Content-Based Depth Estimation in Focused Plenoptic Camera

Kalin Atanassov*, Sergio Goma*, Vikas Ramachandra* and Todor Georgiev**

ABSTRACT

Depth estimation in focused plenoptic camera is a critical step for most applications of this technology and poses interesting challenges, as this estimation is content based. We present an iterative algorithm, content adaptive, that exploits the redundancy found in focused plenoptic camera captured images. Our algorithm determines for each point its depth along with a measure of reliability allowing subsequent enhancements of spatial resolution of the depth map. We remark that the spatial resolution of the recovered depth corresponds to discrete values of depth in the captured scene to which we refer as slices. Moreover, each slice has a different depth and will allow extraction of different spatial resolutions of depth, depending on the scene content being present in that slice along with occluding areas. Interestingly, as focused plenoptic camera is not theoretically limited in spatial resolution, we show that the recovered spatial resolution is depth related, and as such, rendering of a focused plenoptic image is content dependent.

1. INTRODUCTION

Plenoptic cameras are application of light-field capture that exploit redundancy in a projected captured image to recover a light field. The plenoptic camera, a digital realization of Lippmanns integral photography,[1] was introduced in 1992[2] as an approach to solve computer vision problems. An improved version, the focused plenoptic camera (sometimes referred to as plenoptic camera 2.0), has been independently introduced in[3,4,5] and also follows ideas originating from Lippmann[1] and Gershun.[6] In one realization, the camera has microlenses placed at distance b from the sensor, and focused at the image plane of the main camera lens, at a distance a in front of them (see Figure 1). In this configuration, a;b; and the focal length f satisfy the lens equation, and each microlens constructs a relay system with the main camera lens. Capturing data with plenoptic cameras makes possible greater processing capabilities and solves many of the problems faced by photographers using conventional digital cameras. Rendering refocused images[3,2,7] 3D views[8,9] HDR[10] and super-resolution[11,12,13] are just a few applications of plenoptic cameras. At the same time, the focused plenoptic camera overcomes the problem of poor resolution[5,14] that was a main drawback for the first handheld Plenoptic camera.[15]

Our implementation of focused Plenoptic camera is based on an arrangement of main lens and an array of micro lenses placed in front of a 39mpixels full-frame CCD. Based on each rays focus point relative to micro lens plane, there can be distinguished three different operating modes:

*Qualcomm Inc., 5775 Morehouse Drive, San Diego CA 92121.
**Adobe Systems, 345 Park Ave, San Jose, CA 95110

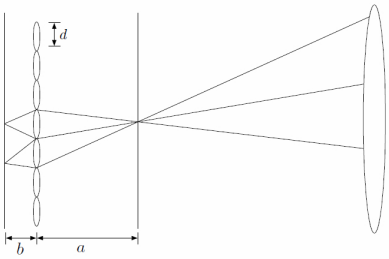


Figure 1. Focused Plenoptic camera: the microlens array image the focal plane of the main lens

Three-Dimensional Imaging, Interaction, and Measurement, edited by J. Angelo Beraldin, et al.,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7864, 78640G · © 2011 SPIE-IS&T
CCC code: 0277-786X/11/$18 · doi: 10.1117/12.872667

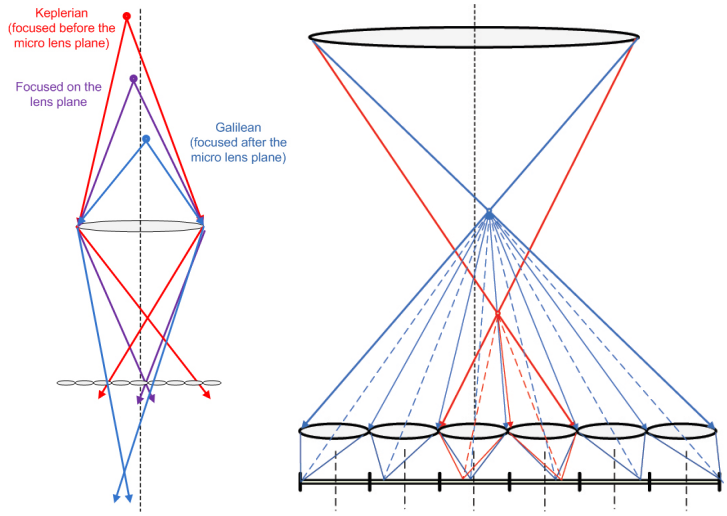


Figure 2. left - Plenoptic camera capture mode depending on distance to content; right - Depth and relative shift in microimages in focused plenoptic camera

- Ray focused on micro lens plane In this mode, the functionality is that of a Plenoptic camera 1.0;
- Ray focused before micro lens plane In this mode, the functionality is as of a Keplerian telescope;
- Ray focused after micro lens plane In this mode, the functionality is as of a Galilean telescope;

As illustrated in the next figure, the operating mode will be dependent on the object distance from camera in essence a scene will be captured as a mixture of modes, depending on the scene content more specifically, depending on the placement of content at a specific depth.

As it can be seen in the figure, some content can appear mirrored (Keplerian mode) whereas other content can be normal (Galilean mode). Due to the microlens array, there will be some redundancy of the information captured, but this redundancy is again, depth dependant. In other words, the same point in space will be captured in several micro-images with a slight different angular perspective aspect discussed in detail in[16].[17] We remark that the spatial resolution of a recovered depth corresponds to discrete values of depth in the captured scene to which we refer to as depth slices. With this in mind, one can easily observe that there is a relation between the relative shift of a projection of a point in different microimages and the depth slice containing that point. Interestingly, for different depth slices, the number of projections of the same point is different, establishing another depth clue. The above is illustrated in the figure - right.

In our algorithm, we are exploiting those two depth clues to recover both the spatial resolution as well as the inferred depth, and in this process we also determine a measure of reliability of the recovered depth. The paper is organized as follows: we describe the framework of our algorithm, followed by implementation details where we formalize our algorithm and detail its implementation. Our algorithm determines for each point its depth along with a measure of reliability allowing subsequent enhancements of spatial resolution of the depth map. We present results on a captured image with our focused plenoptic camera and discuss some of the enhancements we used to recover some of the uncertain data. We apply our research on depth estimation in an example of a rendering of captured plenoptic data to illustrate the importance of this step for further applications of focused plenoptic camera. We conclude by summarizing our findings.

2. ALGORITHM FRAMEWORK

The proposed algorithm will extract from plenoptic data a depth map. We call this depth map a plenoptic depth map, as it has the same characteristics as the original plenoptic data there is a one to one correspondence



Figure 3. Plenoptic Depth estimation algorithm framework

between the plenoptic data and the plenoptic depth map. The usable resolution of the plenoptic data is dependant of the plenoptic operating mode and of the content of the scene. Specifically, it depends on the depth slice where the content is located. As such, depending on the depth slice, from each micro image in the plenoptic data more or less pixels are usable (from resolution perspective). To render a depth slice we crop all the non overlapping pixels around the center of each micro image. To each depth slice corresponds a particular size of the cropping window. This size corresponds to the same shift of a detail found in adjacent microimages. This implies a correspondence between this shift found in adjacent microimages and the depth slice associated to that detail. Our algorithm uses this correspondence to infer from shift its associated depth.

To further illustrate the above we refer to fig.1 noting that our depth estimation algorithm comprises of the following blocks:

- Pixel shift estimator this block would estimate pixel shift as it appears in the adjacent microimages. Of course, this is content dependent and we detail this block in 2.1.
- Unreliable regions estimator this block provides the confidence metric of the shift estimate produced by the previous block. As the shift estimation is content dependent, there are cases where the content does not offer enough details to allow a good estimation in these cases the associated pixels are marked unreliable. We detail this block in 2.2
- Shift estimation for uncertain/undetermined regions this block further processes all the pixels that are deemed unreliable by the previous module trying to infer their depth estimation from reliable neighbors and the original plenoptic data. This block is discussed in 2.3
- Shift/depth mask outage/noise reduction processing in this block we clean the depth mask. We choose here to use diffusion to remove any gaussian type of noise and a median filter to remove salt and pepper type of noise - we note that any edge preserving type of processing would work in this step.

2.1 Shift Estimator

This block will estimate pixel shift as it appears in the adjacent microimages. For every pixel we search for correspondences in the adjacent microimages. We establish a correspondence when the variance of a spatial comb like filter spaced with a specific offset in adjacent microimges is minimal. For simplicity, we call this filter a comb. For each particular offset value we calculate 9 variances corresponding to an area of 5x5 microimages.In doing this, we try to eliminate potential occlusions that would deem unreliable the value of the pixel shift. This issue is illustrated in the following figure:

The red object is the one which depth is to be estimated. If the surrounding microimages are searched for a match, the shift will be estimated incorrectly due to a occlusion from the foreground - microimages on the right do not have the red object. In our case, as we're using a 3x3 comb, the 2 microimages on the left could be used instead of one on the left and one of the right.

In conclusion, our shift estimation algorithm uses the following steps:

- For every pixel that requires depth estimation (usually a square crop around the center of the micro image) do:
  - For every hypothesized offset do:

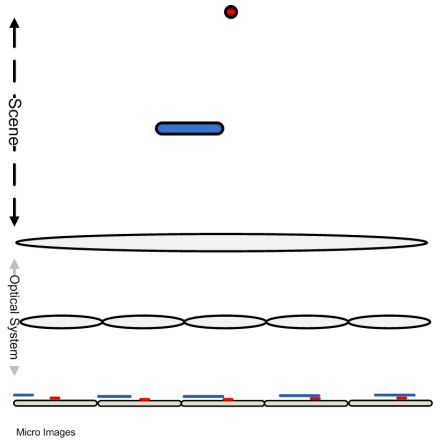


Figure 4. Example of an occluded area around an edge

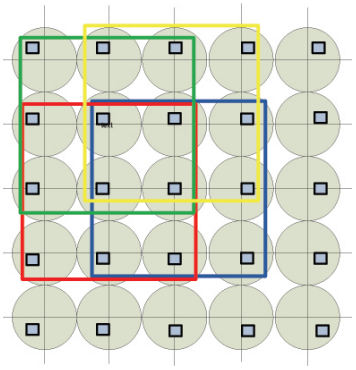


Figure 5. Micro lens search windows - for illustration purposes not all of them are shown

  - Form 9 combs comprised of a small neighborhood around the point (3x3 pixels) over the neighboring lens placed with the hypothesized offset (using 3x3 lens grid) (for illustration purpose only several combs are shown)
  - Pick the comb that has the smallest variance along the lens dimension that is comb that is the most likely to not contain occluded pixels
- The offset/shift that yields the minimum variance over the lens dimension is picked as the estimated shift

Analytical description
Notations

$$S_{K,L}(i,j) - plenoptic\ image$$

$$i,j - pixel\ coordinates\ within\ the\ micro\ image$$

$$K,L - micro\ lens\ coordinates\ \Delta - crop\ size$$

$$S_{K,L}(h/2-\Delta/2 : h/2+\Delta/2, w/2-\Delta/2 : w/2+\Delta/2) - center\ crop\ to\ be\ processed$$

To estimate the shift for a particular pixel, compute the following (the examples used 3x3 lens window)

$$
\begin{aligned}
&for \quad off = y1 : y2 \quad (\text{offset/shift search range}) \\
&\quad for\ (all\ pixels\ in\ the\ pixel\ window)\ m = -1:1, \quad n = -1:1 \\
&\quad\quad for\ (\text{occlusion avoidance margin})\ oi = -1:1, \quad oj = -1:1 \\
&\quad\quad\quad for\ (\text{lens window})\ I = -1:1, \quad J = -1:1 \\
&\quad\quad\quad\quad P(3(I+2)+J+2) = S_{K+I,L+J}(i-(I+oi)\times off + m, j-(J+oj)\times off + n) \\
&\quad\quad V_{off}^{m,n}(oi,oj) = \text{var}(P) \\
&\quad V(off) = \min_{oi,oj}\left(\sum_{m,n} V_{off}^{m,n}(oi,oj)\right) \\
&\hat{off} \quad \rightarrow \quad V\left(\hat{off}\right) = \min(V(off))
\end{aligned}
$$

2.2 Unreliable regions estimator

This block provides the confidence metric of the shift estimate produced by the previous block. Basically, in this block we deem unreliable flat regions. To accomplish this, one or more of the following can be used:

- Minimum variance at the estimated shift
- Maximum variance over all shifts
- High pass operation on the pixel
- Pseudo SNR

We choose to use in our implementation a pseudo SNR, that gives an indication of high-frequencies around the point as reliability estimate:

$$R(i,j) = max(V(off)/min(V(off)) \tag{1}$$

$$R(i,j) > \delta - reliability\ mask \tag{2}$$

2.3 Unreliable regions shift recovery

In this block we will further process all the pixels that are deemed unreliable by the previous module trying to infer their depth estimation from reliable neighbors and the original plenoptic data. Therefore once an estimated pixel shift is deemed unreliable, it should be recovered automatically using the surrounding (reliable) shift estimates. This step can be summarized as follows:

- for every unreliable pixel do:
  - Define a region NxN around that pixel (typically 40x40);
  - Select only pixels from the region that are marked as reliable;
  - From the selected above pixels select the ones that are within certain values from the pixel under estimation;
  - If there is at least one pixel selected:
    - Use the selected above pixel shift estimate(s) to produce shift estimate for the pixel of interest;
    - Mark that pixel as reliable on reliability mask;
  - Once all pixels checked repeat iteratively for the whole image until all unreliable pixels are recovered:

Or analytically:

$$S(i,j) - unreliable\ pixel\ shift(R(i,j) = 0)$$

$$d = \sum_{k=1}^{3} (I(i-\delta : i+\delta, j-\delta : j+delta, k) - I(i,j,k))^2 \tag{3}$$

$$k = 1,2,3 - red,\ green,\ blue,\ \delta - range$$

$$mask = C(d) \times R(i-\delta : i+\delta, j-\delta : j+delta) \tag{4}$$

$$C() - monotonically\ decreasing\ curve$$

$$S(i,j) = \frac{\sum_{m=-\delta}^{\delta} \sum_{m=-\delta}^{\delta} I(i-m, j-n) \times mask(m,n)}{\sum_{m=-\delta}^{\delta} \sum_{m=-\delta}^{\delta} mask(m,n)} \tag{5}$$

where I is the plenoptic data, S is the shift mask and R is the reliability mask. Once the reliability mask is filled with reliable estimates, it is not used further and at this point, the depth mask is fully populated.

2.4 Noise/outage reduction

As explained before, the depth mask can be further cleaned with one or more of the following:

- Anisotropic diffusion
- Morphological operations (dilate , erode, open, close)
- Order statistic filtering (median, min, max)

In our current implementation, a combination of diffusion and median filtering is used in this step.

First diffusion:

$$d = |S_{in}(i-\delta : i+\delta, j-\delta : j+delta) - S_{in}(i,j))| \tag{6}$$

$$mask = C(d) - C()\ monotonically\ decreasing\ curve \tag{7}$$

$$S_{out}(i,j) = \frac{\sum_{m=-\delta}^{\delta} \sum_{m=-\delta}^{\delta} S_{in}(i-m, j-n) \times mask(m,n)}{\sum_{m=-\delta}^{\delta} \sum_{m=-\delta}^{\delta} mask(m,n)} \tag{8}$$

Followed by median filter:

$$S_{out} = median(S_{in}(i-\delta : i+\delta, j-\delta : j+delta)) \tag{9}$$

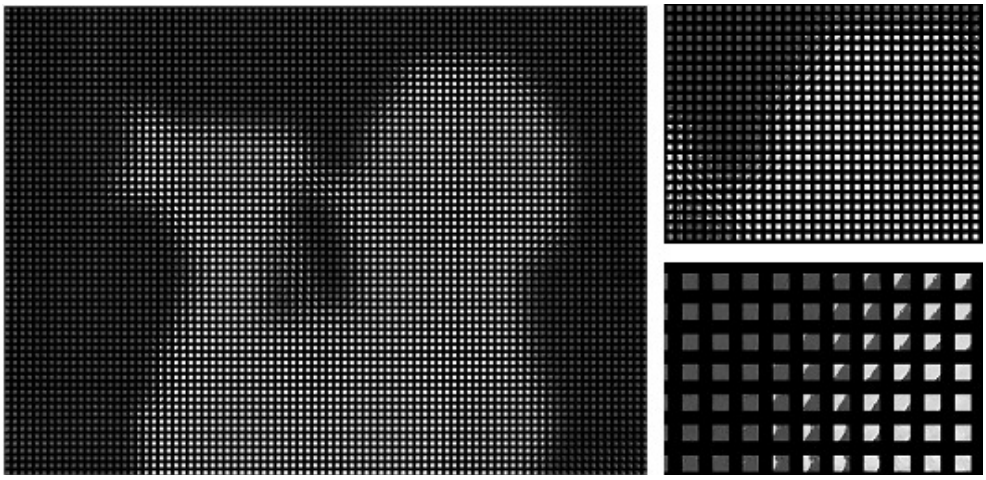


Figure 6. Plenoptic depth mask: full image and crops at different resolutions

3. RESULTS DEPTH MASK USED IN IMAGE RENDERING

Scene depth range determines the range relative to the micro lens plane at which objects will focus. The depth of a point will determine the number of microimages in which that point will be found, the shift associated to that point in different microimages, and the amount of overlap. Example: A point that shows on adjacent microimages and has a small shift will show on more microimages compared with a point with a bigger shift. In other words, knowledge of objects depth is necessary for optimal object rendering (i.e., objects native resolution). We're using as an application example for processing plenoptic data, rendering all in focus. Here, we start by producing the depth mask, a step that should be used for a generic plenoptic data application and concluding by rendering the final image all in focus, noting that this step is just used to exemplify our processing.

3.1 Producing depth mask

A 2D image can be rendered at resolution R from plenoptic data and the depth mapping using the following procedure:

- From the depth mask determine the range of pixel shift [r1:r2] Note: Usually, plenoptic cameras are configured such as they will work either in Keplerian or Galilean mode (signs of r1 and r2 are the same);
- Render mask at different pixel or subpixel resolution from abs(r1) to abs(r2);

Different rendering resolutions can be achieved at its simplest, just by cropping and stitching center windows from microimages.

- Produce different depth slices corresponding to resolution range [abs(r1), abs(r2)] and up-sample them to the target resolution;
- Using the depth slice with the highest resolution as a starting point for the reference image, create a depth mask by fetching every pixel from the depth slice that corresponds to its value in the reference image;
- Update the reference image and perform several iterations.

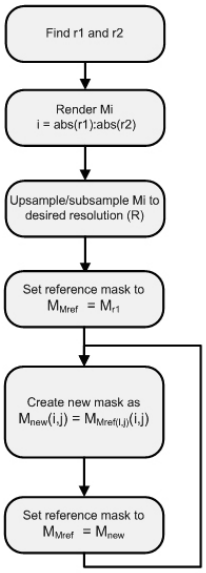


Figure 7. Depth mask rendering flowchart

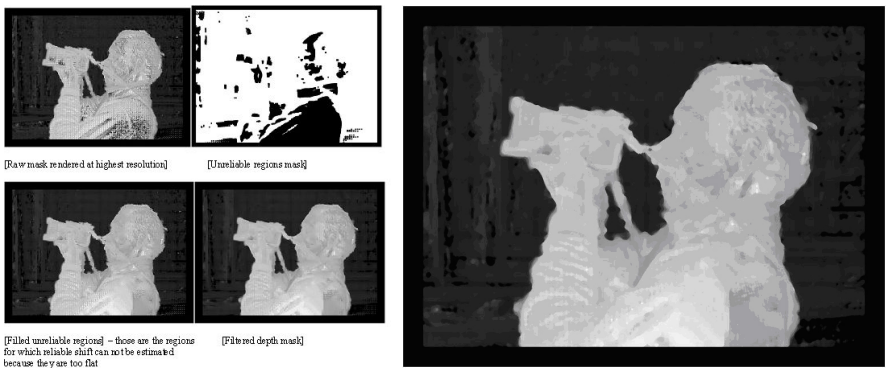


Figure 8. left - depth mask stages; right - rendered depth mask

3.2 Rendering image (all in focus)

Using the depth mask with depth controlled resolution (the rendered depth mask) one can render a 2D image from plenoptic data in following steps:

-
- Using the depth slice with the highest resolution, create an intermediate rendering of the final image by Render the depth mask using the depth slices produced in previous step;
- Using the depth mask with depth controlled resolution create final image by fetching every pixel from the image with resolution that corresponds to its value in the rendered depth mask

Simple rendering at any single resolution is done by fetching NxN pixel around the center of every micro lens and putting the patches together . For that particular example the mask is rendered at 12x12 the lens resolution

since depth mask contains shifts from -12 to -6 , 7 different resolution are created and upsampled to the desired resolution. Then using a starting depth mask (resolution 12x12) every pixel is picked from its corresponding mask



Figure 9. left - one rendered depth slice; right - rendered image using depth map (all depth slices)

The next image is rendered at resolution 12x12 selected on purpose since at that resolution the person (foreground) will be in focus . As seen from the plot the back ground is not in focus

based on the final depth mask . 7 image resolutions are created (from 6 to 12 time lens resolution) and up-sampled to desired resolution. Pixels are then selected from their native resolution based on their depth value.

4. CONCLUSIONS

Using plenoptic data in an application benefits from having the associated depth information, as the plenoptic data can be considered a discrete sum of 2D light-field projections. In this context - looking from the captured plenoptic data - the spatial resolution of the recovered depth corresponds to discrete values of depth in the captured scene - slices. Each slice has a different depth plus will allow extraction of different spatial resolutions of depth, depending on the scene content being present in that slice along with occluding areas. We proposed an algorithm that first recovers from plenoptic data the depth from points that have sufficient frequency content we associate to those points high depth reliability. At this point, we have a map of reliable depth points corresponding to areas in the plenoptic data that have sufficient frequency content. Next, we restore the points that have unreliable depth from their neighbors. The final step in obtaining the depth map is noise and outage reduction, where outliners are removed. Here, the depth map obtained has similar spatial characteristics as the original plenoptic data we call this a plenoptic depth map. Further usage of the plenoptic depth map depends on the application, but a common use case would involve producing of a plenoptic depth map. In the application we present in this paper, which is rendering of the plenoptic data, this is the case, and as such, we present a method of obtaining a rendered depth map (the rendered depth map has the same characteristics as the final rendered image). Our method of producing the rendered depth map involves determining the range of depth which relates to the range of spatial resolutions present in the plenoptic data in this step we determine the maximum resolution of the scene. Next, we use as target spatial resolution the maximum resolution of the scene, but this is not a requirement. At this point, we have the final spatial resolution of the image and are able to produce all the slices corresponding to the different depths present in the plenoptic data. Those slices are produced using crops of the plenoptic depth map that are with the size of the corresponding shift associated to a particular depth.

REFERENCES

1. G. Lippmann., "Epreuves reversibles donnant la sensation du relief," 1908. Journal of Physics, 7(4):821-825,.
2. T. Adelson and J. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis and Machine Intelligence* , pp. 99–106, 1992.
3. R. Ng., "Digital light field photography.," 2006. PhD thesis, Stanford University, Stanford, CA, USA.
4. K. Fife, A. E. Gamal, and H.-S. P. Wong., "A 3mpixel multi-aperture image sensor with 0.7mm pixels in 0.11mm cmos," 2008. IEEE ISSCC Digest of Technical Papers, pages 48-49.
5. A. Lumsdaine and T. Georgiev., "Full resolution lightfield rendering.," 2008. Technical report, Adobe Systems.

6. A. Gershun, "The light field," 1939. Journal of Mathematics and Physics, Vol. XVIII, MIT, 1939, pp. 51151,.
7. A. Veeraraghavan, A. Mohan, A. Agrawal, R. Raskar, and J. Tumblin, "Dappled photography: Mask enhanced cameras for heterodyned light fields and coded aperture refocusing," *ACM Trans. Graph.* .
8. E. H. Adelson and J. Y. Wang, "A stereoscopic camera employing a single main lens," 1991. CVPR91(619-624).
9. A. Isaksen, L. McMillan, and S. Gortler., "Dynamically reparameterized light fields.," 2000. SIGGRAPH 00, pages 297306,.
10. T. Georgiev, A. Lumsdaine, and S. Goma, "High dynamic range image capture with plenoptic 2.0 camera," 2009. Signal Recovery and Synthesis (SRS), Fall OSA Optics and Photonics Congress.
11. T. Bishop, S. Zanetti, and P. Favaro, "Light field superresolution," *Proceedings ICCP 2009* .
12. T. Georgiev and A. Lumsdaine., "Superresolution with plenoptic camera 2.0," 2009. Adobe Tech Report.
13. T. Georgiev and A. Lumsdaine, "Superresolution with plenoptic 2.0 cameras," 2009.
14. T. Georgiev and A. Lumsdaine, "Resolution in plenoptic cameras," 2009. COSI.
15. R. Ng, M. Levoy, M. Br'edif, G. Duval, M. Horowitz, and P. Hanrahan., "Light field photography with a hand-held plenoptic camera.," 2005. Technical Report CSTR 2005-02, Stanford Univ,.
16. T. Georgiev, C. Zheng, B. Curless, D. Salesin, S. Nayar, , and C. Intwala, "Spatio-angular resolution tradeoff in integral photography," 2006. Proceedings of Eurographics Symposium on Rendering,.
17. J. and S.Chan and H.Shum and X.Tong, "Plenoptic sampling," 2000. ACM Trans. Graph. pp307–318.