$, where $h = [1/z_{min} - 1/z_{max}]$ and $B = 1/2\Delta x$ which is related to the cut-off frequency of the camera.[1] This Δt only takes into account the knowledge of the minimum and maximum depths in the scene. The light field can be segmented into M constant depth layers with uniformly spread disparity gradients d_m. In this way, each layer has a tighter spectrum and the new constraint becomes:

$$\frac{\Delta t}{M} = \frac{1}{Bfh}. \tag{2}$$

There is therefore an interplay between the sampling rate Δt or, equivalently the number of images, the minimum and maximum depths in the scene h and the number of depth layers M.

[1]Note that B may also be limited by the band of the texture of the objects observed. However, we assume here that this band is not limited.

III. Layer extraction and rendering

In this section, we present a layer extraction algorithm that takes into account the particular structure of the LF. That is, it uses the fact that points in space are mapped onto lines in the LF and occlusions occur in a specific order. Moreover, it is designed to deal with any number of images (i.e. two or more). Note that the constraints applied to the energy minimization are the same as in [10], [13]. However, instead of relying on active contours and the level-set method, we use a patch-based algorithm to find potential layer boundaries. This enables a drastic speed-up in computation times while being very effective at finding layer edges. The second part of the section describes how new views are interpolated using the segmented layers and the knowledge of occlusions.

A. A patch-based layer extraction

Similar to the stereo methods used in [14] and [15], we assume that layer boundaries occur at intensity and color discontinuities. An initial step in the layer extraction is therefore to segment a reference image into a set of patches S_n using the mean-shift algorithm [16]. Given a set of predetermined possible disparity gradients d_m, each segment S_n is assigned to a layer m using a matching criterion and an occlusion reasoning. Note that the method presented here differs from [14] and [15] in that the number of layers or assigned disparities is an input to the layer extraction. We also use more than two images if they are available. Finally, the algorithm operates in a two-pass manner instead of multiple iterations.

The layer extraction is performed by minimizing the energy functional

$$E_{tot} = \sum_{n=1}^{N} E_n(m_n),$$

where $(S_1, \ldots, S_N)$ are the patches extracted by the mean-shift segmentation and m are the layers with disparity gradient values d_m. In order to minimize the total energy E_{tot}, we minimize each of the partial energies $E_n(m_n)$ defined for each patch. The partial energies are defined as:

$$E_n(m) = \sum_{\mathbf{p} \in S_n} f(\mathbf{p}, m),$$

where $\mathbf{p} = (x, y, 0)$ is a pixel on the reference image and $f(\mathbf{p}, m)$ is a matching function for the pixel in the other images. The matching functional here is simply defined as the sum of absolute differences (SAD)

$$f(\mathbf{p}, m) = \sum_{k=1}^{K-1} |I(\mathbf{p}_{m,k}) - I(\mathbf{p}_{m,k+1})|,$$

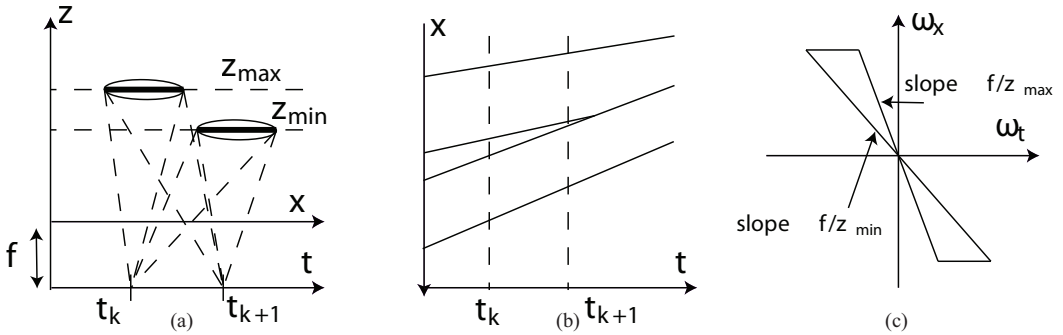


Fig. 1. 2D Light Field structure and spectrum. (a) Two layers observed by two cameras in t_k and t_{k+1}. The x-axis is the image plane, z is the depth and f is the focal length. (b) Position x of the layers on the image plane as a function of the camera position t. Points are mapped onto lines with slope inversely proportional to their depth z. (c) Given this structure, the spectrum of the light field is approximately bound by the maximum and minimum depths.

where K is the number of images under consideration and $\mathbf{p}_{m,k}$ is as defined in (1). The $I(\mathbf{p}_{m,k})$ is therefore the intensity of a point in image k. We use linear interpolation for the intensity since the points $\mathbf{p}_{m,k}$ are not necessarily integer values. The segment S_n is assigned to the layer m with

$$m_n = \underset{m}{\operatorname{argmin}}\{E_n(m)\}.$$

Once each segment has been assigned to a layer, we may build the layer index for all the images under consideration as:

$$L(\mathbf{p}_{m,k}) = m_n \text{ for } \mathbf{p} \in S_n, k \in [1, K],$$

where the layers are constructed in a back to front order (i.e. starting with the smallest d_m). Note that the matching function $f(\mathbf{p}, m)$ may be extended to color images by summing the absolute differences in each of the three color channels. As in [15], we may also use the maximum of the absolute difference in each of the color channels instead of the sum.

This initial depth allocation is now used in a second pass to take into account occlusions. That is, we define the visibility function for each pixel in the images as:

$$V(\mathbf{p}, m, k) = \begin{cases} 1, & d_{L(\mathbf{p}_{m,k})} < d_m \text{ or if } L(\mathbf{p}_{m,k}) = m_n \text{ for } \mathbf{p} \in S_n \\ 0, & \text{otherwise.} \end{cases}$$

Therefore points that have a disparity gradient larger than the one being tested are occluded unless the point belongs to the layer under consideration since a constant depth layer cannot occlude itself. The matching function becomes:

$$f(\mathbf{p}, m) = \frac{\sum_{k=1}^{K-1} |I(\mathbf{p}_{m,k}) - I(\mathbf{p}_{m,k+1})| V(\mathbf{p}, m, k) V(\mathbf{p}, m, k+1)}{\sum_{k=1}^{K-1} V(\mathbf{p}, m, k) V(\mathbf{p}, m, k+1)}, \quad (3)$$

where the denominator is a normalization term. Occluded pixels will therefore be ignored in the second pass.

B. Rendering

Once the layers have been extracted, the interpolation of a new view is obtained through linear interpolation of each layer with a filter steered to match the layer's depth. It is important also to discard occluded pixels since these will cause blurring of the layer's boundary. In order to achieve this, we first generate a layer image for the view to interpolate in $t_i \in]t_k, t_{k+1}[$:

$$L(\mathbf{p}_{m,i}) = m_n \text{ for } \mathbf{p} \in S_n,$$

again in a back to front manner to take into account occlusions. Using linear interpolation, the values in the rendered image become

$$I(\mathbf{p}_{m,i}) = \begin{cases} \beta I(\mathbf{p}_{m,k}) + \alpha I(\mathbf{p}_{m,k+1}), L(\mathbf{p}_{m,k}) = L(\mathbf{p}_{m,k+1}) = m \\ I(\mathbf{p}_{m,k}), L(\mathbf{p}_{m,k}) = m, L(\mathbf{p}_{m,k+1}) \neq m \\ I(\mathbf{p}_{m,k+1}), L(\mathbf{p}_{m,k}) \neq m, L(\mathbf{p}_{m,k+1}) = m, \end{cases} \quad (4)$$

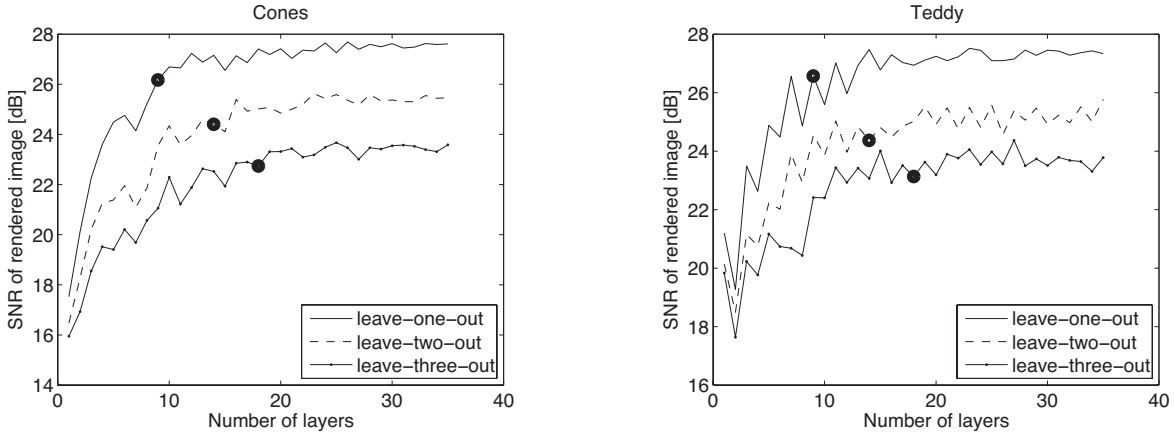
where $\alpha = t_i - t_k$ and $\beta = t_{k+1} - t_i$ are the weights from the linear interpolation. Therefore, if the point is visible in both neighboring images, the value in the rendered image is linearly interpolated with a filter that is skewed according to the disparity gradient d_m of the layer. However, if the point is only visible in one of the neighboring images, the value is taken only from the image in which it is visible. This distinction is not made in [5], [11], [12].

IV. Adaptive layer extraction

In this section, we show how the depth layer extraction can be made adaptive to the scene and the particular application. The adaptive part of the algorithm is based on (2). Assuming the camera parameters B and f are fixed, we are free to select the M and the Δt. For example, given a Δt, it is possible to determine the number of depth layers needed to render the scene without aliasing. According to the sampling theory in [5] the disparity space should be equally divided as

$$d_m = \frac{m - 0.5}{M} d_{max} + (1 - \frac{m - 0.5}{M}) d_{min}, \quad (5)$$

for $m = 1, \ldots, M$ and where d_{min} and d_{max} are the minimum and maximum disparity gradients. This range of possible d_m will then be fed to the layer extraction algorithm in Section III. Each patch will be assigned the disparity gradient d_m that minimizes the matching functional in (3). Note that for a smaller number of depth layers M, the depth estimation is faster since each patch only needs to be tested for a small number of hypothesized depths. This functional will also minimize the difference in intensity of the image points that are used for interpolation in (4) which minimizes blurring



(a) (b)

Fig. 2. Simulation results for the *Teddy* and *Cones* data sets. The SNR of rendered images versus the number of layers used in the layer extraction are shown for the leave-one-out, leave-two-out and leave-three-out cases. The bold points represent the minimum number of layers based on (6). From these plots, we see that adding more layers improves the rendering until minimum sampling is achieved. Following this point, there is no significant advantage to using more layers. We also notice that more layers are needed when more images are left out.

in the rendered images. The layer extraction complexity is linearly proportional to the number of depth layers chosen. Therefore there is a clear advantage to using the minimum number of layers.

We consider three different scenarios for adapting the layer extraction. First, we study the rendering results for different M. Second, we consider non-uniform $\Delta t = t_{k+1} - t_k$ and adaptively estimate the M in order to achieve minimum sampling. Finally, we consider the case where the number of layers is fixed and the Δt is adaptively chosen based on the scene observed.

A. Image quality versus number of layers

In some cases, rendering speed is essential perhaps at the expense of a reduction in the quality of the rendering. We may therefore choose fewer layers than are required by plenoptic sampling theory in order to speed up the interpolation. Note that in general, we should get an improvement in the quality of the rendered image by using more layers. After, a certain number of layers though, the anti-aliasing criterion is achieved and adding more layers gives no further improvement. An extensive study of the rendered images versus number of layers and number of images is presented in [5]. However, the experimental results are obtained with a known geometry. An important feature of the rendering system presented here is the ability to take advantage of the tradeoff between rendering quality and the number of layers.

B. Non-uniform image arrays

It happens in many cases that the sample images of the LF are not uniformly distributed. The M required is therefore not constant throughout the views. Indeed, the number of layers is given by:

$$\begin{aligned} M &= \Delta t B f h = \frac{(\Delta t f / z_{min} - \Delta t f / z_{max})}{2\Delta x} \\ &= \frac{\Delta t}{2}(d_{max} - d_{min}) \end{aligned}$$

where we have used $\Delta x = 1$ pixel. Therefore an estimate of the maximum and minimum disparities $\Delta t d_{max}$ and $\Delta t d_{min}$ enables automatic estimation of the number of depth layers needed in order to meet the minimum sampling criterion. In our current implementation the estimation of $\Delta t d_{max}$ and $\Delta t d_{min}$ is done using a fast block matching algorithm and a simple outlier rejection. The rendering system will therefore extract only the minimum M based on the estimates of the maximum and minimum disparities.

C. Variable minimum and maximum depths

The scenario might call for a fixed number of layers M to reduce depth estimation and interpolation complexity. We may therefore again use (2) to adaptively deduce the Δt necessary given an estimate of the h. For instance, consider the scenario where the camera is moving along a street and is pointed in the direction perpendicular to the movement. The camera may move quickly (i.e. a large Δt) if the scene observed is constrained to a small h and may be forced to move slowly (i.e. a small Δt) when the scene has a large h.

V. Experimental results

The adaptive layer extraction method presented in this paper has been tested on the benchmark Middlebury stereo vision data sets *Teddy* and *Cones*[2] [17]. These data sets both contain nine uniformly sampled multi-baseline stereo images.

In this first part of the analysis, we look at the quality of the rendered image in terms of signal-to-noise ratio (SNR) versus the number of depth layers used. In order to provide a comparison and obtain a ground truth, we perform a leave-one-out test. That is, some of the original images are removed and we use the rendering algorithm to recover an image that was left out. The SNR is then computed with respect to the ground truth. Figure 2 illustrates the results for both data sets. The number of depth layers goes from one to 35 and we use $K = 3$

[2]Available at: http://vision.middlebury.edu/stereo/.

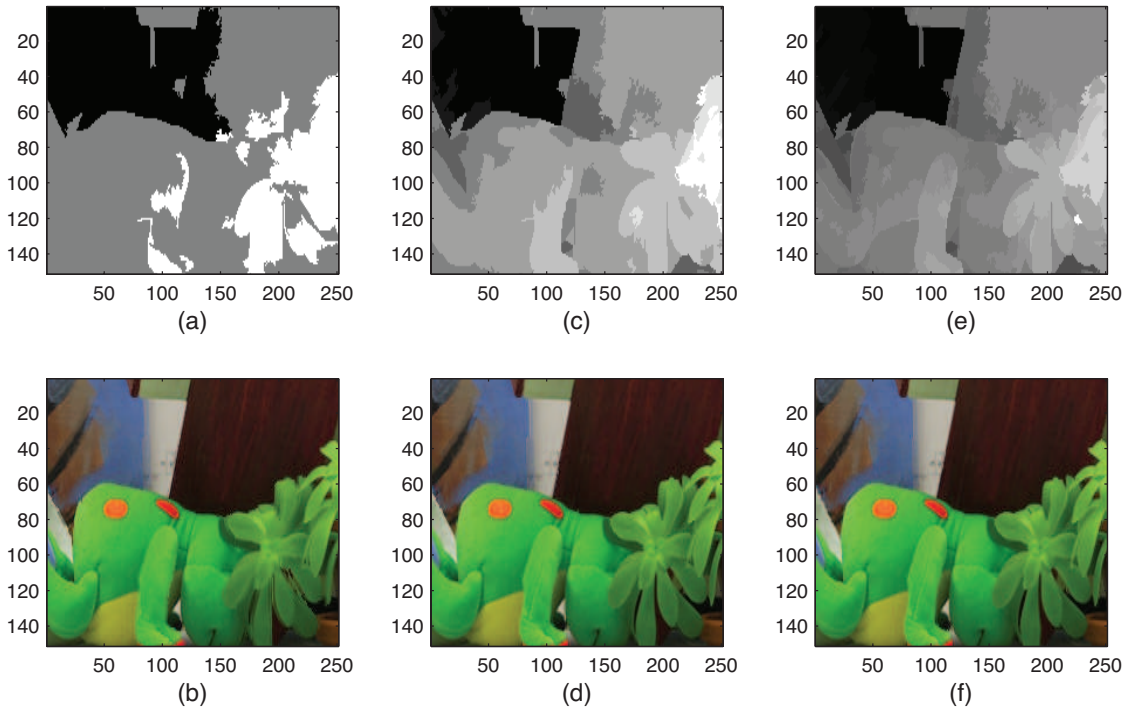


Fig. 3. Rendered images using the *Teddy* data set. (a-b) Layers and interpolated image using $M = 3$ layers (SNR 23.49 dB). (c-d) Layers and interpolated image using $M = 11$ layers (SNR 27.02 dB). (e-f) Layers and interpolated image using $M = 30$ layers (SNR 27.45 dB). Note the improvement in the quality of the rendered image between the three layer case and the 11 layer case. The 30 layer case, however, does not show a significant improvement over the 11 layer case.

images. According to the theory in [5], we should notice two points. First, the SNR of the rendered image increases with the number of layers. After a certain number of layers, the minimum sampling criterion is achieved and adding more layers does not significantly improve the result. Second, the increase in Δt (i.e. using only one out of two or three images) should require more layers in order to achieve the minimum sampling. Both these aspects are visible in Figure 2. The figure also shows in bold the minimum number of layers defined by (6). In practice, the M seems to be a bit conservative. This is due to the fact that the sampling theory does not take into account some effects such as the fattening of the spectrum due to occlusions. Note that the difference in SNR between consecutive choices for M are due to the fact that the depths in the scene are not uniformly distributed. It may be useful in some cases (e.g. scenes with only three depths) to use non-uniformly spaced d_m. Figure 3 illustrates an example of extracted layers and rendered images for different M. The layer extraction therefore behaves well with respect to the sampling theory. Note that the overall degradation in the SNR of the rendered images in the cases were the baseline is bigger is due to the fact that the layer extraction becomes a more difficult task.

For the second set of results, we feed to the rendering system only the images $(0, 1, 2, 4, 6)$ of the *Cones* LF (i.e. the images are not uniformly sampled). The number of layers is adaptively changed based on the method in Section IV-B. Figures 4(a-b) illustrate the layers and the rendered image in $t_i = 0.5$. In this case, the algorithm estimated $M = 5$. Figures 4(c-d) show the layers and the rendered image in $t_i = 3.0$. In this case, the baseline is doubled and the adaptive algorithm increases the number of layers to $M = 10$.

The EDISON implementation of the mean-shift segmentation was used.[3] The segment matching, layer extraction and rendering functions were implemented using a combination of Matlab and C++. In this setup, the segmentation times for the *Cones* images (375 by 450 pixels) are 2.38 seconds for the mean-shift segmentation of the reference image and 1.64 seconds for extracting the layers with $M = 5$ and $K = 3$. Once the layers have been extracted, the rendering time is 0.3 seconds per frame. Note that these times are given for the experimental setup and can be significantly reduced by using optimized code.

VI. Conclusion

Plenoptic sampling theory has shown that there is a clear tradeoff between the amount of geometry and the number of images available. Moreover, there is a minimum sampling criterion that gives the number of depth layers needed based on the spacing of the sample images and the maximum and minimum depths. In this paper, we presented a simple and effective layer extraction method that deals with occlusions and is designed for image based rendering (i.e. the cost function minimizes blurring). Finally, in contrast to previous work, the algorithm takes into account occlusions and automatically

[3] Available at: http://www.caip.rutgers.edu/riul/research/code.html.

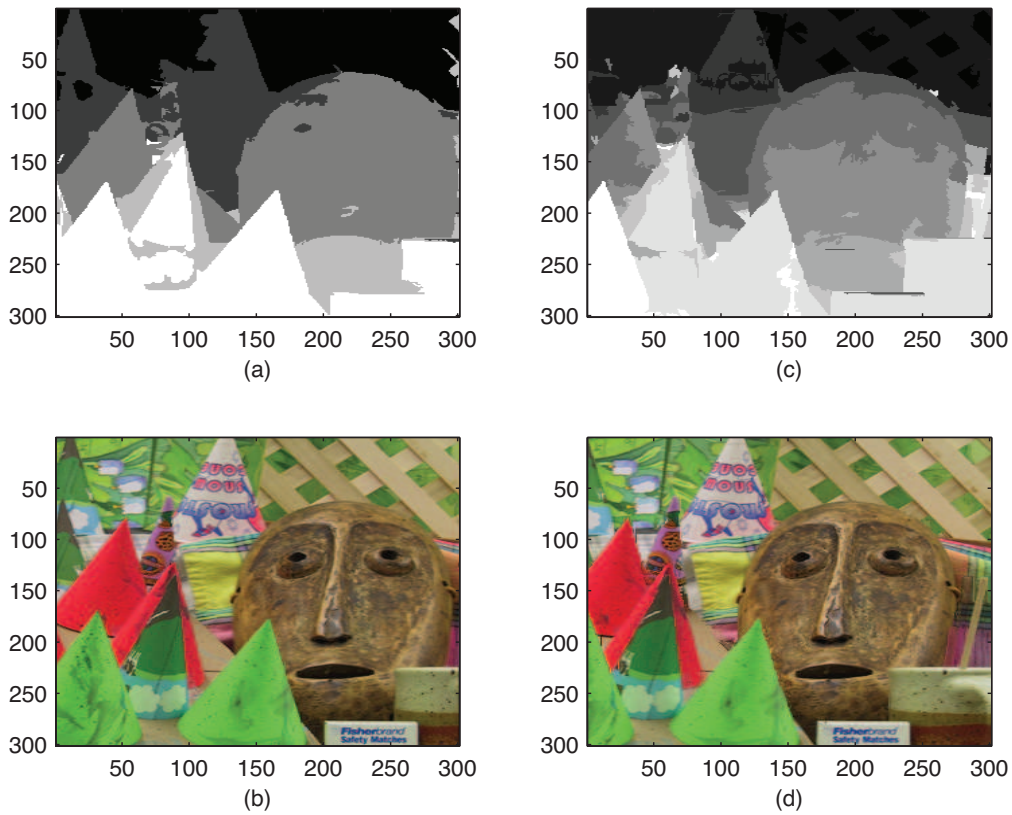


Fig. 4. Adaptive layer extraction on the *Cones* data set. Images $(0, 1, 2, 4, 6)$ are fed to the rendering algorithm. (a-b) When rendering an image in $t_i = 0.5$, the layer extraction is adapted to $M = 5$ layers. (c-d) Rendering the view in $t_i = 3.0$ requires more layers since the baseline is increased. Here, the layer extraction is adapted to $M = 10$ layers.

adapts the number of depth layers to extract based on the scene itself and the spacing between the sample views. In future work, we will extend these results to non-linear camera movements and interpolating viewpoints that are not on the camera path.

ACKNOWLEDGMENT

The work reported in this paper was funded by the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre established by the UK Ministry of Defence.

REFERENCES

[1] M. Levoy and P. Hanrahan, "Light field rendering," in *Computer graphics (SIGGRAPH '96)*, 1996, pp. 31–42.

[2] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Computer graphics (SIGGRAPH '96)*, 1996, pp. 43–54.

[3] C. Buehler, M. Bosse, L. McMillan, S. J. Gortler, and M. F. Cohen, "Unstructured lumigraph rendering," in *Computer graphics (SIGGRAPH '01)*, 2001, pp. 425–432.

[4] E. H. Adelson and J. R. Bergen, "The plenoptic function and the elements of early vision," in *Computational Models of Visual Processing*. MIT Press, Cambridge, MA, 1991, pp. 3–20.

[5] J.-X. Chai, S.-C. Chan, H.-Y. Shum, and X. Tong, "Plenoptic sampling," in *Computer graphics (SIGGRAPH '00)*, 2000, pp. 307–318.

[6] C. Zhang and T. Chen, "Spectral analysis for sampling image-based rendering data," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 13, no. 11, pp. 1038–1050, November 2003.

[7] J. Shade, S. Gortler, L. W. He, and R. Szeliski, "Layered depth images," in *Computer graphics (SIGGRAPH '98)*, 1998, pp. 231–242.

[8] C. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Computer graphics (SIGGRAPH '04)*, 2004, pp. 600–608.

[9] H. Y. Shum, J. Sun, S. Yamazaki, Y. Li, and C. K. Tang, "Pop-up light field: An interactive image-based modeling and rendering system," *ACM Trans. Graph.*, vol. 23, no. 2, pp. 143–162, April 2004.

[10] J. Berent and P. L. Dragotti, "Plenoptic manifolds," *IEEE Signal Processing magazine*, vol. 24, no. 7, pp. 34–44, November 2007.

[11] Y. Li, X. Tong, C. K. Tang, and H. Y. Shum, "Rendering driven depth reconstruction," in *IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, vol. 4, April 2003, pp. 780–783.

[12] K. Takahashi and T. Naemura, "Layered light-field rendering with focus measurement," *Signal Processing: Image Communication*, vol. 21, pp. 519–530, 2006.

[13] J. Berent and P. L. Dragotti, "Unsupervised extraction of coherent regions for image based rendering," in *British Machine Vision Conference (BMVC)*, vol. 1, September 2007, pp. 409–418.

[14] M. Bleyer and M. Gelautz, "A layered stereo algorithm using image segmentation and global visibility constraints," in *IEEE Int. Conf. on Image Processing*, 2004, pp. 2997–3000.

[15] Z.-F. Wang and Z.-G. Zheng, "A region based stereo matching algorithm using cooperative optimization," in *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2008, pp. 1–8.

[16] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE Tran. on Pattern Analysis and Machine Intelligence*, vol. 24, no. 5, pp. 603–619, May 2002.

[17] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. Journal of Computer Vision*, vol. 47, no. 1, pp. 7–42, 2002.