Light-Field Demultiplexing and Disparity Estimation

Neus Sabater, Valter Drazic, Mozhdeh Seifi, Gustavo Sandri, Patrick Pérez

Light-Field Demultiplexing and Disparity Estimation

Neus Sabater Valter Drazic Mozhdeh Seifi Gustavo Sandri
Patrick Perez
Technicolor Research and Innovation, France.
Name.Surname@technicolor.com

Abstract

In this paper we study the post-processing pipeline to recover the views (light-field) from the raw data of a plenoptic camera such as Lytro. First, the microlens centers are estimated and then the raw image is demultiplexed without demosaicing it beforehand. This avoids image artifacts due to view cross-talk. Furthermore, we present a new block-matching algorithm to estimate disparities for plenoptic views that have not been demosaiced. Our algorithm enforces the coherence through the views thanks to the view configuration given by the plenoptic camera: (i) the views are horizontally and vertically rectified and have the same baseline, and therefore (ii) at each point, the vertical and horizontal disparities are the same. Finally, we show that disparity estimation is more accurate when the raw image is demultiplexed without demosaicing the raw image. In particular, we show that our algorithm outperforms the disparity estimation method in [17].

1. Introduction

Plenoptic cameras are gaining a lot of popularity in the field of computational photography because of the additional information they capture compared to traditional cameras. Indeed, they are able to measure the amount of light traveling along each ray bundle that intersects the sensor, thanks to a microlens array placed between the main lens and the sensor. As a result, such cameras have novel post-capture processing capabilities. For example, after the image acquisition, the point of view, the focus or the depth of field can be modified. Also, from the obtained sampling of the light-field the scene depth can be estimated from a single snapshot.

There are several optical designs for plenoptic cameras. In particular, plenoptic cameras including a microlens array are divided in two types depending on the distance between the microlens array and the sensor: either the distance is equal to the microlenses focal length (as presented by Ng et al. in [13]) or it differs from it (as presented by Lumsdaine and Georgiev in [12]). In the first case the number of pixels per rendered view is equal to the number of microlenses. In the second case, the rendered views have a higher spatial resolution but at the cost of decreasing the angular resolution. Depending on the application, one type of camera or another would be preferred.

The concept of *integral photography*, which is the underlying technology in plenoptic cameras, was introduced by Lippmann [11] and then brought up to computer vision by Adelson and Wang [2], but it has recently become practical with the hand-held cameras that Lytro[1] and Raytrix[2] have put on the market for the mass market and professionals respectively. Since then, the scientific community has taken an interest in the light-field technology. In particular the computational photography literature is mostly expanding on image processing rather than hardware improvements regarding plenoptic cameras. Above all, the recent studies in the field address the bottleneck of the plenoptic cameras, namely the resolution problem [8, 4, 14, 18]. Besides super-resolution, depth estimation has also been investigated as a natural application of plenoptic images [4, 18, 17]. Indeed, the intrinsic information of the light-field has the advantage to allow disparity computation without the image calibration and rectification steps required on classic stereo or multi-view algorithms, making it an enormous advantage for 3D applications. However, the last cited works consider the light-field (the set of demultiplexed views) as input for their disparity estimation methods, meaning that they do not study the process that converts the raw data acquired by the plenoptic camera into the light-field. In this paper we show that such processing called *demultiplexing* is of paramount importance for depth estimation.

The contributions of this paper are twofold. First, we model the demultiplexing process of images acquired with a plenoptic camera such as the Lytro [13] and then we present a novel algorithm for disparity estimation specially designed for the singular qualities of plenoptic images: a matrix of views horizontally and vertically rectified with

[1] http://www.lytro.com
[2] http://www.raytrix.de

same baseline, where each view captures the scene under a different angle of incidence. This matrix of views is also called *Ligh-field*. In particular, we show that estimating disparities from undemosaiced views is preferred to using views obtained through conventional linear demosaicing on the raw data. Fig. 1 illustrates the pipeline of our method.

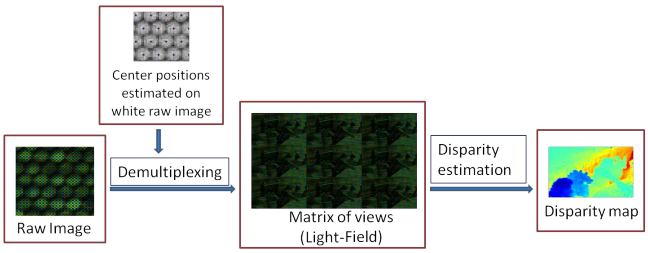


Figure 1. Pipeline of our method. For visualization purposes only a part of the subimages and the views on the matrix are shown. The light-field (matrix of views) is obtained demultiplexing undemosaiced data and using the center subimage positions. Then the disparity map for a reference view is estimated from the light-field.

2. Related Work

The closest works to our demultiplexing method have been published recently [6, 5]. On the one hand, Dansereau et al. [6] propose a demultiplexing algorithm followed by a rectification step where lens distortions are corrected using a 15-parameter camera model. On the other hand, Cho et al. [5] also propose a demultiplexing algorithm for the Lytro camera and study several interpolation methods to superresolve the reconstructed images. The main difference of our approach compared to [6] and [5] is that the raw data is not demosaiced which avoids view cross-talk artifacts.

Considering disparity estimation for plenoptic images, a variational method is described using the epipolar plane image (EPI) in [18]. [3] and [4] propose a variational method too, coupled with an antialiasing filtering to avoid cross-talk image artifacts. [10] estimates disparity maps from high spatio-angular light-fields with a fine-to-coarse algorithm where disparities around objet boundaries are first estimated using an EPI-based method and then propagated. Finally, [17] proposes an interesting approach that combines defocus and correspondence to estimate the scene depth. In this paper, we present a different disparity estimation method based on block-matching. Our method is generic for any plenoptic data, but it is particularly useful for Lytro data as we will see in the experiments. In fact, we have observed that EPI's from Lytro images are highly noisy and in that case the disparity is only reliable on a small number of pixels (object edges).

Finally, image demosaicing has barely been treated in the literature and most of the cited works, with the exception of Yu et al [19], propose to perform the demosaicing on the raw image using any state-of-the art method. However, this simple solution creates image artifacts on the rendered views due to image cross-talk as pointed out by [8]. In addition we have observed that demosaicing the raw image causes tremendous errors on the disparity maps. Therefore, for the sake of accurate disparity estimation, we propose to demultiplex directly the undemosaiced raw image and estimate the disparity with the incomplete but reliable views. To the best of our knowledge this approach has never been proposed for plenoptic disparity estimation.

3. Demultiplexing RAW data

Demultiplexing (also called decoding in [6] or calibration and decoding in [5]) is the data conversion from the 2D raw image to the 4D light-field, usually represented by the two-plane parametrization [9]. Concerning Lytro, the raw data is not provided with its software, therefore we have used the tool from [1] to convert the data with the Lytro file format to raw data. Then, the demultiplexing process consists in reorganizing the pixels of the raw image in such a way that all pixels capturing the light rays with a certain angle of incidence are stored in the same image creating the so-called *views*. Each view is a projection of the scene under a different angle of view. The set of views create a block matrix where the central view store the pixels capturing perpendicuar light rays to the sensor. In fact, the angular information of the light rays is given by the relative pixel positions in the *subimages*[3] with respect to the subimage centers. After demultiplexing, the number of restored views (entries of the block matrix) corresponds to the number of pixels covered by one microlens and each restored view has as many pixels as the number of microlenses.

Estimating Subimage Centers

The microlens array is placed in front of the sensor but the microlens centers are not necessarily well aligned with the pixels of the sensor. Indeed, the microlens diameter do not cover an integer number of pixels. Furthermore, there is a rotation offset between the sensor and the microlens plane. Finally, the microlenses are arranged on a quincunx grid to efficiently cover the space. Thus, in order to robustly estimate the microlens centers, we estimate the transformation between two coordinate systems (CS), the Cartesian CS given by the sensor pixels and K the microlens center CS defined as follows: the origin is given by the center of the topmost and leftmost microlens and the basis vectors are the two vectors from the origin to the adjacent microlens centers (see Fig.2). Formally, if $\mathbf{x}$ and $\mathbf{k}$ are respectively

[3]The image that is formed under a microlens and on the sensor is called a subimage of the scene.

the coordinates on the sensor and microlens CS, then, we aim to estimate the system transformation matrix $\mathbf{T}$ and the offset vector between the origins $\mathbf{c}$ such that

$$\mathbf{x} = \mathbf{T}\mathbf{k} + \mathbf{c}\,. \tag{1}$$

More precisely,

$$\mathbf{T} = \begin{pmatrix} 1 & 1/2 \\ 0 & \sqrt{3}/2 \end{pmatrix} \begin{pmatrix} d_h & 0 \\ 0 & d_v \end{pmatrix} \begin{pmatrix} \cos(\theta) & -\sin(\theta) \\ \sin(\theta) & \cos(\theta) \end{pmatrix}, \tag{2}$$

where the first matrix accounts for the orthogonal to hexagonal grid conversion due to the quincunx microlens configuration, the second matrix deals with the vertical and horizontal shears and the third matrix is the rotation matrix. Thus, estimating the microlenses model parameters $\{\mathbf{c}, d_h, d_v, \theta\}$ gives the microlenses center positions.

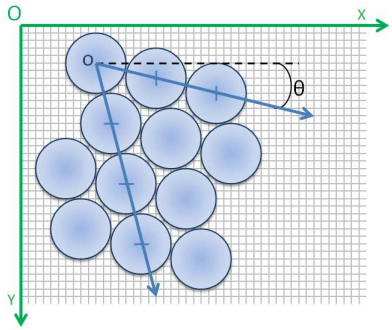
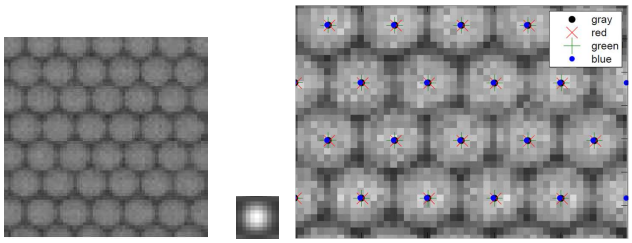


Figure 2. Microlenses projected on the sensor plane in a quincunx configuration. The green and blue axes represent the two coordinate systems. There is a rotational (θ) and a translational ($\mathbf{c} = \mathbf{O} - \mathbf{o}$) offset.

In practice, the subimage centers are computed from a *white image* depicted in Fig. 3-(a), this is an image taken through a white Lambertian diffuser. Actually, the subimage centers $\mathbf{x_i}$ in the raw image are computed as the local maximum positions of the convolution between the *white image* and the mask shown in Fig. 3-(b). Then, given $\mathbf{x_i}$ and the integer positions in the K coordinate system $\mathbf{k_i}$, the model parameters (and consequently $\mathbf{T}$ and $\mathbf{c}$) are estimated via least square errors from the equations $\mathbf{x_i} = \mathbf{T}\mathbf{k_i} + \mathbf{c}$. Thus, the final center positions used in the demultiplexing step are the pixel positions given by $\mathbf{c_i} := round(\mathbf{T}\mathbf{k_i} + \mathbf{c})$. Fig. 3-(c) shows the subimage center estimation obtained with the method described above. Since the raw white image has a Bayern pattern, we have compared the results when estimating the centers using only red, blue or green pixels or using all color pixels at the same time. We observe that the results are essentially the same. Indeed, demosaicing the raw *white image* does not create image cross-talk since the three color channels are the same for all pixels in the center of the subimages.



(a) (b) (c)

Figure 3. (a) Lytro raw image of a white scene. (b) Mask used to locally estimate subimage center positions. (c) Estimated center positions. They coincide when estimated from the red, green, or blue pixels only or from all the pixels in the raw image (gray).

Reordering pixels

In the following, we assume that the raw image has been divided by the white image. This division corrects considerably the vignetting[4] and is enough for our purposes but we refer to [6] for a precise vignetting modeling for plenoptic images. Now, in order to recover the different views, pixels in the raw image need to be reorganized. Pixels under each microlens at the same relative position w.r.t. the microlens center belong to the same view. In particular, pixels are organized as illustrated in Fig. 4. In order to preserve the pixel configuration in the raw image (hexagonal pixel grid due to the quincunx microlens placement) and aliasing, empty spaces are left between pixels on the views as shown in Fig. 5.

Notice that, since the raw image has not been demultiplexed, the views inherit new color patterns. Because of the shift and rotation of the microlenses w.r.t. the sensor, the microlens centers (as well as the other relative positions) do not always correspond to the same color. As a consequence, each view has its own color pattern (mainly horizontal color lines in Lytro) as shown in Fig. 5. In fact, the color pattern of views in odd positions of the matrix are *very similar* to each other, as well as the color patterns of views in even positions. However the color pattern between odd and even positions are not the same. This is because the Lytro camera has a microlens diameter close to an even number of pixels ($\sim$ 10 pixels). As a matter of fact, the color pattern would be the same for all the views (odd and even) if the microlens diameter was an *exact* odd number of pixels.

After demultiplexing, the views could be demosaiced

[4]Light rays hitting the sensor at an oblique angle produce a weaker signal than light rays hitting it with a normal angle. Plenoptic cameras suffer from the main lens vignetting and the microlenses vignetting.

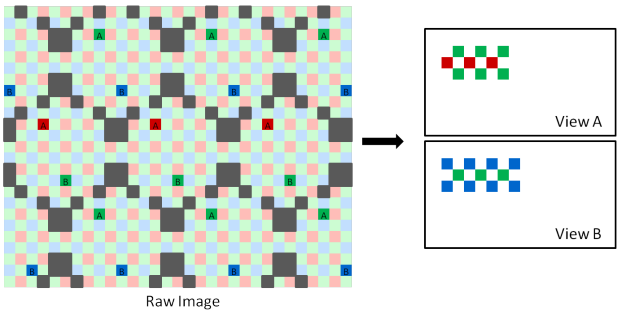


Figure 4. Pixel reordering from the raw image to extract two different raw views. Pixels with the same relative position (same angular information) w.r.t. the subimage centers are stored in the same view. In general, the number of recovered views is equal to the number of pixels per microlens. Color corresponds to sensor color on original Bayer pattern, and is carried over to assembled raw views.

without risking to fuse pixel information from different angular light rays. However, classic demosaicing algorithms are not well adapted to these new color patterns, specially on high frequencies. Instead we simply fill the empty pixels (white pixels in Fig. 4) when the right and left pixels have the same color information. For example, if an empty pixel of the raw data has a green pixel on the right and on the left, then the empty pixel is filled with a green value by interpolation (1D Piecewise Cubic Hermite interpolation). Demosaicing the images with this new color pattern is left for future work but we will show that there is enough information to robustly estimate the disparity from the undemosaiced views.

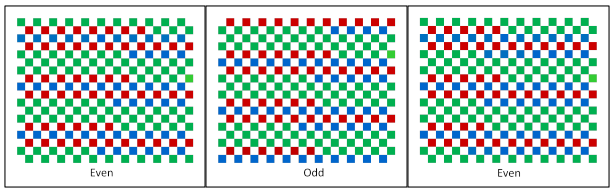


Figure 5. Color patterns of three consecutive vertically-aligned undemosaiced views (even, odd and even positions) for a Lytro camera with a number of pixels per microlens close to an even number ($\sim$ 10 pixels). Color patterns from the views at even positions are very similar while the color pattern at the odd position is significantly different although there are horizontal color stripes too. White pixels correspond to empty pixels (no color information).

Differences with Cho et al. [5] and Dansereau et al. [6] methods

The main difference with the demultiplexing method in [6] is the fact that in their method the raw data of a scene is demosaiced before being demultiplexed. This approach mixes information from different views and, as we will show in the next section, it has dramatic consequences on the disparity estimation step. Besides, the method in [6] estimates the microlenses centers similarly to us but it does not force the center positions to be integer as we do in our global optimization step. Instead, the raw image is interpolated to satisfy this constraint. Even if theoretically this solution should provide a more accurate light-field (microlens center positions have indeed sub-pixel coordinate positions), interpolating the raw data implies again mixing information from different views (a different pixel has information from a light rays with a different angle) which is the same problem pointed out for demosaicing.

The method for estimating the center positions in [5] differs considerably from ours since the centers are found via local maxima estimation in the frequency domain. First, the raw image is demosaiced and converted to gray and the final center positions are the result of fitting the local estimation on a Delaunay triangular grid. Moreover, the second step to render the views is coupled with super-resolution providing views of size 1080×1080 (instead of 328×328, which is the number of microlenses).

4. Disparity Estimation

In this section, we present a new block-matching disparity estimation algorithm adapted to plenoptic images. In particular, we assume that a matrix of views is available (obtained as explained in the previous section) such that the views are horizontally and vertically rectified, i.e., satisfying the epipolar constraint. Therefore, given a pixel in a reference view, its corresponding pixels from the same row of the matrix are only shifted horizontally. Similar reasoning is valid for the vertical pixel shifts among views from the same column of the matrix. Furthermore, consecutive views have always the same baseline a (horizontally and vertically). As a consequence, for each point, its horizontal and vertical disparities with respect to nearest views are equal provided the point is not occluded. In other words, given a point in the reference view, the corresponding point in its consecutive right view is displaced horizontally by the same distance than the corresponding point in its consecutive bottom view is displaced vertically. By construction, the plenoptic camera provides a matrix of views with small baselines, which means that the possible occlusions are small. In fact, each point of the scene is seen from different points of views (even if it is occluded for some of them). Thus, the horizontal and vertical disparity equality is true for almost all the points of the scene.

Since the available views have a color pattern as in Fig. 5, we propose a block matching method so that only pixels in the block having the same color information are compared. We propose to use a similarity measure between

blocks based on the ZSSD (Zero-Mean Sum of Squared Differences). Formally, let I be a reference view of the matrix of views and let I^p and I^q be two views belonging to the same matrix row as I. Let a_p and a_q be the respective baselines with respect to I (multiples of a). Then, the cost function between I^p and I^q at the center (x_0, y_0) of a block B_0 in I is defined as a function of the disparity d:

$$CF^{p,q}_{B_0}(d) = \\ = \frac{\sum_{x,y \in B_0} W(x_p, x_q, y)\Big(I^p(x_p, y) - \overline{I^p} - I^q(x_q, y) + \overline{I^q}\Big)^2}{\sum_{x,y \in B_0} W(x_p, x_q, y)}, \tag{3}$$

where

$$\begin{aligned} x_p &:= x + a_p\, d\,, \\ x_q &:= x + a_q\, d\,. \end{aligned} \tag{4}$$

$\overline{I^p}$ and $\overline{I^q}$ are the average values of I^p and I^q over the block centered at $(x_0 + a_p\, d, y_0)$ and $(x_0 + a_q\, d, y_0)$ respectively and W is a window function defined as follows

$$W(x_p, x_q, y) = G_0(x, y)\ \cdot S(x_p, x_q, y)\,,$$

where G_0 is a Gaussian function centered at (x_0, y_0) and supported in B_0 and S is the characteristic function controlling that only pixels in the block with same color information are taken into account in the cost function:

$$S(x_p, x_q, y) = \begin{cases} 1 & \text{if } I^p(x_p, y) \text{ and } I^q(x_q, y) \\ & \text{have the same color information,} \\ 0 & \text{otherwise.} \end{cases} \tag{5}$$

The cost function is similarly defined when I^p and I^q are views from the same matrix column. In that case we define $y_p := y + a_p\, d$ and $y_q := y + a_q\, d$. Note that if $I_p = I$, then $a_p = 0$ and the cost function defined in Eq. (3) is similar to a cost function for binocular stereovision with the particularity that it is adapted to our specific color pattern.

Now, our algorithm takes advantage of the multitude of views given by the light-field and estimates the disparity through all the lines and columns of the matrix. Let Θ be the set of index-view pairs such that the disparity can be computed horizontally or vertically w.r.t. the reference view I. In other words, Θ is the set of index-view pairs from the same row and column than I. In fact, consecutive views are not considered in Θ since consecutive color patterns are essentially different as explained above for the Lytro camera. Besides, views on the borders of the matrix of views do not capture as much light as the views in the center. So, it is reasonable to only consider the 8×8 or 6×6 matrix of views placed in the center for the Lytro camera. Fig. 6 depicts the pairs of considered images for disparity estimation in a matrix line. In this figure, only the 6×6 matrix is used for disparity estimation.

Finally, given a reference view I the disparity at (x_0, y_0) is given by

$$d(x_0, y_0) = \text{Med}_{(p,q)\in\Theta}\Big\{ \arg\min_d\ CF^{p,q}_{B_0}(d) \Big\}\,, \tag{6}$$

where Med stands for the 1D median filter. This median filter is used to remove outliers that may appear on a disparity map computed for a single pair of views, specially in low-textured areas.

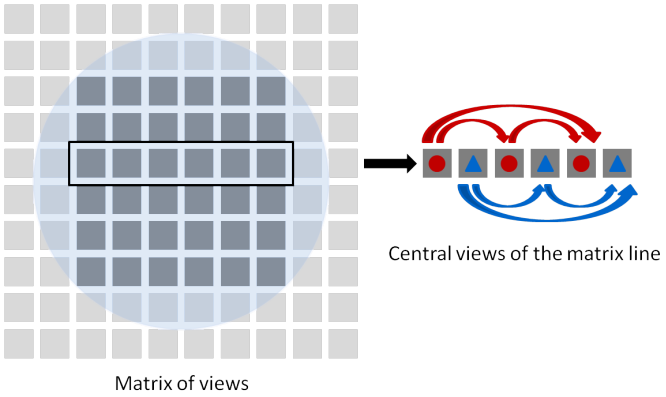


Figure 6. On the left: light-field (matrix of views). Views in the center get more radiance than views of the border of the matrix (pixels coming from the border of the microlenses). Usually the central 6×6 matrix is used. On the right: 6 central views from the same line of the matrix. Odd and even views have different color patterns between them (but very similar patterns between odd views and even views). This is represented with a red circle and a blue triangle. The index-view pairs in Θ corresponding to this matrix line are represented with the red and blue arrows.

Removing outliers

Block-matching methods tend to provide noisy disparity maps when there is a matching ambiguity, e.g., for repeated structures in the images or on poorly textured areas. Inspired by the well-known cross-checking in binocular stereovision [16] (i.e., comparing let-to-right and right-to-left disparity maps), our method can also remove unreliable estimations comparing all possible estimations. Since a large amount of views are available from a light-field, it is straightforward to rule out inconsistent disparities. More precisely, points (x_0, y_0) are considered unreliable if

$$\mathrm{Std}_{(p,q)\in\Theta}\Big\{\arg\min_{d} CF^{p,q}_{x_0,y_0}(d)\Big\} > \varepsilon\,, \qquad (7)$$

where Std stands for standard deviation and ε is the accuracy in pixels. In practice, we consider an accuracy of an eight of a pixel, $\varepsilon = \frac{1}{8}$, and a block size of 13×13.

Sub-pixel disparity estimation

The baseline between the views is small, specially between views with close positions in the matrix. So the disparity estimation for plenoptic images must achieve sub-pixel accuracy. Such precision can be achieved in two different ways: either by upsampling the views or by interpolating the cost function. Usually the first method achieves better accuracy but at a higher complexity burden, unless GPU implementations are used [7]. For this reason, the second method (cost function interpolation) is usually used. However, it has been proved [15] that block-matching algorithms with a quadratic cost function as in Eq. (3) achieve the best trade-off between complexity and accuracy only by firstly upsampling the images by a factor of 2 and then interpolating the cost function. We follow this rule in our disparity estimation algorithm.

Differences with Bishop and Favaro [4] and Tao et al. [17] methods

The closest disparity estimation method for plenoptic images compared to ours is the method presented by Bishop and Favaro [4] but there are several differences between both methods. First, our method properly demultiplexes the views before estimating the disparity, whereas the method in [4] considers full RGB views and proposes an antialiasing filter to cope with image cross-talk artifacts. Then, the energy in [4] (compare Eq. (6) of this paper with Eq. (3) in [4]) considers all the possible pairs of views even if in practice, for complexity reasons, only a subset of view pairs can be considered. In [4], no criteria is given to define such subset of view pairs while a reasonable subset is given by the color pattern in our views. Finally, the proposed energy in [4] considers a regularization term in addition to the data term and the energy is minimized iteratively using conjugate gradients. Unfortunately, a qualitative disparity map between both methods is not possible since the considered input data is not the same.

On the other hand, [17] combines spatial correspondence with defocus. More precisely, the algorithm uses the 4D EPI and estimates correspondence cues by computing angular variance and defocus cues by computing spatial variance after angular integration. Both cues are combined in a MRF global optimization process. This approach exploits nicely the benefits of each cue and the source code has been published by the authors. Nevertheless their disparity estimation method does not take care accurately of the demultiplexing step. Their algorithm not only demosaices the raw image but it stores it using JPEG compression. So, the resulting light-field is affected by image cross-talk artifacts and compression artifacts. In next section we compare our results with the disparity map obtained with their algorithm.

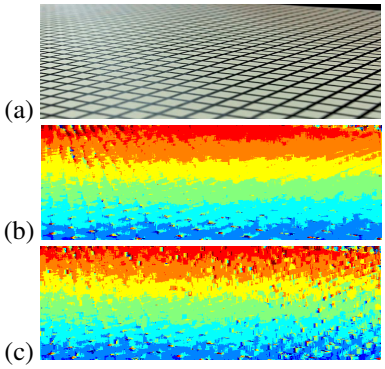


Figure 7. (a) Reference view (for visualization purposes we show the Lytro output). (b) Disparity estimation from two horizontal views without raw image demosaicing (c) Disparity estimation from two horizontal views with raw image demosaicing.

5. Experimental Results

In this section we show the results obtained with our algorithm. First of all we have compared the disparity maps obtained with and without demosaicing the raw image. Intuitivelly one can think that demosaicing the raw image will get better results since more information is available on the views. However we observe in Fig. 7 that this is not the case. The disparity map in Fig. 7-(b) is obtained by minimizing the cost function in Eq. (3) between two consecutive even views of the same row and in the center of the matrix of views with color pattern as in Fig. 5. Similarly Fig. 7-(c) is obtained with the same cost function but the characteristic function is equal to one for all the points since the views are in full RGB.

Fig. 8 shows the demultiplexing and disparity estimation results for a Lytro image (Fig. 8-(a)). Fig. 8-(b) shows one view with the obtained color pattern when demultiplexing directly the raw data without demosaicing it beforehand. In a zoomed version of this image, Fig. 8-(c) shows the result of horizontally interpolating empty pixels when the neighbor pixels have same color information. This simple processing does not cause any image artifact. In Fig. 8-(d,e,f), we also compare the different disparity maps when using our algorithm only with the views of the same column, the views of the same row or all the views in Θ respectively. Note that the computed disparities from views

of the same column (Fig. 8-(d)) are more prone to errors compared to disparities views of the same row (Fig. 8-(e)). This is because the color pattern between two blocks in the same row are more similar than in the same column (color patterns on the views tend to have horizontal color lines). However, it is clear that considering all the views gives the best disparity map (Fig. 8-(f)).

Fig. 9 shows the disparity map obtained with the method by Tao et al. [17] using the same raw data than in Fig. 8. We have used the code and parameters provided by the authors which means that the raw image is demosaiced and compressed (JPEG) before depth is estimated. In this figure the correspondence and defocus cues are also shown with the final depth map. It is obvious that the results obtained with our algorithm are more accurate.

Finally, Fig. 10 shows the disparity map of a "simple" scene of a slanted surface. In this case it is straightforward to compare our results with the known ground-truth for a quantitative evaluation of the method: the RMSE (Root Mean Squared Error) is equal to 0.0179. Fig. 10-(c) shows a horizontal profile plot of the disparity map in blue and the ground-truth profile in red. Results are more accurate for pixels in the center of the image. All in all, our method has been tested on a considerably number of images from Lytro with different conditions and provides more accurate results than state-of-the-art disparity estimation method for plenoptic images (Tao et al. method [17]).

6. Conclusion

Plenoptic cameras are promising tools to expand the capabilities of conventional cameras because they capture the 4D light-field of a scene. However, specific image processing algorithms should be developed to make the most of this new technology. In this work we have addressed the disparity estimation problem for plenoptic images and we have seen that this problem should be studied together with the demultiplexing problem. In particular, we have seen that accurate disparity estimation should be done without demosaicing the raw data to avoid image cross-talk artifacts. Throughout this work, the rendered views have a color pattern obtained when demultiplexing a Bayer raw image which depends on the camera features (number of pixels per microlens, rotation and translation offset between microlenses and sensor, quincunx microlens configuration, etc...). This raises the question of demosaicing the views with the new color patterns which is one of our perspectives for future work.

References

[1] *http://code.behnam.es/python-lfp-reader/.*

[2] E. H. Adelson and J. Y. Wang. Single lens stereo with a plenoptic camera. *TPAMI*, 14(2):99–106, 1992.

[3] T. E. Bishop and P. Favaro. Full-resolution depth map estimation from an aliased plenoptic light field. in computer vision. In *ACCV*, pages 186–200, 2011.

[4] T. E. Bishop and P. Favaro. The light field camera: Extended depth of field, aliasing, and superresolution. *TPAMI*, 34(5):972–986, 2012.

[5] D. Cho, M. Lee, S. Kim, and Y.-W. Tai. Modeling the calibration pipeline of the lytro camera for high quality light-field image reconstruction. In *ICCV*, 2013.

[6] D. G. Dansereau, O. Pizarro, and S. B. Williams. Decoding, calibration and rectification for lenselet-based plenoptic cameras. In *CVPR*, 2013.

[7] V. Drazic and N. Sabater. A precise real-time stereo algorithm. In *ACM Conf. on Image and Vision Computing New Zealand.*, pages pp. 138–143, 2012.

[8] T. Georgiev, G. Chunev, and A. Lumsdaine. Superresolution with the focused plenoptic camera. In *SPIE Electronic Imaging*, 2011.

[9] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen. The lumigraph. In *Conf. on Computer graphics and interactive techniques*, 1996.

[10] C. Kim, H. Zimmer, Y. Pritch, and M. Sorkine-Hornung, A. andGross. Scene reconstruction from high spatio-angular resolution light fields. *SIGGRAPH.*, To appear, 2013.

[11] G. Lippmann. Epreuves reversibles donnant la sensation du relief. *J. Phys. Theor. Appl.*, 7(1),:821–825., 1908.

[12] A. Lumsdaine and T. Georgiev. The focused plenoptic camera. In *ICCP*, 2009.

[13] R. Ng, M. Levoy, M. Brdif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. Technical report, Computer Science Technical Report CSTR, 2(11)., 2005.

[14] F. Perez, A. Perez, M. Rodriguez, and E. Magdaleno. Fourier slice super-resolution in plenoptic cameras. In *ICCP*, 2012.

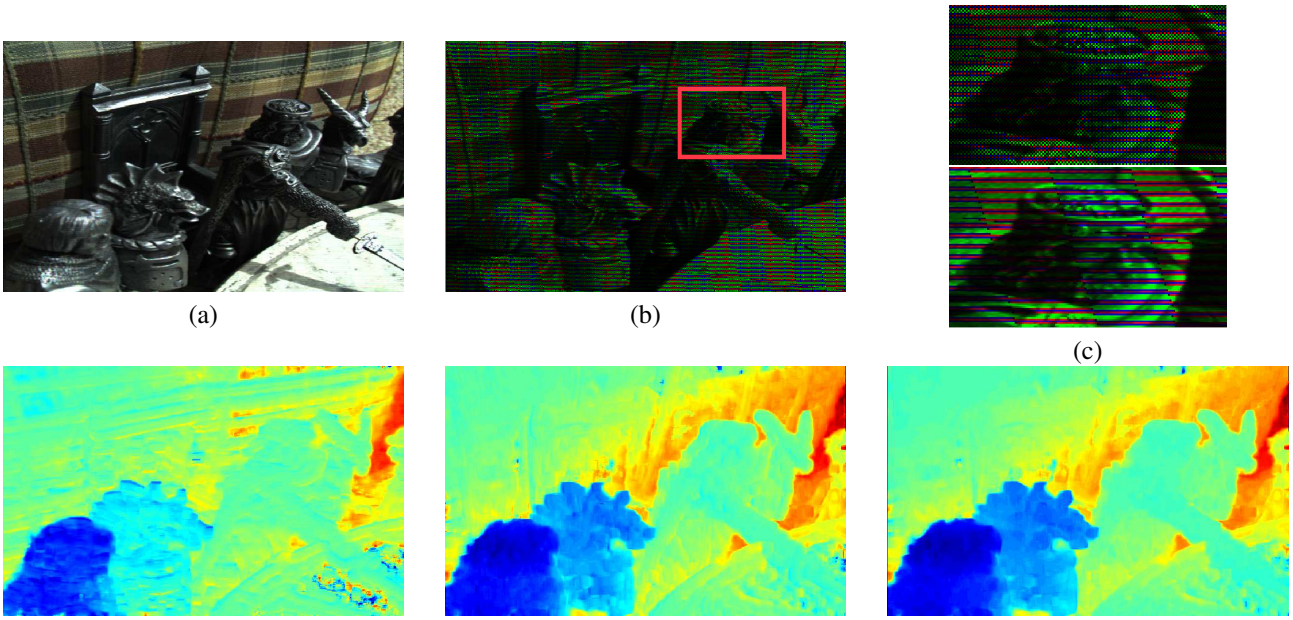
[15] N. Sabater, J. M. Morel, and A. Almansa. How accurate can block matches be in stereo vision? *SIAM Journal on Imaging Sciences*, 4(1):472–500, 2011.

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47(1-3):7–42, 2002.

[17] M. W. Tao, S. Hadap, J. Malik, and R. Ramamoorthi. Depth from combining defocus and correspondence using light-field cameras. In *ICCV*, 2013.

[18] S. Wanner and B. Goldluecke. Variational light field analysis for disparity estimation and super-resolution. *TPAMI*, 2013 (to appear).

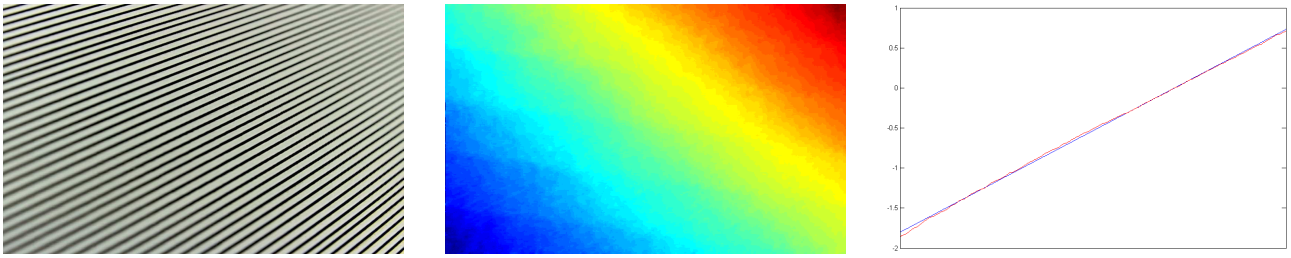
[19] Z. Yu, J. Yu, A. Lumsdaine, and T. Georgiev. An analysis of color demosaicing in plenoptic cameras. In *CVPR*, pages 901–908. IEEE, 2012.



(a) (b) (c) (d) (e) (f)

Figure 8. (a) Lytro image (for visualization purposes). (b) Undemosaiced view. (c) Top: red rectangle in view (b) zoomed. Bottom: same zoom with horizontal interpolation of empty pixels (black). (d) Disparity obtained with horizontally aligned views. (e) Disparity obtained with vertically aligned views. (f) Final disparity obtained with all the views aligned (horizontally and vertically).

Figure 9. From left to right: Disparity map from defocus cues, disparity map from correspondence cues and final regularized disparity map combining defocus and correspondence cues. Disparity values are represented here with gray values.



(a) (b) (c)

Figure 10. (a) Lytro image (for visualization purposes). (b) Disparity map. (c) Plot of the disparity map profile (blue) and ground-truth profile (red).