Plenoptic layer-based modelling for image based rendering

James Pearson*, *Student Member, IEEE,* Mike Brookes, *Member, IEEE,* and Pier Luigi Dragotti, *Senior Member, IEEE*

***Abstract*—Image based rendering is an attractive alternative to model based rendering for generating novel views due to its lower complexity and potential for photo-realistic results. In order to reduce the number of images necessary for alias-free rendering, some geometric information for the 3D scene is normally necessary. In this paper, we present a fast automatic layer-based method for synthesising an arbitrary new view of a scene from a set of existing views. Our algorithm takes advantage of the knowledge of the typical structure of multiview data in order to perform occlusion-aware layer extraction. Moreover, the number of depth layers used to approximate the geometry of the scene is chosen based on Plenoptic sampling theory with the layers placed non-uniformly to account for the scene distribution. The rendering is achieved by using a probabilistic interpolation approach and by extracting the depth layer information on a small number of key images. Numerical results demonstrate that the algorithm is fast and yet is only 0.25 dB away from the ideal performance achieved with the ground-truth knowledge of the 3D geometry of the scene of interest. This indicates that there are measurable benefits from following the predictions of Plenoptic theory and that they remain true when translated into a practical system for real world data.**

***Index Terms*—View synthesis, Plenoptic function, depth layer, multi-view.**

EDICS Category: ARS-SRV

I. Introduction

VIEW synthesis is the process of generating an arbitrary new view of a scene from a set of existing views. One approach is to create a textured 3D model of the entire scene and to use this for synthesising new views. This approach allows freedom in the final rendering but creating the complex 3D model in the first place can often be computationally intensive. Moreover, the synthesised output images, in particular for cluttered scenes, are often noticeably artificial. An alternative approach is image based rendering (IBR) [2], [3], in which new views are generated by combining individual pixels from a densely sampled set of input images. This approach requires little geometric information and can give potentially photo-realistic results but requires many more input images [4]. These two approaches can be thought of as opposite extremes of a spectrum where a reduction of one resource, geometric completeness, requires a corresponding increase in another, the number of images, to maintain a consistent quality.

The authors are with the Department of Electrical and Electronic Engineering, Imperial College, London, SW7 2AZ, UK .

e-mail: j.pearson09@imperial.ac.uk, mike.brookes@imperial.ac.uk, p.dragotti@imperial.ac.uk.

This work was in part presented at ICASSP 2011 [1].

Plenoptic sampling theory [5], [6] gives us a theoretical framework to understand this trade-off. In particular, Plenoptic sampling shows that, in the absence of occlusions, the number of views necessary for alias-free rendering does not depend on the geometrical complexity of the scene but only on the depth variation within the scene (e.g., [7]). Consequently, a layer-based representation [8], [9], [10], [11], where the scene is split into separate depth layers each with a reduced depth range, is a good way of introducing a variable amount of geometric complexity to allow accurate view synthesis from a moderate number of input images. In particular, the trade-off between geometric information and rendering quality reduces, in this way, to a trade-off between the number of images, the depth variation within the scene and the number of layers. A layer based model also has other advantages including implicit occlusion ordering and scalability.

For a layer based system each pixel in the input images needs to be assigned to a particular layer. This is normally achieved by matching points in two or more images and comparing the pixel position shift and the camera position shift which gives us the depth. Several methods have operated at a local pixel level, often with high speed (e.g., [12]) the accuracy of which can be improved by expanding the matching scope, for example by utilising a semi-global approach to improve the edge accuracy (e.g., [13]). One popular alternative to a pixel based method is the use of blocks of pixels [14]. Although more robust to noise and requiring a less iterative approach it introduces the problem of blockiness and poor reproduction of object edges. Various post-processing methods have been proposed to refine a coarse disparity map in conjunction with the original images [15], [16]. An alternative to dealing with the issue of matching object edges is through the use of a collection of sub-blocks processed together, as suggested in [17], [18], or using segments based on the image content rather than a regular grid [19]. Although an initial segmentation step is required and some assumptions are made about picking segments that remain within the same layer, there are several advantages to this approach as discussed by Zhang et al. [20]. One is a higher robustness to noise, another is that it allows good edges to be formed without a highly iterative approach. An extension of the general segmentation method is the use of high level object segmentation [21], [22] often based on human intervention [9].

There are many ways to use the layer model to synthesise new image views, various image warping techniques applied to the whole image have been proposed in [23], [24] although the resultant output is a complete image, it may be significantly

distorted and often may not fully take into account the occlusions and disocclusions inherent in the set-up. An alternative approach is a rigid layer shift accounting for the occlusion ordering on the layers, this models a scene more accurately but dissoclusion may lead to gaps in the final output which need to be filled as discussed in [25].

Depending on the type of rendering and the quality of the depth geometry a number of rendering artifacts can arise. Various approaches to mitigate these have been proposed such as enhancing depth geometry using the images [26] or merging separately generated geometry together [27]. One major, though inevitable, difficulty with layer shifts is the introduction of holes in the output image due to regions in the output image that are not visible in any input image. Several innovative approaches have been suggested to solve this for specific situations with varying degrees of complexity, for example [28], [29], [30]. Work has also been done to measure and predict the extent of errors in a system [31] to pick the particular approach to be used.

In this paper we present a fast automatic algorithm for IBR from a set of input images that uses Plenoptic sampling theory as a guide to the required number of layers. The theory shows that, provided certain assumptions are met, alias-free rendering can be achieved by spacing the layers uniformly in inverse depth. In practice however, these assumptions, which include the absence of occlusions, an infinite field of view and a perfect reconstruction filter, are not fully met and some aliasing is inevitable. In Sec. V-B, we demonstrate that this residual aliasing distortion can be reduced by placing the layers closer together than the minimum spacing predicted by Plenoptic theory. Conversely, for any given number of layers, the impact of the residual aliasing on rendering quality is affected by the chosen layer positions. Accordingly, our algorithm selects non-uniformly spaced layer positions according to the depth distribution of objects within the scene by increasing the density of layers at depths that occur frequently while reducing the density at depths that occur infrequently. In Sec. V-B we demonstrate on a range of datasets that despite the use of non-uniformly spaced layers, Plenoptic theory accurately predicts the minimum number of layers required for high quality rendering even though its underlying assumptions are not fully met in practice.

The algorithm handles occlusions effectively by assigning image regions to layers in two non-iterative stages. It further improves rendering quality by selectively merging small regions into surrounding layer regions and by the use of a probabilistic interpolation method. Moreover we propose a novel method of using multiple depth maps in a master-slave approach that is effective and scalable. We also note that the overall algorithm scales naturally with the number of input images, can be adaptive in the choice of the number of layers and can be used on different camera arrays such as the EPI volume [32] or the Lightfield [33], [34].

Simulation results show that our method is only 0.25 dB away from the ideal performance achieved when having access to the ground truth pixel based geometric information of the scene and comparisons are also made to alternative methods. In addition these results demonstrates the effectiveness of our method and the validity of the layer-based model.

We note that previously Tong et al. [35] have investigated the trade-off between geometry and the number of input images. There are several similarities in our approaches including the use of a layered geometric model and the combination of discrete input images to directly synthesise the output rather than using a pre-generated unified reference image model. However, they use a stereo-matching algorithm to extract layers whereas we use a two-stage approach which allows us to handle occlusions effectively. Moreover, they have investigated situations in which the trade-off between geometry and number of images can have several optimal points and experimentally determine their validity. In contrast, we have used only a single operating point, as given by Plenoptic sampling theory, based on a fixed input image spacing, and have investigated the behaviour either side of this operating point.

The paper is organised as follows: In Sec. II we discuss the Plenoptic function and its relation to the depth layer model. Sec. III describes how we extract the layers from the input images. In Sec. IV we present the view synthesis algorithm and in Sec. V we analyse the performance of our method. Finally, Sec. VI concludes the paper.

II. The Plenoptic Function and layer approximation

A convenient way of regarding a multiview image set is to consider the collection of light rays emanating from the scene. The complete seven dimensional parametrization of the rays at any position and time is known as the Plenoptic function, introduced by Adelson and Bergen [36]:

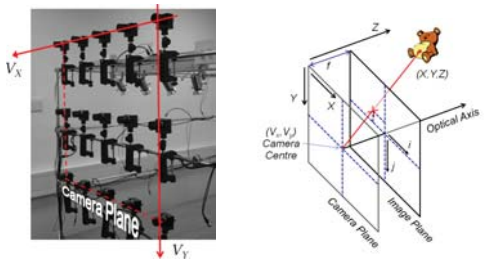
$$P = P_7(i, j, \lambda, t, V_X, V_Y, V_Z), \tag{1}$$

in which λ is the wavelength, t is the time, (V_X, V_Y, V_Z) is the position of the camera centre and (i, j) a point in the image. The dimensions of the Plenoptic function can be reduced by imposing restrictions on the acquisition setup. Thus we can omit t for a static scene and we can omit λ by considering separate red, green and blue images. A convenient parametrization, the Light Field or Lumigraph, introduced in [33], [34], assumes the light ray intensity is constant along its length, the cameras are restricted to the plane $V_Z = 0$ and define a light ray by the coordinates of its intersections with two parallel planes, the image plane (i, j) and the camera plane (V_X, V_Y). This leaves us with the four dimensional parametrisation,

$$P = P_4(i, j, V_X, V_Y). \tag{2}$$

In this paper, we will assume that a static scene is sampled by an array of identical pinhole cameras whose optical centres lie on a camera plane perpendicular to their optical axes as illustrated in Fig. 1(a). We define a right-handed world coordinate system with its origin at the optical centre of the upper left camera position and the Z-axis pointing towards the scene.

The geometry of the pinhole camera Lightfield is illustrated in Fig. 1(b). The camera centre location is (V_X, V_Y) on the camera plane which is separated from the image plane by the

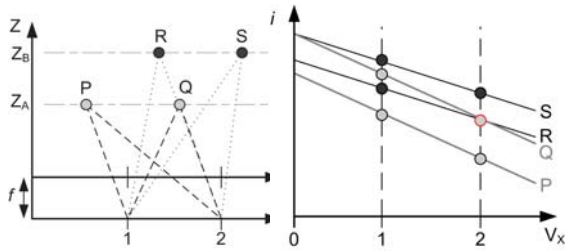


(a) Camera Array (b) Pinhole model

Fig. 1. (a) Our array of cameras allows us to sample the Plenoptic function in the image, (i, j), and camera, (V_X, V_Y), planes. (b) The pinhole camera model of how the rays within a scene are captured by a camera, with the lens modelled as a single point, and the ray vector described as the intersection with two planes.

focal length f. The image plane for each camera has a separate coordinate system (i, j), centred on the optical axis. For a light ray that originates at point (X, Y, Z) in real world space and passes through the camera position (V_X, V_Y), the intersection with the image plane (i, j) is given by,

$$(i, j) = \frac{f}{Z}(X - V_X, Y - V_Y). \tag{3}$$



(a) Layered view (b) EPI lines

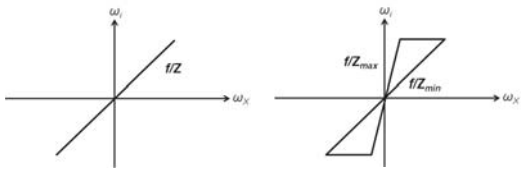
Fig. 2. Four points at two different depths, Z_A and Z_B observed by a camera in positions $V_X = 1$ and $V_X = 2$, (a) shows the top down real world scene and (b) shows the EPI plot.

The Plenoptic function can be further simplified by fixing V_Y, thereby restricting the camera positions to a horizontal line. This set-up is known as the 3-D EPI (Epipolar-plane Image) [32]. Fig. 2(a) shows the view from above of four points in a scene, P, Q, R and S at two different depths, Z_A and Z_B, from the camera line. The figure shows the light rays from the four points that are received at two different camera positions $V_X = 1, 2$. For each of the four scene points, Fig. 2(b) plots i,the intersection with the image plane of a light ray from the point as a function of the camera position, V_X. The locus corresponding to each scene point is known as its EPI line [32]. Each EPI line has a constant gradient, the 'disparity gradient' (DG) that is inversely proportional to the depth, Z, of its scene point; thus the lines corresponding to P and Q have a steeper gradient than those corresponding to R and S. From Fig. 2(a) we can see that when the camera is at $V_X = 2$, point Q occludes point R; this occlusion is predicted by the intersection of the EPI lines shown in Fig. 2(b) since lines with a steeper gradient occlude lines with a shallower gradient when they intersect. When we consider a full scene with many points and hence many EPI lines we call the whole an EPI line volume (ELV) [37].

A. Plenoptic Spectrum

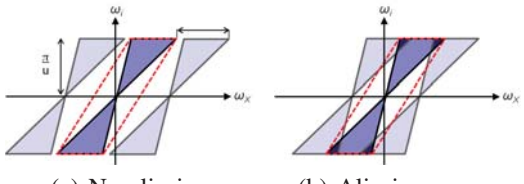
In [5], Chai et al. use spectral analysis to investigate the EPI structure described above. The two dimensional Fourier transform of a line in the EPI domain is a line perpendicular to the original and with a gradient f/Z. This is shown in Fig. 3(a) for a point at depth Z. In the more general case of a scene with varying depth, each point leads to a line in the EPI spectrum and all the line gradients are bounded by the minimum and maximum depths of points within the scene. For a scene comprising points with $Z_{min} \leq Z \leq Z_{max}$, we end up with a bandlimited spectrum with a characteristic bow-tie shape support as shown in Fig. 3(b). This theory makes several assumptions including, for example, the absence of occlusions and an infinite field of view. Since these assumptions are not fully met in practice, the spectrum of a real scene is only approximately bandlimited [7], [38], [39]. If the EPI is uniformly sampled with cameras spaced ΔV_X apart, the spectrum repeats at intervals of $2\pi/\Delta V_X$, as shown in Fig. 4(a), where u, the pixel spacing, determines the maximum frequency in the ω_i direction. An optimal reconstruction filter (dotted line) can be constructed around the fundamental section of the spectrum defined by Z_{max} and Z_{min}. This allows us to pick a sufficiently low camera spacing ΔV_X such that aliasing does not occur. If ΔV_X is made too large, aliasing will occur as the repeated spectra overlap; this is shown in Fig. 4(b). Combining the relationships shown in Fig. 3 and Fig. 4 we determine the maximum camera spacing [5] as,

$$\Delta V_X = \frac{1}{Bfh} \tag{4}$$



(a) Fourier transform (b) Bow-tie bounding

Fig. 3. (a) Shows the Fourier transform of an EPI line. (b) Taking the minimum, Z_{min}, and maximum, Z_{max}, depths bounds the bundle of EPI lines into a characteristic bow-tie shape.



(a) No aliasing (b) Aliasing occurs

Fig. 4. (a) Using an optimal reconstruction filter (dotted line) and a finite depth of field we can calculate a sufficiently small sampling spacing to avoid aliasing effects. (b) A higher ΔV_X leads to aliasing as parts of the repeated spectrum lie within the optimal reconstruction filter (shaded regions).

where $h = [1/Z_{min} - 1/Z_{max}]$ and $B \leq 0.5/u$ is the highest image bandwidth given a pixel spacing u.

B. Layer Model

The Plenoptic model describes a scene in terms of light rays emanating from points within a scene. A geometric model helps us describe and store the position of these points. One method of achieving this is a full 3D model in which every point has its own individually recorded position in (X, Y, Z). An alternative is a layer based model where the volume in which the points reside is partitioned into a set of constant-depth layers parallel to the camera plane and each point is assigned to the closest layer.

In this work we use a depth-layer based geometric model because it is robust, offers a good description of many real scenes and is computationally efficient. Fig. 5 shows the layer model of a simple scene, where each surface point is projected along the Z axis onto the nearest layer to form a series of fronto-parallel planes. Associated with each layer l, at depth Z_l, is a unique disparity gradient (DG), $g_l = \Delta i / \Delta V_X = \frac{f}{Z_l u}$, for a pixel spacing of u.

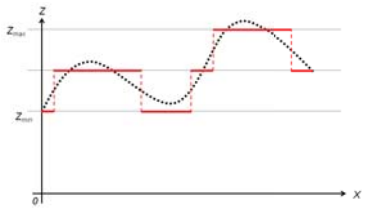


Fig. 5. Layer model, each point in the continuous real world (dotted) is projected onto the nearest layer to give a series of planes (solid).

By dividing the scene into layers, we can reduce the depth range within any given layer; this reduces h in (4) and therefore allows sparser sampling in V_X. Conversely if we have a fixed camera spacing, ΔV_X, we can use (4) to determine h. Assuming the layers are uniformly spaced in Z^{-1} with a pixel spacing of u, this allows us to determine the minimum number of layers,

$$L_{min} = f \Delta V_X B h \tag{5}$$

$$= \frac{f \Delta V_X}{2u} \left(\frac{1}{Z_{min}} - \frac{1}{Z_{max}} \right) \tag{6}$$

$$= \frac{\Delta V_X}{2} (g_{max} - g_{min}), \tag{7}$$

necessary for successful rendering, known as the minimum sampling criterion (MSC). This value gives us a guideline for the number of layers necessary for high quality rendering. Generally the range of Z for a scene will be constrained by the real world geometry, so if we are given a fixed camera spacing we can determine the optimal number of layers, or conversely if we have a fixed number of layers we can determine the corresponding maximum camera spacing.

We use this Plenoptic sampling framework to advise our layer extraction algorithm. In the initial stage of our algorithm we calculate Z_{min}, Z_{max} and ΔV_X in order to determine the necessary L_{min}. Since this computation can be performed on any number of input images, our algorithm allows us to adaptively modify the number of layers extracted as the visible scene depth range or camera spacing change.

III. Layer Extraction

Our view synthesis is dependent on the depth layer model generated from a collection of camera views. As discussed in Sec. II, the required number of layers can be determined from the Z_{min} and Z_{max} of the scene. This can be calculated from a sparse estimate of the scene geometry, which is used to initialise the next step. At this point we diverge from the Pleoptic theory which suggests evenly spaced layers, as objects within a real scene are not uniformly distributed in depth so there are advantages to assigning the output layers with uneven spacings. To do so a more detailed knowledge of the scene geometry is needed to assign layers to the best positions, this is discussed further in Sec. III-C. Once the layer positions have been chosen we can assign each pixel within an image to a particular layer, this flat representation of the geometry is known as a disparity gradient map (DG map). This gives us a final version of the ELV quantised to the chosen layers that we then use to synthesise new views. For the sake of convenience and to simplify explanation we have assumed the input images have already been rectified.

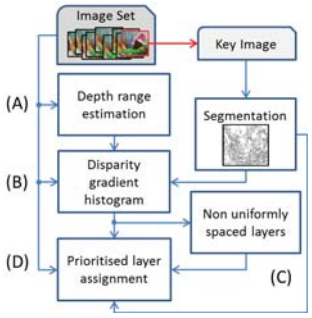


Fig. 6. Algorithm flow diagram, the main stages of the algorithm are (A) estimating the depth range of the scene (Sec. III-A), (B) calculate an accurate disparity gradient histogram (Sec. III-B), (C) assign the best layers using the Lloyd-Max algorithm (Sec. III-C) and (D) assign segments to layers (Sec. III-D).

The layer extraction algorithm as shown in Fig. 6 and described in detail below, comprises the following main stages:

i) *Depth range estimation*: Z_{min}, Z_{max} and ΔV_X are found by examining the depth estimation of features within the scene.
ii) *Disparity gradient histogram*: a more detailed estimate of the depth distribution of the scene, bounded by the previously calculated Z_{min} and Z_{max}, is obtained.
iii) *Non uniformly spaced layers*: the detailed depth distribution estimate from the previous step is used to determine the optimum layer positions that will minimise the total error.
iv) *Prioritised layer assignment*: pixels are assigned to layers in a single pass taking occlusions in the scene into account.

The algorithm can potentially compute a separate DG map for all the available input images, however, we found that for all the datasets tested, the DG map only needs to be calculated for a few key images, typically two images. The use and the choice of these key images will be covered in more detail in Sec. IV-C.

Finally, the algorithm outlined so far is for the EPI case where camera motion is only along the X axis, this is for ease of presentation and understanding. However the extension for the more general case of camera motion in more dimensions is straightforward. When our input is a 2-dimensional camera array we can parallelise the calculation along the V_X and V_Y axis, as shown in Fig. 7, and then combine the results. The

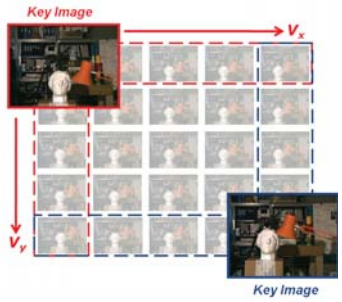


Fig. 7. With a 2-dimensional camera array the EPI sets for a key image can be separately calculated along both V_X and V_Y axis in parallel with a shared key image. Calculations along both separate axis can then be combined for a more robust and accurate result.

benefit of this approach is that we can also use the same algorithm for this type of camera array. Choosing two EPI subsets from the camera array that intersect makes it possible for them to share a common key image to ensure consistent segmentation. Additionally by choosing two perpendicular EPI sub-sets we maximise the assignment diversity and hence the coverage of the scene. For example in Fig. 7 if we select the top left camera view $V_X = 0, V_Y = 0$ as a key image we would choose the row $V_Y = 0$ for one EPI line and the column $V_X = 0$ for the other EPI line. This extension and how to utilise the extra dimension of information is covered in detail in Sec. III-E.

A. Depth range estimation

The first stage of the algorithm is to determine the Z_{min} and Z_{max} for the visible scene. To achieve this aim as efficiently as possible we match a limited number of distinctive features between two images.

FAST features [40] are extracted from the key image and matched to an adjacent image using the pyramidal Lucas-Kanade feature tracker [41], [42]. The implementation for both these algorithms is taken from the OpenCV 2.4 library [43].

If we compare the DG histograms, as shown in Fig. 8, for the FAST feature points (solid line) and the ground truth (dotted line) we can see they have a similar distribution, however there are regions which do not match well and there are several key discrepancies. The most obvious is the large peak between the DG values 3.8 - 4.2 which is only partially represented by a small spike in the FAST points at DG = 4, in addition the spike at DG = 9.4 is also missing. This is because, as can be seen in Fig. 9, the FAST points are not uniformly distributed in the image, as they cluster around distinctive features and are sparse in low texture regions in the background (DG values 3.8 - 4.0) and the roof area at DG = 9.4. Although there are not enough FAST points to determine the final layer positions robustly, we can reliably estimate Z_{min} and Z_{max} from this DG and move onto the next stage of the algorithm. By comparing the DG histograms we can easily calculate ΔV_X for different image pairs.

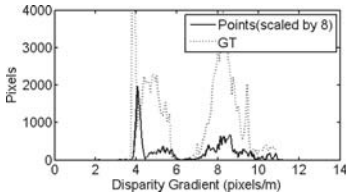


Fig. 8. Comparison of the DG histograms for image 0 from the Teddy sequence; the ground truth (dotted line) and the FAST features (solid line scaled by a factor of 8). Peaks in the ground truth histogram that correspond to regions with few FAST points (e.g. at disparity gradients 3.9 and 9.4) are missing from the FAST point histogram.

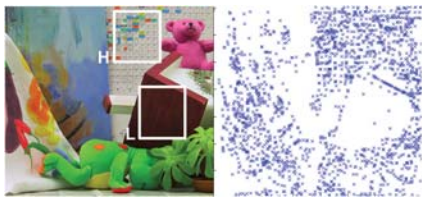


Fig. 9. Teddy image 0 and the corresponding FAST features. The features are not uniformly distributed, there are (H)igh concentrations of points within highly textured areas and (L)ow concentrations within regions having little texture variation.

B. Disparity gradient histogram

Matching the features between images gives a good estimate for the DG range but a more detailed estimate of the scene disparities is needed to assign layers. Although we want an estimate of the DG, g, for each pixel we will not calculate this on a pixel by pixel basis. Rather than assigning each pixel to a layer individually, we segment the images, using a 2D spatial and colour based method (eg. [44]), then assign entire segments to a particular layer. This has two advantages: it makes the algorithm more robust to noise and, since object edges are normally aligned to segment boundaries, results in sharp and consistent edges.

We need a reasonably accurate estimate of the g for each segment in the image. Based on our assumptions we can discount any features whose matches do not lie on the epipolar line, so shifts will only be along one axis. Most segments within the image have at least one feature, however not all feature tracking is reliable. To account for this we can compare the estimate from several features within the same segment and if they agree we can conclude that there is sufficient evidence to assign that segment to a particular g. However the more features we require, the fewer segments are available. We found experimentally that a valid threshold was 10 feature

points in a segment. The remaining segments are assigned using the following method.

We have an estimate for the Z_{min} and Z_{max} of the scene and hence their inverse relation g_{max} and g_{min}. We need to calculate the best match for each remaining, un-assigned segment within this DG range. For any given camera pair, with separation ΔV_X, we can calculate the expected disparity shift d of a segment with gradient g. We can evaluate the result of assigning a segment to a particular g and see how well the predicted shift of a segment from the key image applies as a prediction for the other images. We sample the DG histogram uniformly between g_{min} and g_{max}, with sufficient resolution to represent pixel-accurate disparities between the images of the most widely spaced cameras. Following [45] our matching metric is based on sum of absolute differences (SAD) where we are trying to minimise the error function ϵ for each of the N segments. Each segment, S_n, contains K_n pixels each of which has a position index $(i_k^{(n)}, j_k^{(n)})$ within an image I_m. Therefore the matching error ϵ is,

$$\epsilon\left(S_n\right) = \frac{1}{M}\sum_{k=0}^{K_n-1}\sum_{m=1}^{M-1}\left|I_0\left(i_k^{(n)}, j_k^{(n)}\right) - I_m\left(i_k^{(n)} + g^{(n)}V_m, j_k^{(n)}\right)\right| \tag{8}$$

where K_n is the total number of pixels within the segment S_n which is being evaluated over M images. I_0 is the key image and I_m is the target image. $g^{(n)}$ is the proposed DG and V_m is the V_x position of image m.

This allows us to select the layer assignments that will minimise the global ϵ for a scene.

C. Non uniformly spaced layers

Previous authors [45] have selected layers that are uniformly spaced in disparity as suggested by Plenoptic theory. For the case of a precisely bandlimited Plenoptic spectrum with an ideal reconstruction filter, this results in alias-free rendering with the minimum number of layers. Because the assumptions underlying Plenoptic theory are not fully met in practice, some aliasing is always present and its impact on rendered output images can be reduced by increasing the layer density beyond that indicated by the theory. As will be seen in Sec. V-B, the overall rendered image quality for a given number of layers can be improved by increasing the layer density at depths that occur frequently in the observed scene while decreasing it at depths that occur less often.

This assignment requires some geometric knowledge of the scene, so we take our new DG histogram, shown in Fig. 10 for the Teddy data set, and use it to assign the layers. We want to minimise the error from quantising disparities to these layer positions so the Lloyd-Max algorithm [46] with a quadratic cost function is used to find the values of g_l for each of the L_{min} layers. The DGs of the resulting layers are shown as the vertical lines in Fig. 10. It can be seen that these cluster around the regions with a higher density of pixels, minimising the assignment error when using the layer model.

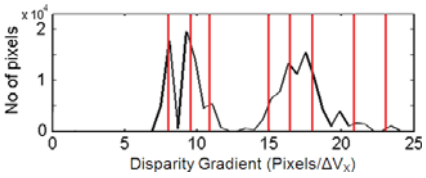


Fig. 10. Disparity gradient distribution (black curve) for Teddy sequence with its associated DG layers (vertical red lines), where L is 8.

If layers can be placed non-uniformly, the potential improvement in performance is several dB, as will be shown in Sec. V.

The use of non-uniform layer spacing represents a trade-off in which the aliasing error at frequently occurring scene depths is reduced at the expense of increased aliasing error at rarely occurring scene depths. This trade-off is controlled by the cost function used in the Lloyd-Max algorithm; we have found that the use of a quadratic cost function consistently gives the greatest improvement in PSNR on our evaluation datasets.

D. Prioritised layer assignment

We know from the Plenoptic theory that occlusions are hierarchical and predictable in that segments with higher g always occlude those with a lower g. Our key innovation is to refine the DG assignment in a separate step, initially analysing each segment in isolation (as discussed previously) and then taking into account the predicted occlusions from surrounding segments to refine the initial estimate. The improvements can be seen in Fig. 11.

(a) Original DG map. (b) Refined DG map.

Fig. 11. Using the prioritised segment assignment improves the accuracy of assignment for the whole DG map, especially for segments (marked) that are occluded by foreground objects.

Plenoptic sampling theory suggests that only a limited number of layers is required for alias-free synthesis, so we can conduct the final occlusion-aware segment assignment using the layers calculated with the Lloyd-Max algorithm (eg. 8 layers shown in Fig. 10) with no loss of quality.

The new matching error ϵ is calculated using,

$$\epsilon\left(S_n\right) =$$

$$\frac{1}{M}\frac{\sum_{k=0}^{K_n-1}\sum_{m=1}^{M-1} O_k^{(n)}\left|I_0\left(i_k^{(n)},j_k^{(n)}\right)-I_m\left(i_k^{(n)}+g^{(n)}V_m,j_k^{(n)}\right)\right|}{\left(\sum_{k=0}^{K_n-1}O_k^{(n)}\right)\log\left(\sum_{k=0}^{K_n-1}O_k^{(n)}\right)} \tag{9}$$

where O is a visibility mask and,

$$O_k^{(n)}=\begin{cases}1 & \text{if } I_m(i_k^{(n)}+g^{(n)}V_m,j_k^{(n)}) \text{ is visible;}\\ 0 & \text{if } I_m(i_k^{(n)}+g^{(n)}V_m,j_k^{(n)}) \text{ is occluded.}\end{cases} \tag{10}$$

As the segments were previously matched with no concept of occlusions the results were independent of the assignment order. However we can use the previous results to aid us in re-calculating the segment disparity in a more efficient manner. Assuming that the first pass is relatively accurate all S_n assigned to the top level, g_{max}, should be well assigned as they will not have any occlusions. If ϵ is sufficiently low then the top level segments are used to form the occlusion map for the subsequent layer. If ϵ is too high then it is likely that the segment has been misassigned and so it is omitted from the occlusion map. This process is repeated for each layer until g_{min} is reached. Segments with a poor matching score are ignored until the very end at which point they are then assigned using the most recent and complete occlusion map. The benefits of this prioritised procedure is that occlusions are estimated for all new assignments, rather than the less accurate assignments of Eq. (8), and that unreliably assigned segments are ignored when estimating occlusions. We note that the prioritised approach does not increase the complexity of the method in that it only changes the order in which segments are tested but it does improve the quality of the occlusion map and hence the final reliability of the algorithm. The weighting in the SAD (9) is biased towards preferring larger segments whenever possible, so the increased reliability of large segments is reflected in the confidence metric.

E. Segment extraction for 2D camera arrays

For the 2D camera array case the two intersecting camera lines are calculated separately and then combined afterwards. This combination is simple as the camera lines intersect with the shared key image at the intersection point, as seen in Fig.7. This means that only one image needs to be segmented and that the matching error for each segment can be minimised in both directions. By choosing to use an additional camera line perpendicular to the first we maximise the diversity of the segment matching as some objects may be largely occluded or contain poor texture in a certain direction but these problems might not be apparent in the orthogonal direction. For example in Fig. 12 we look at the matching confidence (inverse error) of a segment for different potential disparities and we note that the confidence along V_X (dashed line) shows a small peak while that along V_Y (solid line) shows a large distinct peak which is closer to the ground truth (labelled GT). We therefore find that the most robust and reliable improvement comes from choosing either one direction or the other based on the strength

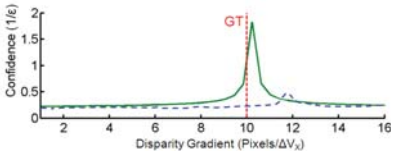
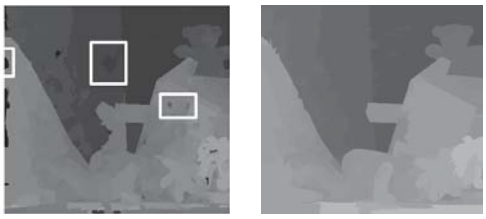


Fig. 12. For this segment there is a small (incorrect) peak when matching along V_X (dashed line) but along V_Y (solid line) there is a distinct peak in the segment assignment confidence close to the marked ground truth (GT).

and sharpness of the peak, rather than combining and possibly exacerbating any errors. As both EPI sub-sets have the same key image, combining the results is very simple.

F. Depth Flattening

The prioritised segment matching step is good for dealing with the types of errors shown in Fig. 11 where a segment is grossly mis-assigned due to an occlusion. Fig. 13 illustrates an example of a few types of error that are not resolved. Segments that are small and affected by frame occlusions or a segment wrongly assigned to a slightly different DG will cause a minor but unsightly artefact in the final synthesis. An additional step is required to deal with this type of issue. To reduce the



(a) Original DG map (b) Enhanced DG map

Fig. 13. Using the prioritised segment assignment and applying the flattening algorithm with an α of 0.4 and ζ of 0.01 per iteration allows us to deal with un-assigned and slightly miss-assigned segments.

artefacts in the final output we can include the DG of the surrounding layers as a weight in the segment assignment. This will favour adjacent segments with the same DG as long as there is not a major reduction in the matching confidence. The first step is to find the percentage of the segment border

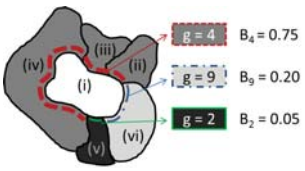


Fig. 14. For segment (i) there are three different adjacent disparity gradients, g =2 , g =4 and g =9. Segments (ii) to (iv) all have g = 4 and their combined border ratio with (i) is 0.75 so $B_4^{(n)}$ = 0.75 , similarly from segment (vi) $B_9^{(n)}$ = 0.20 and segment (v) $B_2^{(n)}$=0.05.

bounded by each of the g_l, giving us the border ratio $B_g^{(n)}$ for each segment n and disparity gradient g (see Fig. 14 for an illustrative example). This border ratio allows us to determine which would be the best disparity to re-assign the

segment to in terms of flattening the DG map. The second step is to determine the cost of such a disparity re-assignment. We do this by looking at the DG confidence histogram $H_g^{(n)}$ for each segment n. Initially each segment is assigned to the highest peak at $\hat{g}_n$ and the cost of re-assignement to a new g is the percentage shift in the confidence value in relation to $\hat{g}_n$. For a low texture background segment with a wide peak (which is the main type of segment to have slight variations of assignment), as shown in Fig. 15, a small shift in g leads to a small shift of H so there is little cost in the re-assignment. Conversely a highly textured foreground object with a sharply

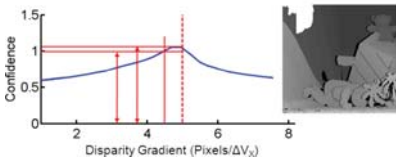


Fig. 15. When the peak is shallow and smooth, slight changes in g do not lead to a large change in confidence.

defined peak which we do not want to flatten with surrounding segments, such as Fig. 16, has a high cost for the same degree of re-assignment. Combining these gain and cost functions

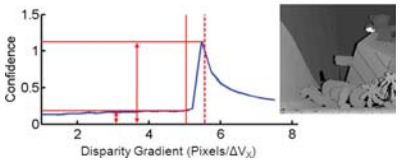


Fig. 16. When the peak is steep and sharp, slight changes in g lead to a large change in confidence.

together gives us the flattening metric $F_g^{(n)}$,

$$F_g^{(n)} = B_g^{(n)} - \alpha \left(\frac{H_{max}^{(n)} - H_g^{(n)}}{H_{max}^{(n)}} \right) \tag{11}$$

which balances the gain of flattening a segment to the surrounding segment DG, based on the border length, versus the cost of a less confidence assignment with a weighting term α to allow fine tuning. The segment, S_n, will be assigned to the layer associated with the highest $F_g^{(n)}$, as long as it is above a re-assignment threshold of 0.6. The process is iterative with all calculations occurring with the current segment assignments and a simultaneous re-assignment of all the segments after the round of calculations has finished. However in certain cases the segments can end up in periodic pattern, flip-flopping between a series of states. To force the system to stabilise a damping term ζ is added to the equation,

$$\tilde{F}_{g,n}(k+1) = B_g^{(n)}(k) - \zeta(k) - \alpha \left(\frac{H_{max}^{(n)} - H_g^{(n)}}{H_{max}^{(n)}} \right) \tag{12}$$

where a high ζ will stabilise the system in fewer iterations.

IV. View Synthesis

View synthesis is the creation of novel views of a scene based on existing images. Our synthesis algorithm consists of the following steps: First we need layer based geometry for all of the input images and the view to be synthesised. As described previously, we calculate the layer models for a few key images and then use these to predict the geometry for all the other views. This geometry allows us to use the EPI line structure to interpolate a new image from existing images.

The synthesis method for each new image comprises the following steps:

i) *Plenoptic synthesis*: using the EPI line structure we predict the intersection in adjacent images of the EPI line that passes through each new output image pixel, accounting for occlusions.
ii) *Probabilistic pixel interpolation*: based on their spatial positions and pixel value similarity a probabilistic estimate is made to interpolate the new pixel position.
iii) *Multiple key images*: multiple key images are utilised to fill in any gaps in the output image.
iv) *Orphan edges and alpha blending*: post-processing is applied to the image on a pixel by pixel basis to remove minor rendering artefacts.

Generally to minimize errors the closest two images either side of the new view are used for the synthesis, as described in Sec. IV-B.

Due to the differing amounts each layer is shifted, regions of one layer may move to occlude a layer with a lower DG. Consequently when the layers are shifted, regions of the scene also become disoccluded leaving gaps. It is important to understand the causes of different types of occlusion/disocclusion as different approaches are required to deal with them. Three types of possible disocclusion are illustrated in Fig. 17; (A) shows tearing, where a missing region appears in a oblique surface which is assigned to multiple depth layers; (B) shows a region of inter object disocclusion; these two types of occlusion are covered in Section IV-C. Type (C) errors demonstrate disocclusions due to the lack of available image information outside the field of view. Type (A) and (B) errors can be in-filled directly, either from surrounding pixels or different image sources if available. Type (C) holes can cause problems if in-filled directly, as described in the latter part of Sec. IV-C.

Fig. 17. The view from the Teddy sequence at $V_X = 0$ is projected layer by layer to $V_X = 8$, with resulting disocclusions left as black pixels. Three different types of disoclusion are highlighted.

A. Plenoptic synthesis

Earlier we described EPI lines for individual points, if we consider the EPI lines for the whole image sequence we will have a 3D volume of lines. Novel views are generated by interpolating new points from the other input images along the corresponding EPI line. We assume a Lambertian scene meaning that the intensity value is constant along each EPI line. In Fig. 18 we illustrate a simplified 2D case with four EPI lines on two layers. The new sample on an EPI line, at position $V_X = 1.7$, is interpolated from the samples provided by input images, $V_X = 1$ and $V_X = 2$, either side. For points P, Q and S the EPI line is un-occluded on both sides so the new sample will be interpolated as a blended distance-dependant mixture of the two input images. In the case of R only one side of the EPI line is un-occluded so only the sample from $V_X = 2$ will be used.

We synthesise the image on a layer by layer basis, starting with the lowest disparity and hence the most distant layer, and move through the layers progressively closer to the camera to preserve the occlusion ordering.

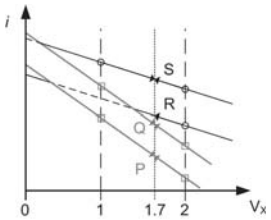


Fig. 18. To synthesise a new view at $V_{1.7}$ we take pixels along the EPI line from bracketing views V_1 and V_2 and combine them to form a new interpolated value. If a potential source pixel is occluded it is not included in the interpolation.

B. Probabilistic pixel interpolation

To synthesise a new view we scan through all the empty output pixels synthesising each individually by interpolating along the EPI lines from the two closest bracketing views, as shown in the top down view in Fig. 19. Because the $g \cdot$

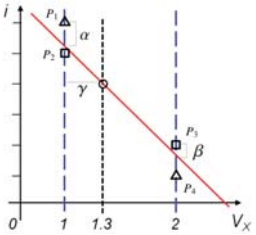


Fig. 19. When synthesising a new view (dotted line) at $V_{1.3}$ we interpolate along the EPI line using samples from bracketing views (dashed lines) V_1 and V_2. Because the sample point in i for the existing views does not always lie exactly on a pixel we have to use the two closest pixels from each bracketing view.

V_X for a point has a sub-pixel precision the projection to the bracketing images will not always lie exactly on a pixel. The traditional approach would be to linearly interpolate the value of the intersection point from the pixels either side of the intersection based on their spatial separation. For example using linear interpolation for the synthesised point in Fig. 19 we obtain

$$P_{1,2,3,4} = (1-\gamma)P_{1,2} + \gamma P_{3,4}, \tag{13}$$

where at $V_X = 1$,

$$P_{1,2} = (1-\alpha)P_1 + \alpha P_2, \tag{14}$$

similarly for $V_X = 2$ $P_{3,4}$ is calculated using β and

$$\gamma = \frac{V_s - \lfloor V_s \rfloor}{(\lceil V_s \rceil - \lfloor V_s \rfloor)}, \tag{15}$$

where γ is the distance between the synthesised image V_s (in the example shown in Fig. 19 $V_s = 1.3$) and the lower bracket image position, $\lfloor V_s \rfloor$ normalised relative by the total distance, $(\lceil V_s \rceil - \lfloor V_s \rfloor)$. α and β are the distances in pixels from the EPI line to P1 and P3 respectively.

However in some cases the pixels are not all equally valid as sample points, for example, we need to make sure our interpolation only uses pixels from the current layer and that we account for any potential error in our layer assignment. So rather than the fixed interpolation scheme of (13), we use a probabilistic method, weighting each input pixel based on its estimated reliability. The first stage is to set a very low weight to any of the four input pixels which are not on the same layer as the output pixel. Secondly we compare the pixel pairs on either side of the bracket, if they are close in value then they have a higher probability of being correct so they are weighted higher.

So the probabilistic prediction for the interpolated pixel now becomes,

$$\hat{P}_{1,2,3,4} = \frac{\hat{P}_{1,2} + \hat{P}_{3,4}}{\sum_{p=1}^{4} G_p}, \tag{16}$$

where

$$\hat{P}_{1,2} = (1-\gamma)(G_1\phi(1-\alpha)P_1 + G_2\theta\alpha P_2), \tag{17}$$

$$\hat{P}_{3,4} = (\gamma)(G_4\phi\beta P_4 + G_3\theta(1-\beta)P_3), \tag{18}$$

$$\theta = \frac{|P_1 - P_4|}{|P_1 - P_4| + |P_2 - P_3|}, \tag{19}$$

and

$$\phi = \frac{|P_2 - P_3|}{|P_1 - P_4| + |P_2 - P_3|}. \tag{20}$$

Where G_p is the weighting for a pixel, p, based on the distance of its g layer from the output pixel g layer $\mathrm{g_{diff}}$,

$$G_p = \begin{cases} 1 - |\mathrm{g_{diff}}| & \text{if } |\mathrm{g_{diff}}| \leq 0.5; \\ 0 & \text{otherwise.} \end{cases} \tag{21}$$

and α and β are ratios of the intersection distance of the EPI line in relation to the pixel pair either side, as shown in Fig. 19. The benefits of this approach are an improvement in PSNR and visual quality due to unreliable pixels having less effect on the interpolation.

C. Multiple key images

For complex scenes all regions of the scene may not be visible from a single key image. Using more key images increases the coverage of the scene and allows reliable assignment of these regions. For the EPI sequences tested, with between 5 and 9 images, we use two key images as we found that increasing the number of key images beyond this point provides little additional benefit to the output quality. By selecting images at opposite ends of the sequence we can increase the parallax and hence maximise coverage. A similar reasoning leads to choosing opposite corners when using a Lightfield source.

When using multiple key images it is important that all the calculated key image DG maps have the same layer positions. The DG histograms, Fig. 10, are estimated for each key image independently. These results are then combined before the Lloyd-Max algorithm is applied jointly to both in order to estimate a common set of layer disparity gradients. This allows easy and smooth combination of the key image DG maps as well as making sure that the layer positions are placed efficiently even for objects that might not be visible from some key images.

When synthesising a novel view, the consistent layer model used in all key images allows them to be used in a master-slave relationship. For each output view the closest key image is set as the master and any other available key images as slaves. Priority is given to the information from the master image in the case of any conflicts, so the slave key images are used only to fill holes in the resulting projection.

There are two different causes for the three types of occlusion shown in Fig. 17. Firstly, the types (A) and (B), are caused by objects occluding other objects within the scene, known as internal occlusions. As these occlusions are consistent within the scene they can be filled in from other slave images. Type (C) errors are more problematic, because these framing occlusions are not consistent within the scene as they will be unique for each image position, so they will therefore cause problems when they are projected beyond the camera position. For example Fig. 20 shows a few examples of continuous objects that are occluded by the image framing but would be visible as a continuous surface in other views.

Fig. 20. A few examples of contiguous regions within the scene that extend beyond the image framing and would therefore be occluded by the field of view.

Fig. 21(a) shows the DG map directly calculated for the camera position $(V_x, V_y) = (4, 4)$ from the Tsukuba sequence. Fig. 21(b) shows the prediction for the same camera position, based on the calculation for camera position $(V_x, V_y) = (0, 0)$. If we compare the two there are a number of errors.

In this case all the disocclusions are type B or C, as shown in Fig. 17. For type B disocclusions such as Fig. 21(b)(B) the error is a hole in the DG map so it can be in-filled either by interpolating from surrounding pixels, expanding the adjacent lowest disparity layer [47] or from the DG map of another key image if there is one available. Some type C disocclusions can be dealt with in a similar way, for example in the case of Fig. 21(b)(C-i), although the error is caused by framing rather than internal occlusion there is no information at all in the error area so it can be in-filled as previously described for a type B error. Fig. 21(b)(C-ii)) on the other hand poses a problem, although the region is missing part of the table lamp, due to the framing occlusion, there is already a lower layer present, that should be occluded, so no in-filling will occur. Because of their higher g layer, the foreground objects near the edge of the field of view are very vulnerable to this effect. Our method to prevent this is to project the slave DG

(a) $(V_x, V_y) = (4, 4)$ (b)Projected from $(V_x, V_y) = (0, 0)$

Fig. 21. Comparing the original DG map for (a) and the DG map for $(V_x, V_y) = (0, 0)$ projected to the same position shows that some regions (C-ii) cannot accurately be predicted without accounting for framing occlusion effects, whereas some can: (B), (C-i).

maps onto the master and record which regions fall outside the frame and hence correspond to regions unseen from the master map. An example of this is shown in Fig. 22, showing the regions of image $(V_x, V_y) = (4, 4)$ which are occluded by the framing of $(V_x, V_y) = (0, 0)$. These selected regions of the slave DG map can therefore legitimately occlude regions of the master map, if they have a higher g, which solves the problem caused by framing occlusions.

Fig. 22. Inter image projection allows us to calculate which parts of the slave key image are occluded by the master image frame.

D. Removing orphan edges and alpha blending

If the layer segmentation does not exactly match the underlying image then, as illustrated in Fig. 23, shifting a layer results in the edges of an object being left behind. These orphan edges are normally only a pixel or two wide but can cause very obvious rendering artefacts and can be distributed throughout the image (depending on the difference in disparity gradient on the object edge).

The orphan edges can be included in the correct layer if we pre-process the disparity map, enlarging each layer by

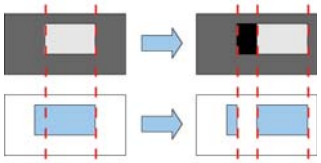


Fig. 23. If the layer segmentation (top layer) doesn't match the underlying image (bottom layer) then prediction projection results in the edges of an object being left behind.

extending the boundary outwards by two pixels. An additional benefit of this approach is that any small holes or thin intrusion into layers are also absorbed, which generally improves the modelling of a typical scene.

Layer extension solves the problems caused by orphan edges but introduces a different error, if the extension goes beyond the true layer boundary it leads to a halo of pixels round a foreground object that should be assigned to a lower layer causing an unsightly visual artefact.

As these errors will be on the edges of layers rather than distributed through the image they are easier to predict, additionally they are much easier to deal with via a technique called alpha blending or coherence matting [9]. We allow a degree of transparency for each pixel in the layer between 0 (completely transparent) and 1 (completely opaque). We model the layers separately so for each pixel we can sum up all the pixels in proportion to their alpha transparency. If all pixels had a transparency of 0.8, a pixel would consist of 80% the top layer then 16% of the next layer (0.8 times the remaining 0.2) and the remaining 4% from the final background layer. If there are no layers underneath the alpha transparency of a layer pixel will always be 1. Alpha blending mitigates the haloing effect and has the added benefit of smoothing any jagged layer edges.

It is important that the blending is done with true in-line blending rather than just blurring the edges to avoid adding unwanted inaccuracies and artefacts. The first stage is to generate a alpha blending map for each layer. We use a linear blending profile,

$$A_{i,j,g}^{blend} = \begin{cases} \dfrac{\mathrm{p_l}}{\mathrm{p_{max}}+1} & \text{if } p_l \leq p_{max}; \\ 1 & \text{otherwise.} \end{cases} \quad (22)$$

where p_{max} is the number of extended pixels and p_l is the current distance from the closest edge (in pixels) of position (i,j) on the appropriate g_l. If the underlying layer is not explicitly known it is interpolated from the surrounding geometry.

This blending layer is used to calculate an alternative for the pixel in question which is then blended with the top level pixel value. Fig. 24 shows the improvements using this combined extend/blend method. Orphan edges are removed and the edges of the foreground object are smoother and more natural looking without any loss of clarity or sharpness for the rest of the image.

V. Evaluation

For our evaluation we used the Teddy (450×375, 9 RGB images), Cones (450×375, 9 RGB images), Barn1 (432×381, 7 RGB images) and Sawtooth (434×380, 7 RGB images) datasets [48], [49]. The key images were segmented using the mean shift algorithm [50], [44]. We used the 'leave m out' method of evaluation in which only every $(m+1)^{th}$ image is included in the input image set. These are used to synthesize one of the omitted images for which the ground truth is known. In all cases an infilling algorithm was used to fill any holes with the lowest adjacent disparity in a similar way to [47].

(a) Original (b) Enhanced (c) Diff

Fig. 24. By extending the d map by 2 pixels and then alpha blending by the same amount the orphan edge effects seen in (a) can be removed (b). The orphan edges can clearly be seen in the exaggerated diff map (c).

A. Validation of the layer model

Plenoptic theory suggests that by choosing the appropriate number of layers we can have alias-free rendering, and that no further improvement will be gained by adding extra geometry. We validate this analysis and the effectiveness of our algorithm in Fig. 25 which shows the variation of PSNR with the number of layers averaged over all the evaluation datasets. This demonstrates that the gap between our algorithm and rendering based on the knowledge of the GT geometry is only 0.25 dB. It also shows that the layer-based representation incurs no-loss in performance when compared to the rendering based on complete geometry.

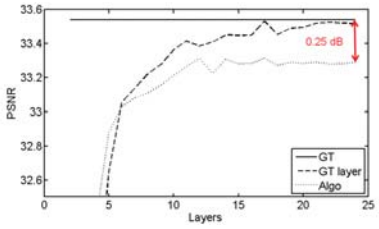


Fig. 25. The horizontal line shows the best average possible performance using the raw ground truth DG map. The dashed line shows the average effect of applying the layer model to the raw ground truth (with no segmentation). The dotted line is our average algorithm result when the layer model is applied to our own calculated DG map (with segmentation). All three results are obtained by averaging over all the datasets. The average L_{min} based on the MSC is 14, it can be seen that the results have plateaued by this point.

Specifically, the solid horizontal line represents the best possible rendering result using the provided raw ground truth (GT) DG map, which provides full and accurate pixel based geometric information. The dashed line shows the effect of applying the layer model to this data by calculating the best layer positions and assigning all the pixels to the closest layer. As the number of layers used increases so does the quality of the output until the improvement plateaus with no further improvement with additional layers. Importantly this plateau point is indistinguishable from the raw GT result showing that there is no inherent loss in quality if a sufficient number of

TABLE I
THIS TABLE CONTAINS THE COMPARISON RESULTS BETWEEN OUR 1ST STAGE ALGORITHM, AS DESCRIBED IN SEC. III-B, AN ALTERNATIVE STEREO-MATCHING METHOD [51] AND THE RESULT OF APPLYING OUR 2ND STAGE ALGORITHM WITH AND WITHOUT VARIABLY SPACED LAYERS USING LLOYD-MAX. ALL RESULTS ARE FROM THE TEDDY DATASET [48] WITH THE SAME PARAMETERS AND FINAL RENDERING ALGORITHM.

Method	PSNR (dB)
1st stage only (48 Layers)	32.43
Alternative method [51]	32.65
Alternative method [51] + 2nd stage (14 layers)	33.04
1st + 2nd Stage (48 layers)	33.20
1st + 2nd Stage (14 layers)	33.25

layers is used. Finally the dotted line shows the result of our layer based DG extraction and rendering algorithm, which has only a 0.25 dB drop from the best possible performance. Part of this drop is due to the use of segments and the remainder due to minor assignment errors.

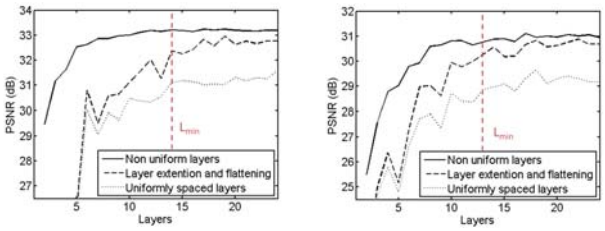
Table I includes the results obtained when using an alternative pixel-based algorithm [51], [52] for which code was available. The stereo-matching performance of this algorithm on standard test sets is very high (94.5% of pixels within ± 0.5 pixel disparity error [48]) although slightly worse that the current state-of-the-art (98% within ± 0.5 pixel disparity error). Using only the 1st stage of our algorithm from Sec. III-A (row 1) results in a lower performance than this alternative algorithm (row 2), primarily because of a small number of wrongly assigned segments. Applying the 2nd stage of our algorithm from Sec. III-D improves the performance of both the alternative method (row 3) and our method (row 5). The disparity gradient histogram is here generated using either [51] or our 1st stage method, the layers are assigned using the Lloyd-Max algorithm from Sec. III-C and the number of layers is 14 as indicated by the minimum sampling criterion, L_{min}, from (7).

Although the raw performance of our 1st stage method is worse than that of [51], its disparity gradient estimates have a lower median error; this results in more accurate layer depth values and a slight increase in overall performance when the 2nd stage of our algorithm is applied (row 5 versus row 3). Row 4 of the table shows the results of using the full depth resolution (48 layers) in both stages of our algorithm. We note that not only does this require much more computation, but the performance is actually slightly degraded by 0.05 dB.

B. Algorithm breakdown

There are several major separable elements to the algorithm, the breakdown of the geometric calculation is shown in Fig. 26(a) for the Teddy sequence. With uniformly spaced layers (dotted line) the performance improves slowly with the number of layers and a very large number is required to reach the performance limit. The PSNR can be increased (dashed line) by incorporating layer extension (Sec. IV-D), and disparity gradient flattening (Sec. III-F). With these improvements, the use of uniform layer spacing (dashed line) comes close to its limiting performance when using the number of layers, L_{min}, predicted by Plenoptic theory and shown in Fig. 26(a) as the vertical dashed line at L_{min} =14. As noted in Sec. III-C, the assumptions of Plenoptic theory are not fully met in practice and increasing the number of layers beyond L_{min} gives an additional performance improvement when using uniformly spaced layers. By using non-uniform layer spacing in our algorithm (solid line), we fully reach limiting performance with L_{min} layers and obtain significant performance improvement when using fewer layers than this.

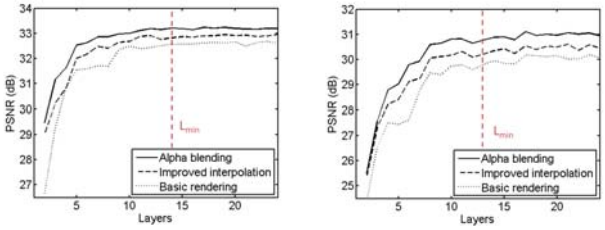
The corresponding graph for the Cones sequence is shown in Fig. 26(b) where we see that the relationship between the three curves is very similar. The use of non-uniform layer spacing again provides a clear benefit although its magnitude is less than with the Teddy sequence because the objects in the Cones sequence are more uniformly spread in depth. We note that L_{min} again indicates the number of layers required to reach limiting performance.



(a) Teddy L_{min} = 14 (b) Cones L_{min} = 13

Fig. 26. Showing the improvements in the algorithm results by using uniformly spaced layers (dotted), uniformly spaced layers with extension and layer flattening (dashed) and finally the best layer model with all enhancements and non-uniformly spaced layers. Results are for the Teddy sequence. The vertical line shows the calculated L_{min}.

We can also breakdown the improvements in the results due to various elements within the synthesis, as shown in Fig. 27(a) for the Teddy sequence. The basic rending method (dotted), with fixed pixel interpolation and no post-processing, can be improved by using probabilistic interpolation (dashed line), as described in Sec. IV-B. As well as smoothing the results it gives a dramatic improvement to the overall quality especially when few layers are used. Further improvements can be made across the board by using alpha blending (solid line) to minimise the errors on the edges of layers (see Sec. IV-D). Very similar effects may be seen in Fig. 27(b) for the Cones sequence although the differences are slightly increased.



(a) Teddy L_{min} = 14 (b) Cones L_{min} = 13

Fig. 27. Rendering improvements broken down into the basic rendering (dotted), improved interpolation (dashed) and the final alpha blended rendering (solid). Results are for the Teddy sequence. The vertical line shows the calculated L_{min}.

Finally we note that on a desktop PC the total time to read in the input frames, extract the layers and synthesise an

output image is 2.8 seconds, 0.6 seconds of which is the third party segmentation algorithm and 0.2 seconds is the time to synthesise each output image.

C. Output examples

In Fig. 28(a) we can see an example output of the algorithm from the Teddy sequence. With a PSNR of 33.9 dB and no major visual artefacts the rendering quality is very high with a definite photo-realistic feel. Looking at the luminance error map, Fig. 28(b), for the image we can see that 86% of the image has an error of one or less, the overall mean error is 1.004 (for a full scale of 255) and that the larger errors are only to be found on the edges of segments in thin bands. These edge errors are reduced due to the layer extension and alpha blending.

(a) Output (b) Error

Fig. 28. In (a) is an example rendered "miss one out" output for $V_X = 1$ from the Teddy sequence with a PSNR of 33.9 dB, with 18 layers. In (b) is an exaggerated difference error map (error $\times$ 10) for the image, with an average error of 1.004.

Our algorithm scales in multiple dimensions, the extra dimensions in for example a Lightfield sequence such as Tsukuba, Fig. 7, provide extra information that we can use to improve the layer allocation (see Sec. III-E). Additionally the extra dimensions allows us more degrees of freedom in moving the camera and synthesising new images without missing information for large regions of the output. Fig. 29 shows the results of changing V_Z for the output image, with increasing V_Z from left to right. Note that this is not a zoom but rather a true movement into the scene with resulting occlusions by foreground objects. The important change is that the layers are no longer rigid as movement of the camera in V_Z translates into movement in (i, j) for a point based both on its DG value and on its position within the image. The layer and position dependent scaling and warping can clearly be seen in the different relative sizes of objects within the scene as you move from left to right, foreground objects drastically change size while the background is largely unaffected. It should be noted that even with a large amount of movement into the scene the output quality is still maintained.

Fig. 29. These images show the results of moving the position of the output viewpoint in V_Z as well as V_X or V_Y. V_Z increases left to right.

VI. Conclusion

In this paper we have presented a novel layer based algorithm for IBR. Our approach uses Plenoptic sampling theory to infer the amount of geometric information required for artefact-free rendering. Guided by this prediction it takes advantage of the typical structure of multiview data in order to perform a fast occlusion-aware non-uniformly spaced layer extraction. The rendering is improved by using a probabilistic interpolation approach and by an effective use of key images in a scalable master-slave configuration. Numerical results demonstrate that the algorithm is fast and yet is only 0.25 dB away from the ideal performance achieved with the ground-truth knowledge of the 3D geometry of the scene of interest.

Finally we have shown that the Plenoptic theoretical framework is applicable to real world cases, that a layer based model does not lead to any loss in output quality and that the number of layers required is correctly predicted by the theory.

References

[1] J. Pearson, P.-L. Dragotti, and M. Brookes, "Accurate non-iterative depth layer extraction algorithm for image based rendering," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, May 2011, pp. 901–904.

[2] H.-Y. Shum, S.-C. Chan, and S. B. Kang, *Image Based Rendering*. Springer, 2007.

[3] C. Zhang and T. Chen, "A survey on image-based rendering – representation, sampling and compression," *Signal Processing: Image Communication*, vol. 19, no. 1, pp. 1–28, 2004.

[4] M. Tanimoto, "FTV: Free-viewpoint television," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 555–570, 2012.

[5] J.-X. Chai, X. Tong, S.-C. Chan, and H.-Y. Shum, "Plenoptic sampling," in *Int. Conf. on Comp. Graphics and Interactive Techniques*. ACM Press, 2000, pp. 307–318.

[6] L. McMillan and G. Bishop, "Plenoptic modeling: An image-based rendering system," in *Proc. SIGGRAPH*, Los Angeles, 1995, pp. 39–46.

[7] M. Do, D. Marchand-Maillet, and M. Vetterli, "On the bandwidth of the plenoptic function," *IEEE Trans. Image Proc.*, vol. 21, no. 2, pp. 708–717, Feb. 2012.

[8] J. Shade, S. Gortler, L.-w. He, and R. Szeliski, "Layered depth images," in *Proc. SIGGRAPH*, New York, 1998, pp. 231–242.

[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, 2004.

[10] Y. Li, X. Tong, C.-K. Tang, and H.-Y. Shum, "Rendering driven depth reconstruction," in *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing*, vol. 4, Apr. 2003, p. 780.

[11] J. Berent and P. L. Dragotti, "Plenoptic manifolds," *IEEE Signal Processing Mag.*, vol. 24, no. 6, pp. 34–44, 2007.

[12] K. Li, S. Wang, M. Yuan, and N. Chen, "Scale invariant control points based stereo matching for dynamic programming," in *Proc. Intl. Conf. Elec. Meas. Inst.*, 2009, pp. 769–774.

[13] H. Hirschmuller, "Accurate and efficient stereo processing by semi-global matching and mutual information," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 2, Jun. 2005, pp. 807–814 vol. 2.

[14] ——, "Improvements in real-time correlation-based stereo vision," in *Proc. IEEE Stereo and Multi-Baseline Vision*, 2001, pp. 141–148.

[15] O. Gangwal and R.-P. Berretty, "Depth map post-processing for 3D-TV," in *Int. Conf. Consumer Electronics*, Jan. 2009, pp. 1–2.

[16] S.-Y. Kim, E.-K. Lee, and Y.-S. Ho, "Generation of ROI enhanced depth maps using stereoscopic cameras and a depth camera," *IEEE Trans. on Broadcasting*, vol. 54, no. 4, pp. 732–740, Dec. 2008.

[17] S. Ince, E. Martinian, S. Yea, and A. Vetro, "Depth estimation for view synthesis in multiview video coding," in *Proc. 3DTV Conference*, May 2007, pp. 1–4.

[18] H. Hirschmuller, "Real-time correlation-based stereo vision with reduced border errors," in *Proc. Journal Comp. Vision*, vol. 47, no. 1, 2002, pp. 229–247.

[19] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. SIGGRAPH*. New York, NY, USA: ACM, 2004, pp. 600–608.

[20] G. Zhang, J. Jia, T.-T. Wong, and H. Bao, "Consistent depth maps recovery from a video sequence," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 31, no. 6, pp. 974–988, 2009.

[21] S. Chan, H.-Y. Shum, and K.-T. Ng, "Image-based rendering and synthesis," *IEEE Signal Processing Magazine*, vol. 24, no. 6, pp. 22–33, Nov. 2007.

[22] S.-C. Chan, Z.-F. Gan, K.-T. Ng, K.-L. Ho, and H.-Y. Shum, "An object-based approach to image/video-based synthesis and processing for 3-D and multiview televisions," *IEEE Trans. on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 821–831, 2009.

[23] Y. Mori, N. Fukushima, T. Yendo, T. Fujii, and M. Tanimoto, "View generation with 3D warping using depth information for FTV," *Signal Processing: Image Communication*, vol. 24, pp. 65–72, 2009.

[24] I. Daribo, C. Tillier, and B. Pesquet-Popescu, "Distance dependent depth filtering in 3D warping for 3DTV," in *Proc. IEEE Workshop on Multimedia Signal Processing*, Oct. 2007, pp. 312–315.

[25] M. Do, Q. Nguyen, H. Nguyen, D. Kubacki, and S. Patel, "Immersive visual communication," *IEEE Signal Processing Magazine*, vol. 28, no. 1, pp. 58–66, Jan. 2011.

[26] D. Min, J. Lu, and M. Do, "Depth video enhancement based on weighted mode filtering," *IEEE Trans. Image Processing*, vol. 21, no. 3, pp. 1176–1190, 2012.

[27] K. Takahashi, "Theoretical analysis of view interpolation with inaccurate depth information," *IEEE Trans. Image Processing*, vol. 21, no. 2, pp. 718–732, Feb. 2012.

[28] S.-T. Na, K.-J. Oh, and Y.-S. Ho, "Joint coding of multi-view video and corresponding depth map," in *Proc. Intl. Conf. Image Processing*, vol. 15, Oct. 2008, pp. 2468–2471.

[29] K.-J. Oh, S. Yea, and Y.-S. Ho, "Hole filling method using depth based in-painting for view synthesis in free viewpoint television and 3-d video," in *Proc. Picture Coding Symposium*, May 2009, pp. 1–4.

[30] S. Yamashita, N. Katoh, Y. Sasaki, Y. Akita, H. Chikata, and K. Yano, "Hole filling: a novel delay reduction technique using selector logic," in *Proc. IEEE Custom Integrated Circuits Conference*, May 1998, pp. 291–294.

[31] H. Nguyen and M. Do, "Error analysis for image-based rendering with depth information," *IEEE Trans. Image Proc.*, vol. 18, no. 4, pp. 703 –716, 2009.

[32] R. C. Bolles, H. H. Baker, David, and H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," in *Int. Journal of Computer Vision*, 1987, pp. 1–7.

[33] M. Levoy and P. Hanrahan, "Light field rendering," in *Proc. SIGGRAPH*, New York, 1996, pp. 31–42.

[34] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proc. SIGGRAPH*, New York, 1996, pp. 43–54.

[35] X. Tong, J. Chai, and H.-Y. Shum, "Layered lumigraph with lod control," *The Journal of Visualization and Computer Animation*, vol. 13, no. 4, pp. 249–261, 2002. [Online]. Available: http://dx.doi.org/10.1002/vis.293

[36] E. H. Adelson and J. Bergen, "The plenoptic function and the elements of early vision," *Computational Models of Visual Processing*, pp. 3–20, 1991.

[37] A. Criminisi, S. B. Kang, R. Swaminathan, R. Szeliski, and P. Anandan, "Extracting layers and analyzing their specular properties using epipolar-plane-image analysis," *Comp. Vis. Image Underst.*, vol. 97, no. 1, pp. 51–85, Jan. 2005.

[38] C. Gilliam, P. L. Dragotti, and M. Brookes, "A closed-form expression for the bandwidth of the plenoptic function under finite field of view constraints," in *Proc. Intl. Conf. Image Processing*, Hong Kong, Sep. 2010, pp. 3965–3968.

[39] C. Gilliam, M. Brookes, and P. L. Dragotti, "Image-based rendering and the sampling of the plenoptic function," in *Emerging Technologies for 3D Video: Creation, Coding, Transmission and Rendering*, F. Dufaux, B. Pesquet-Popescu, and M. Cagnazzo, Eds. Wiley, 2013, ch. 12, pp. 231–248.

[40] E. Rosten and T. Drummond, "Machine learning for high-speed corner detection," in *Proc. European Conf. Computer Vision*, 2006, pp. 430–443.

[41] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. Imaging Understanding Workshop*, 1981, pp. 121–130.

[42] J. Bouguet. (2000) Opencv. Intel Corporation. [Online]. Available: http://opencv.org/

[43] G. Bradski, "The OpenCV Library," *Dr. Dobb's Journal of Software Tools*, 2000.

[44] C. Christoudias, B. Georgescu, and P. Meer, "Synergism in low level vision," in *Proc. Int. Conf. on Pattern Recognition*, vol. 4, 2002, pp. 150–155.

[45] H. Hirschmuller and D. Scharstein, "Evaluation of stereo matching costs on images with radiometric differences," *IEEE Trans. Pattern Anal and Machine Intell*, vol. 31, no. 9, pp. 1582–1599, 2009.

[46] S. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Information Theory*, vol. 28, no. 2, pp. 129–137, Mar. 1982.

[47] S. Zinger, L. Do, and P. de With, "Free-viewpoint depth image based rendering," *Journal of Vis. Com. and Img. Rep.*, vol. 21, no. 56, pp. 533 – 541, 2010.

[48] Middlebury. (2003) Teddy stereo dataset. Website. Middlebury. [Online]. Available: http://vision.middlebury.edu/stereo/data/

[49] D. Scharstein and R. Szeliski, "High-accuracy stereo depth maps using structured light," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, vol. 1, Jun. 2003, pp. 195–202.

[50] C. M. Christoudias and B. Georgescu. (2002) Website. Rutgers University. [Online]. Available: http://coewww.rutgers.edu/riul/research/code/EDISON/index.html

[51] V. Kolmogorov, R. Zabih, and S. Gortler, "Generalized multi-camera scene reconstruction using graph cuts," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 2003, pp. 501–516.

[52] V. Kolmogorov. (2012) Match - stereo matching algorithm. [Online]. Available: http://pub.ist.ac.at/~vnk/software.html