DENSE LIGHTFIELD RECONSTRUCTION FROM MULTI APERTURE CAMERAS

Matthias Ziegler, Frederik Zilly, Peter Schaefer,
Joachim Keinert, Michael Schöberl and Siegfried Foessel

Fraunhofer Institute for Integrated Circuits IIS,
Am Wolfsmantel 33, 91058 Erlangen, Germany

ABSTRACT

Plenoptic cameras based on micro lens arrays as well as multi aperture cameras are able to capture a multitude of images with slightly shifted viewpoints. Although the amount of parallax between adjacent views is limited, precautions have to be taken in order to avoid alias when performing direct lightfield rendering. Against this background, we present an approach for the dense reconstruction of a lightfield based on a sparse lightfield acquired from a multi aperture camera with subsequent disparity estimation and depth image based view interpolation. Results show that the approach is suitable for all-in-focus-rendering.

Index Terms— *Lightfield, multi-aperture camera, plenoptic camera, view rendering*

1. INTRODUCTION

Since the first presentation of a commercial hand-held plenoptic camera by Ng in 2005 [1], the interest on this technology which previously focused on research in the academic sector [2], [3] was extended towards industrial applications and consumer electronics. Lightfield technology promises effects like digital refocus, free-viewpoint or rendering of synthetic apertures. However, many implementations show an inferior image resolution compared to traditional cameras. Active research is performed in the field of optics and algorithmic design with the goal to overcome this problem and to improve image quality.

The most important designs include different types of multi-camera arrays and plenoptic cameras like the ones commercially available by Raytrix [4], Lytro [5] or the PiCam presented by Pelican Imaging [6]. The basic principles of these cameras are similar while the implementations come with different baselines, resolutions, depth of field and number of views.

Multi camera-arrays are typically built from off the shelf cameras with no special requirements and are used for rendering a virtual camera at a position within the dimensions spanned by the array. Due to the relative low density of real camera positions additional information about the depth of objects in the scene is required for exact rendering of virtual camera positions [7].

In contrast a typical plenoptic camera as presented by Georgiev [8] and Ng [1] uses a microlens array in front of the sensor and an additional main lens. The angular sampling of the lightfield is denser compared to an array but due to the small baseline the amount of parallax is lower.

The PiCam [6] uses a total of 16 views. It does not require a main lens and can be built in a very compact way. Therefore, Venkataraman et al. [6] expect its application in mobile devices. The overlap between the single images of these cameras is very high. In combination with super-resolution approaches, the authors in [6] claim that the final output has higher resolution compared to the resolution of each single micro image.

A similar design was presented by Brückner et al. [9]. Their multi-aperture camera uses an array of 17x13 channels and also omits the main lens. In [10] Oberdörster et al. presented a stitching algorithm that can merge the channels using a constant disparity. As a result the final image shows sharp objects only at a specific distance and contains artifacts due to the missing parallax correction. As long as the raw lightfield data is preserved, digital refocus is possible by selecting the appropriate disparity value.

Against this background we present an improved processing chain for a multi-aperture camera. By combining disparity estimation with a subsequent depth-image-based rendering, a parallax-compensated result image can be generated.

The remainder of the paper is structured as follows: In section 2 we will have a brief look on specific properties of the multi-aperture camera and the necessary algorithms. Section 3 will explain the merging of disparity maps and how we can use them to render a final image. Results are presented in section 4 along with a comparison to a approach proposed by Georgiev in [8] and [11].

2. PREVIOUS WORK

2.1. Electronic cluster eye Camera (eCley)

In this paper we use the eCley camera presented in [9] which has 17x13 channels. Compared to a traditional camera with the same field of view (FOV) it has shorter track length.

According to [10], the overlap between channels is selected half the FOV. This means that a pixel at infinite distance can only be observed in adjacent channels. This is an important difference compared to the PiCam which has to be taken into account while designing the disparity estimation process. On the other hand, the eCley offers a larger FOV after combining all channels.

The camera sensor delivers a total resolution of 1536x2048 pixels while each channel has a resolution of 59x59 pixels.

2.2. Disparity estimation

A critical point in the processing chain for lightfield cameras is the disparity estimation. These algorithms provide estimates for the geometry corresponding to the scene. Georgiev presented in [8] an algorithm that computes disparity values for each microlens image and uses them to improve the rendering. This approach was further improved by Bishop [12], who's approach generates disparity maps at full resolution.

978-1-4799-5751-4/14/$31.00 ©2014 IEEE

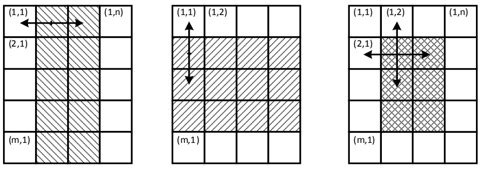

Figure 1: Merging process. Left: The horizontal merged disparity map at position (1,2) is obtained by merging disparity to left with disparity to right. Center: Vertical merge of position (2,1) with disparity to top and to bottom. Right: Merging of horizontal and vertical disparity maps to obtain a final disparity map at position (2,2).

In this work we used the algorithm presented in [13] to find an estimate for the disparity between adjacent channels. Typically, this algorithm is used for images taken with a camera array and computes stereo disparity estimates. A comparison with the approach from [8] and [11] is performed.

2.3. View rendering

Plenoptic and multi-aperture cameras do not directly deliver a final image. Instead, they deliver a number of images that show only a part of the scene. To exploit their full potential an additional step is required that fuses different views or generates new views.

As shown in [7], complexity and additional requirements for this step depend on the lightfield density. For very high densities, image rendering is possible without any knowledge about the underlying geometry. In case of a sparser lightfield sampling, image rendering requires additional information about the geometry in the scene which can be obtained by disparity estimation, a depth-camera, or a geometric model.

A depth-based rendering algorithm for plenoptic cameras was proposed by Georgiev [8]. In principle, this type of processing generates views with all objects in focus given high quality disparity maps. In order to obtain additional effects like refocus, additional processing steps are required. An improved processing chain that proposes a depth estimation algorithm in combination with a rendering algorithm specifically designed for plenoptic cameras has been presented by Bishop in [12]. However, their algorithm requires many floating point operations and is therefore hard to implement in hardware.

3. PROPOSED METHOD

We start from image data containing 17x13 individual channels. An example image is presented in Figure 4 (left). Details about the underlying pre-processing can be found in [14]. These single channel images are used as input to our processing chain. In order to increase precision and minimize the influence of rounding errors the channel images are upsampled by a factor two leading to a channel resolution of 118x118 pixels.

3.1. Stereo disparity estimation and merging

In the first processing step we estimate disparities between adjacent channels. Starting in the first row we compute disparities between the first and the second image and obtain one disparity map with disparities pointing from the left image to the right one and a disparity map with opposite disparities. This step is followed by disparity estimation between the second and the third image.

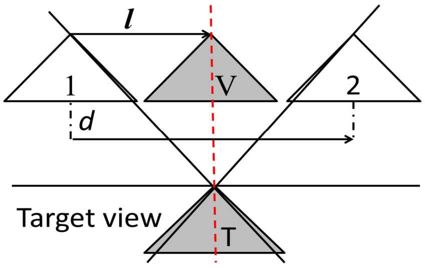

Figure 2: Cameras 1 and 2 capture images with parallax *d*. The target view T is below the baseline of channels. A projection of the red dashed ray in the target view can be obtained by warping the channel images to position V. *l* denotes the warping vector and is selected such that the targeted ray crosses the center of the warped view in V.

This continues to the end of the row and is repeated for every row. Similarly, we compute disparities from top to bottom images starting with the first image in the first column and estimate the disparity between this image and the image below.

This is repeated for the second and the third image, continues to the end of the column and is repeated for every column.

Unfortunately, these disparity maps might contain errors that need to be corrected by a subsequent confidence check and merging process. Possible errors include missing disparity values as well as invalid values. The merging process is depicted in Figure 1 and will be described in detail in the following. For all camera positions except the first and the last column we merge the disparity to the left with the disparity to the right and obtain a horizontally merged disparity map. We need to skip the first and last column as they don't have a disparity to the left or to the right, respectively. This step simply compares the disparities in both maps. If the difference for a pixel is below a threshold Θ an average value is computed and set in the output disparity map. If the difference is too high, the disparity at this point is marked as invalid. If only one input disparity map has a valid value it is set in the output map. This step has of course to be repeated for the top and bottom disparity maps to obtain vertically merged maps. In this case we need to skip the first row and the last row.

Next, we obtain merged disparity maps for the positions in the first and in the last row. Here we only have a disparity map to the bottom or to the top respectively instead of a vertically merged disparity map. The merging is the same as in the previous step. Accordingly, we apply the same step on the first and last column except the first and the last row.

Similar to the previous step we merge the obtained horizontal and vertical disparity maps to get a final disparity map for each camera position. During the merging process, we again compare disparity from the input disparity maps. In the case of inconsistent disparities, the luminance gradients in vertical and horizontal directions are analyzed using the Sobel operator. Based on the idea, that disparity estimation is more reliable at edges orthogonal to the search direction, the disparity value from the horizontal or vertical search directions are chosen depending on the gradients.

Finally we need to merge the disparity maps in the corners. As an example, we obtain the top-left disparity map by merging the disparity map to the right with the disparity map to the bottom. This step is repeated for all other corners with the appropriate inputs.

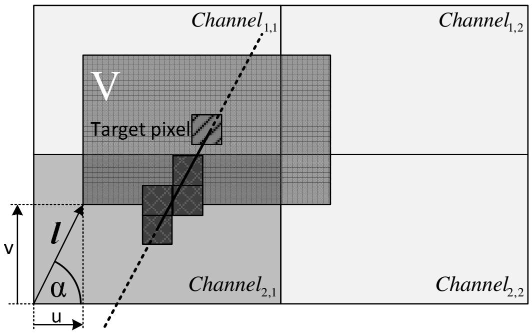

Figure 3: Channel $_{2,1}$ is warped to the position of view V. To find the target pixel in the center of V we need to warp only the disparity values along the shifted vector *l*. These pixels are marked in channel $_{2,1}$.

3.2. Parallax compensated view rendering

The disparity maps obtained are now used to render an all-in-focus image. A one dimensional model for the rendering is depicted in Figure 2. The extension towards 2D data is straightforward. The figure shows the position of the sub-cameras at the top and the position of the virtual target view at the bottom.

The rendering algorithm projects rays incident at the sub cameras and finds their corresponding position in the target view. As Figure 2 shows it is not possible to get a direct projection for every ray as is denoted by the red dashed ray. To obtain a projection for this ray we can warp the image from camera 1 by a distance *l* and obtain a view V. By definition we select *l* such that the targeted ray intersects with this warped view in its center. As one can see in Figure 3, for most pixels it is even possible to warp up to four channel images. Each of these channels can be warped to obtain a view at V and the targeted pixel in the center. At the end we need an additional step to merge these rendering results.

The process has to be carried out for every pixel in the final image. Relative to a given camera position, a channel image needs to be warped forward by an amount *u* in horizontal direction and *v* in vertical direction to obtain a view at V. The disparity in the center of V can subsequently be used for a backward warping to compute the target pixels' intensity value.

As we are only interested in the disparity value of the center pixel of our view we don't want to warp the full disparity map. Instead, we developed an optimized algorithm that is depicted in Pseudo code 1. In the forward warping step we only select the disparity values along the direction of ***l*** as shown in Figure 3. This vector is defined by *u* and *v* and can also be expressed by its length *l* and angle α. Given these two variables we know that the disparity value we are looking for can be found somewhere on the path between the center of the sub camera image (i.e. Channel $_{2,1}$) and the target view V. This path is shown in Figure 3 together with the marked pixels in channel $_{2,1}$. In addition with the maximum allowed disparity *dispMax* we can further decrease the search area by computing the maximum distance *radius* from the center pixel in the target view V to the pixel we are looking for assuming that all disparities have positive values. These selected pixels are stored in a vector, warped and finally scaled. The algorithm does not account for subpixel accuracy and non-integer pixel positions are rounded towards zero. In this step we further account for the possibility that objects in the foreground can be warped over background elements. Precedence is given to foreground objects.

```
Render pixel:
   % collect disparity values along l
   radius = dispMax · l
   for r = 0 to radius
      dx = cos(α)·r
      dy = sin(α)·r
      posX = floor(centerX + dx)
      posY = floor(centerY + dy)
      % assume (posX,posY) is a valid
        image position
      disp(r) = disparity(posX,posY)
   end

   % warp disparity vector
   for r = 0 to radius
      d = disp(r)
      r′ = r - floor(d · baseline)
      % Higher disparity overrides lower
        disparity
      if dispWarped(r′) <= d
         dispWarped(r′) = d
      end
   end

   % compute median disparity value over
     2p+1 values and scale by baseline
   d = median(dispWarped(-p:p)) · baseline

   % backward warping of rgbImage
   posX = floor(centerX - cos(α) · d)
   posY = floor(centerY - sin(α) · d)
   pixel = rgbImage(posX, posY)
```

Pseudo code 1: Render a pixel in the target view at (centerX, centerY) by warping a disparity map 'disparity'. l denotes the warping distance and alpha denotes its angle.

In order to account for errors in the disparity maps that can arise due to inevitable rounding operations, the resulting estimates can be stabilized by taking the median of adjacent *p* disparity values with *p* set to 3 in our case.

Finally, we obtain the pixel value in the center of that virtual camera by backward warping with the computed disparity. This forward and backward warping step is repeated for every camera position in the surrounding of this virtual camera delivering up to four disparity and pixel estimates. These estimates are averaged to get the final pixel and disparity value.

The code shows the case when *u* and *v* are both positive (right-upward warping) as shown in Figure 3. In other cases the sign of *dx* and *dy* has to be set according to the direction of ***l***. Compared to alternate view synthesis algorithm, minimized the number floating point operations while still delivering a high image quality and at full resolution.

4. EXPERIMENTAL RESULTS

In an experiment we compared the image quality of our proposed method with a rendering similar to the one found in [9] and [11]. As we didn't have an option to capture ground truth disparity, we cannot provide a numerical comparison and need to stick to visual evaluation. In Figure 4 details from the different processing steps are presented.

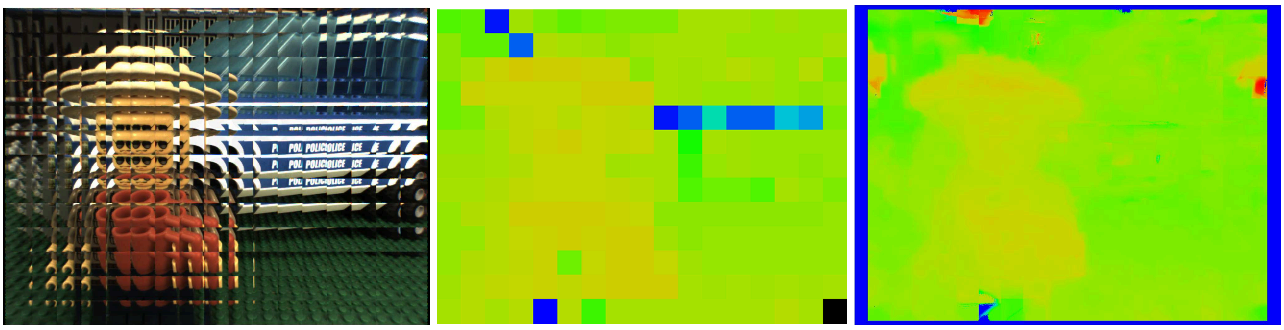

Figure 4: Left: Input image from eCley camera with preprocessed single channel images. Center: Disparity map obtained with algorithm presented in [8]. Red marks areas with high disparity. Green areas have medium disparity and blue denotes low disparity. Right: Warped disparity map obtained with proposed algorithm. Some unreliable values are visible in the border channels. It is clear to see that the policeman is closer to the camera compared to the car.

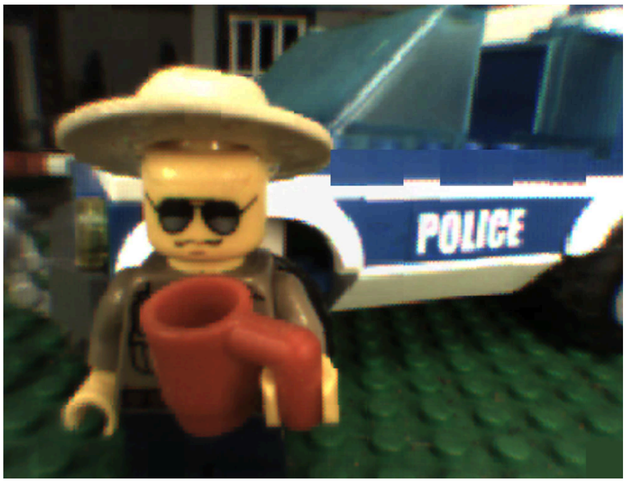

Figure 5: Patch based rendering from [8] and [11]. Artifacts due to invalid disparity values are visible on the car. The writing on the car shows some color artifacts.

The left image shows the preprocessed input image acquired with the eCley. The single channels and the overlap between channels can clearly be distinguished. The center image shows the disparity map obtained with the algorithm presented in [8] and [11]. In this image red areas mark high disparity whereas green areas mark lower disparity. Blue areas show invalid disparity. The right image finally shows the rendered disparity map obtained with our proposed processing.

The image in Figure 5 shows the result of the all-in-focus rendering algorithm proposed in [8] and [11]. The policeman in the foreground appears sharp without visible artifacts. In contrast, some artifacts are visible on the police car. As one can see in the center image of Figure 4, the disparity map contains errors in this area. Especially in homogenous areas we found that the block size used for disparity estimation in [8] is critical. Additional, the image shows some color artifacts around edges with high contrast.

Figure 6 shows the result obtained with our proposed method. The objects in the foreground as well as in the background appear sharp with almost no artifacts. Some artifacts are visible above the policeman's head and on the car. These are caused by small errors in the disparity maps that can also be seen in the right image in Figure 4. Color artifacts are clearly reduced.

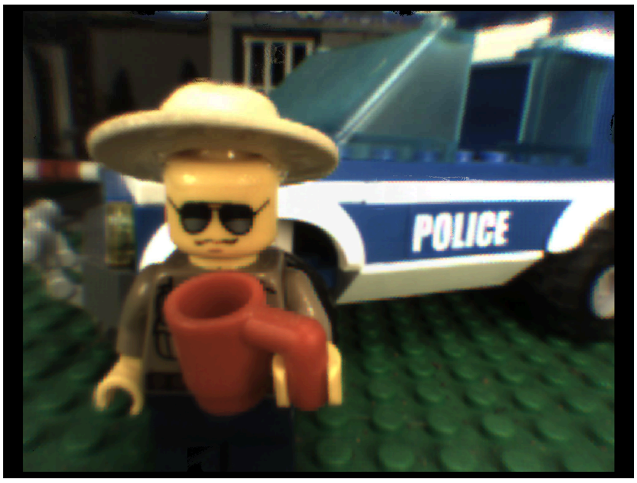

Figure 6: Proposed rendering method. Small artifacts exist above the policeman's hat and on the car. Color artifacts at edges with high contrast are clearly reduced.

5. CONCLUSION

The processing chain presented in this paper proposes a series of steps that improve image quality for the eCley camera. The employed stereo disparity estimator provides horizontal and vertical estimates that are merged in a subsequent step to obtain dense and consistent disparity maps at full resolution. These estimates are finally used in the rendering to obtain an image at high resolution and small artifacts.

Instead of a patch based rendering, the final image is rendered with an algorithm that accounts for the estimated depth for every pixel.

The final image quality is compared at visual level. These images show on the one hand that our processing removes artifacts and increases image quality. On the other hand, one can see that our disparity maps still contain errors. However, only small artifacts are visible in the final image and implementation of an optimized rendering algorithm i.e. on a GPU is a reasonable task for future work.

The overall image quality will further benefit by improving the underlying algorithms for disparity estimation and merging which is a task for future research.

7. REFERENCES

[1] Ng R., Levoy M., Brédif M., Duval G., Horowitz M. and Hanrahan P., “Light field photographie with a hand-held plenoptic camera”, *Computer Science Technical Report,* vol. 2, no. 11, Stanford, USA, 2005

[2] Gortler S. J., Grzeszczuk R., Szeliski R., Cohen M. F., „The Lumigraph“, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, pp. 43-54, 1996

[3] Levoy M., Hanrahan P., “Light field rendering”, *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, ACM, pp 31-42, 1996

[4] Perwaß C. and Wietzke L., "Single lens 3d-camera with extended depth-of-field," *Human Vision and Electronic Imaging XVII,* Proc. SPIE, Vol. 8291, Burlingame, USA, 2012.

[5] https://www.lytro.com/

[6] Venkataraman K., Lelescu D., Duparré J., McMahon A., Molina G., Chatterjee P. and Mullis R., "PiCam: an ultra-thin high performance monolithic camera array." *ACM Transactions on Graphics* (TOG), vol. 32, no. 166, New York, 2013.

[7] Shum H. and Kang S., "Review of image-based rendering techniques," *Visual Communications and Image Processing 2000,* Proc. SPIE, vol. 4067, Perth, Australia, 2000

[8] Georgiev, T. and Lumsdaine A., "Focused plenoptic camera and rendering," *Journal of Electronic Imaging,* vol. 19, no. 2, 2010

[9] Brückner A., Duparré J., Leitel R., Dannberg R., Bräuer A. and Tünnermann A., "Thin wafer-level camera lenses inspired by insect compound eyes," *Optics Express,* OSA, vol. 18, no. 24, pp. 24379-24394, 2010

[10] Oberdörster A., Brückner A., Wippermann F., Bräuer A., Lensch H., "Digital focusing and refocusing with thin multi-aperture cameras," *Proc. SPIE 8299, Digital Photography VIII*, Burlingame, USA, vol. 8299, 2012.

[11] Georgiev, T., and Lumsdaine A., "Reducing plenoptic camera artifacts," *Computer Graphics Forum,* Blackwell Publishing Ltd, vol. 29, no. 6, 2010.

[12] Bishop, T. E., and Favaro P. "Full-resolution depth map estimation from an aliased plenoptic light field," *Computer Vision–ACCV 2010,* Springer, vol. 6493, pp. 186-200, 2011

[13] Riechert, C., Zilly F., Müller M. and Kauff P., "Real-time disparity estimation using line-wise hybrid recursive matching and cross-bilateral median up-sampling," *21st International Conference on Pattern Recognition (ICPR),* IEEE, Tsukuba, Japan, pp. 3168 – 3171, 2012.

[14] Oberdörster A., Brückner A., Wippermann, F. C., Bräuer, A., "Correcting distortion and braiding of micro-images from multi-aperture imaging systems," *Proc. SPIE 7875, Sensors, Cameras, and Systems for Industrial, Scientific, and Consumer Applications XII*, San Franciso, USA, vol. 78750B, 2011