Int. Conf. on Development and Learning and on Epigenetic Robotics, IEEE ICDL-EpiRob, 2017

Structure from Plenoptic Imaging

Simão Graça Marto[†], Nuno Barroso Monteiro[†‡], João Pedro Barreto[‡] and José António Gaspar[†]
[†] Institute for Systems and Robotics (ISR/IST), Instituto Superior Técnico
Universidade de Lisboa, Portugal
Email: {smarto, nmonteiro, jag}@isr.tecnico.ulisboa.pt
[‡] Institute for Systems and Robotics (ISR/Coimbra)
Universidade de Coimbra, Portugal
Email: jpbar@isr.uc.pt

***Abstract*—Plenoptic cameras allow to recover the structure of a scene from a single image. Recent works on scene structure estimation recover the disparity associated with each pixel but do not consider metric depth. In this work, we propose a methodology for scene metric reconstruction that estimates initial disparity values by analyzing the epipolar plane images. A dense estimate of disparities is obtained by regularization. These disparity estimates are then converted to 3D metric information using the intrinsic parameters of the plenoptic camera. Experiments have been conducted both with simulated and real cameras yielding promising results.**

I. Introduction

The human eye is based in a large retina and a single large lens. In contrast, the compound eye, found for instance in insects, is based in thousands of microlenses, each one conducting light to a number of photoreceptors. Man made plenoptic cameras combine characteristics of both natural designs: each plenoptic camera consists of an artificial compound eye with a large lens placed in front. The artificial compound eye is based on a conventional imaging sensor and an array of microlenses. In other words, the biological equivalent of a plenoptic camera would be a human eye with the retina replaced by an insect eye [1].

In formal terms, common plenoptic cameras acquire light field images or lumigraphs [2], [3], which correspond to 4D samplings of the 7D plenoptic function [4]. A light field image represents light intensity in multiple directions for each point in a plane, while the plenoptic function represents the light intensity spectrum, for all directions at every 3D point, along time.

A light field image can be interpreted as a combined-image acquired by an array of cameras with projection centers placed close to each other, forming an approximately continuous baseline. The array of cameras captures images from slightly different viewpoints and form the so called viewpoint images. A given world feature is projected in each of the viewpoint images at a slightly different location. In viewpoint images, the term disparity is used to refer to the variation in the pixel position of a world feature relative to the variation in the camera considered, i.e., a gradient. Like in stereo, disparity information can be used to reconstruct a scene. Plenoptic cameras have the advantage that the approximately continuous baseline allows computing disparities as gradients of epipolar plane images [5], [6] instead of using feature correspondences.

Prolific research on disparity estimation for reconstruction can be found nowadays, continuously improving the results. However, one finds it very scarce on the aspect of metric reconstruction. This paper proposes a methodology for dense metric reconstruction from plenoptic camera images. In this work, we apply a structure tensor analysis on epipolar plane images to obtain the direction of gradients, from which disparities can be inferred. Interpreting this information with knowledge of the camera intrinsic properties allows metric reconstruction.

The structure of this paper is the following: Section II reviews work on camera geometry and on disparity estimation using plenoptic cameras. Section III describes the back projection model of a plenoptic camera. Section IV describes the estimation of disparity from epipolar plane images and its conversion to 3D metric information. In Section V are described experiments conducted both with simulated and real cameras. Finally, Section VI draws conclusions and proposes future work.

II. Related Work

Depth estimation was the application presented by Adelson and Wang [7] associated to the first prototype of a plenoptic camera. Ng et al. [1] introduced the plenoptic camera as a setup of a single camera with a microlens array between the main lens and the image sensor. Ng et al. were also concerned with recovering depth from focus.

Considering a geometrical study of the lightfield, Dansereau et al. [6] have showed that under a Lambertian scene surface hypothesis, there is a plane of constant value in the light field for each 3D surface point. The orientation of the plane represents the depth of the surface point. The plane orientation can be estimated by gradient operations on the acquired light field. The depth information is estimated essentially at image edge points, and completed by region growing to the complete area of the image.

In Wanner et al. [8], a dense disparity estimation is obtained using a variational approach. Despite using a more conventional terminology, disparity instead of plane orientation, the authors compare the results of regularizing the disparity estimates of one (sensor) vertical and horizontal baselines with a generalized case of multiple baseline directions (multiview

stereo). The multiview stereo was found not bringing significant improvement on the good regularization results obtained.

Monteiro et al. [9] also considered disparity estimates using (sensor) vertical and horizontal directions. In this work, Monteiro et al. [9] considered multiple horizontal and vertical baselines to obtain disparity estimates for the entire lightfield by resorting to an Alternating Direction Method of Multipliers considering periodic boundaries. In this work, there are still missing studies on the estimation of depth from disparities.

Recently, Dansereau et al. [10] presented models, methods and a public toolbox for decoding light field images and calibrating light field cameras. The proposed models characterize the intrinsic parameters of plenoptic cameras, much like the ones found in pinhole cameras, but in a higher dimension since plenoptic cameras also represent the direction of light rays.

In this paper we focus on how to use the intrinsic parameters to promote regularized disparity estimates to a metric structure estimate.

III. Back-Projection Model

The most common light field representation describes the light distribution in the so called two planes parametrization. In the two planes parameterization, the intersection with the first plane defines the position of the ray while the intersection with the second plane defines the direction.

Let us denote the light field in the object space as

$$L_{obj} : (s,t,u,v) \in \mathbb{R}^4 \mapsto I \in \mathbb{R} \tag{1}$$

where (s,t) is a point and (u,v) is a direction. The domain of I can also be $\mathbb{R}^3$ for the case of e.g. RGB color instead of gray levels. The point (s,t) is defined as the intersection of the ray with a plane, at the center of which lies the reference frame of the camera, with z perpendicular to it, pointing towards the scene. The direction (u,v) can be seen as the intersection point of the ray with a plane parallel to the first. This parameterization using a point and a direction is equivalent to a local two plane parameterization with the distance between the two planes fixed at unity.

Consider an arbitrary point $\mathbf{m} = [x,y,z]^T$ in the camera coordinate system. Using the different positions (s,t) of the rays and their directions (u,v), we can define the relation between a world point and the light field in the object space

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} s \\ t \\ 0 \end{bmatrix} + \lambda \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} \ , \ \lambda \in \mathbb{R} \quad . \tag{2}$$

A. Object and Image Spaces

The definition of the light field in the object space, equation (1), does not consider the camera optics. In order to represent the light field in the image space, one needs a transformation of variables.

Dansereau et al. [10] proposed an intrinsic parameters matrix $\mathbf{H}$, which transforms the light field in the image sensor given in pixel and microlens indices $L_{img}(i,j,k,l)$ into the light field in the object space $L_{obj}(s,t,u,v)$

$$\begin{bmatrix} s \\ t \\ u \\ v \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} h_{si} & 0 & h_{sk} & 0 & h_s \\ 0 & h_{tj} & 0 & h_{tl} & h_t \\ h_{ui} & 0 & h_{uk} & 0 & h_u \\ 0 & h_{vj} & 0 & h_{vl} & h_v \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}}_{\mathbf{H}} \begin{bmatrix} i \\ j \\ k \\ l \\ 1 \end{bmatrix} \quad . \tag{3}$$

The indices (i,j) can be seen as selecting a viewpoint, and (k,l) as selecting a pixel within the viewpoint image. More in detail, (i,j) selects the pixel underneath a certain microlens, while (k,l) selects the microlens under which the pixel is [10]. This interpretation is coherent with the observation that pixels emerging from the same (i,j) coordinates intercept at the same point, which can be seen as the center of projection for the given viewpoint.

In Dansereau et al. [10], it is assumed that the horizontal coordinates are completely separate from the vertical ones, which means that, in intrinsic matrix $\mathbf{H}$, only terms that relate horizontal parameters (s,u) and horizontal indices (i,k) or vertical parameters (t,v) and vertical indices (j,l) are non-zero. Therefore, a typical intrinsic matrix $\mathbf{H}$ will have the form presented in equation (3).

B. Intrinsic Matrix from Viewpoint Intrinsic Parameters

Ligthfields can be captured by commercial plenoptic cameras usually built based on arrays of microlenses (lenslets). Alternatively, one can consider camera arrays [11], comprised of several identical cameras arranged in a rectangular array, each pointing in the same direction. Or by acquiring images of a static scene with a single camera moving along positions forming a regular grid [12].

In all cases, obtaining metric values implies modelling and estimating intrinsic parameters of the cameras.

In order to define the intrinsic matrix $\mathbf{H}$ let us analyze the coordinates (s,t) and (u_p,v_p). The coordinates (s,t) correspond to the position of the projection centers of the cameras and (u_p,v_p) correspond to the world points in the plane defined at a distance f from, and parallel to, the plane containing the points (s,t). In a camera array setup, each viewpoint image is obtained pointing in the same direction, and using identical cameras (same intrinsic parameters defined by the intrinsic matrix $\mathbf{K}$). Hence, nothing varies from camera to camera apart from their position (s,t), which does not affect the coordinates (u_p,v_p) because of the local parameterization used to define (u_p,v_p). Therefore, in this setup (s,t) is independent from (u_p,v_p) and can be analyzed separately.

Regarding the (s,t) coordinates, one can assume that the projection centers of the viewpoints are equally spaced in the plane defined by the projection centers of the camera array and the distance between consecutive projection centers is denoted by d_s and d_t, which leads to

$$\begin{bmatrix} s \\ t \\ 1 \end{bmatrix} = \begin{bmatrix} d_s & 0 & s_0 \\ 0 & d_t & t_0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} i \\ j \\ 1 \end{bmatrix} \quad , \tag{4}$$

where s_0 and t_0 are defined such that the center of the plane lies in coordinates $(0,0)$, $s_0 = -d_s(w+1)/2$ and $t_0 = -d_t(h+1)/2$, with w and h defining the number of cameras in the camera array in width and height, respectively.

Regarding the (u_p, v_p) coordinates, one can use the formula for the pinhole camera to describe the relationship between a world point and the pixel coordinate. This is, essentially, projecting the pixels from the image plane onto the plane which is f units of distance away from the plane containing the array, i.e. $[k\ l\ 1]^T = \mathbf{K}[u_p\ v_p\ f]^T$. Inverting the projection equation and combining with equation (4), the intrinsic matrix $\mathbf{H}$ is defined as

$$\mathbf{H} = \begin{bmatrix} d_s & 0 & 0 & 0 & s_0 \\ 0 & d_t & 0 & 0 & t_0 \\ 0 & 0 & & & \\ 0 & 0 & & \mathbf{K}^{-1} & \\ 0 & 0 & & & \end{bmatrix} \quad . \tag{5}$$

In this work, the parameterization of the lightfield in the object space is defined in terms of a position in a plane, and the direction of the ray. As mentioned previously, this is equivalent to a local two plane parameterization with the planes set at a unit distance apart. This means that by defining f to be one, the parameterization (s,t,u_p,v_p) here defined for a camera array, becomes equivalent to the one we are using, (s,t,u,v).

IV. Reconstruction

Unlike a standard camera, a single feature in a scene has multiple projections on the light field. Plenoptic cameras observe rays which do not converge on the same point. In other words, sampled rays do not have a single projection center. Using the intrinsic matrix $\mathbf{H}$, mapping the rays (i,j,k,l) to rays (s,t,u,v), allows finding the constraint that a collection of rays corresponding to a single feature must follow. The feature location can therefore be estimated from the multiple rays passing through the feature.

A. Relationship between position in space and pixel indices within a light field

Rewriting equation (2) as $[x\ y]^T = [s\ t]^T + z\,[u\ v]^T$ and replacing the light field on the object space by the light field on the image sensor using the mapping defined in equation (3) one obtains:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \mathbf{H}_{ij}^{st} \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{H}_{kl}^{st} \begin{bmatrix} k \\ l \end{bmatrix} + \mathbf{h}_{st} + z \left(\mathbf{H}_{ij}^{uv} \begin{bmatrix} i \\ j \end{bmatrix} + \mathbf{H}_{kl}^{uv} \begin{bmatrix} k \\ l \end{bmatrix} + \mathbf{h}_{uv} \right) \quad , \tag{6}$$

where we follow the notation $\mathbf{H}_{(\cdot)}^{(\cdot)}$ to represent a submatrix of $\mathbf{H}$. For instance, $\mathbf{H}_{ij}^{st}$ selects the first two columns, denoted by the subscript ij, and the first two lines, denoted by the superscript st.

More precisely, the intrinsic matrix $\mathbf{H}$ is partitioned in four 2×2 diagonal sub-matrices and two 2×1 vectors

$$\mathbf{H}_{ij}^{st} = \begin{bmatrix} h_{si} & 0 \\ 0 & h_{tj} \end{bmatrix},\ \mathbf{H}_{kl}^{st} = \begin{bmatrix} h_{sk} & 0 \\ 0 & h_{tl} \end{bmatrix},\ \mathbf{h}_{st} = \begin{bmatrix} h_s \\ h_t \end{bmatrix},$$

$$\mathbf{H}_{ij}^{uv} = \begin{bmatrix} h_{ui} & 0 \\ 0 & h_{vj} \end{bmatrix},\ \mathbf{H}_{kl}^{uv} = \begin{bmatrix} h_{uk} & 0 \\ 0 & h_{vl} \end{bmatrix},\ \mathbf{h}_{uv} = \begin{bmatrix} h_u \\ h_v \end{bmatrix} . \tag{7}$$

Let us consider that the feature's position $\mathbf{m}$ is constant and let us compute the derivative of equation (6)

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \mathbf{H}_{ij}^{st} \begin{bmatrix} \partial i \\ \partial j \end{bmatrix} + \mathbf{H}_{kl}^{st} \begin{bmatrix} \partial k \\ \partial l \end{bmatrix} + z \left(\mathbf{H}_{ij}^{uv} \begin{bmatrix} \partial i \\ \partial j \end{bmatrix} + \mathbf{H}_{kl}^{uv} \begin{bmatrix} \partial k \\ \partial l \end{bmatrix} \right) \quad . \tag{8}$$

Considering both equations separately, and solving each for z, one obtains a relation between the depth of the world feature $\mathbf{m}$ and the disparity represented by the gradients $\frac{\partial k}{\partial i}$ and $\frac{\partial l}{\partial j}$:

$$z = -\frac{h_{si} + h_{sk}\frac{\partial k}{\partial i}}{h_{ui} + h_{uk}\frac{\partial k}{\partial i}} \quad \vee \quad z = -\frac{h_{tj} + h_{tl}\frac{\partial l}{\partial j}}{h_{vj} + h_{vl}\frac{\partial l}{\partial j}} \quad . \tag{9}$$

In this work, we assume that the intrinsic parameters in $\mathbf{H}$ are the same in the horizontal and vertical directions, i.e. $h_{si} = h_{tj}$, $h_{sk} = h_{tl}$, $h_{ui} = h_{vj}$ and $h_{uk} = h_{vl}$, such that both derivatives yield in general the same value.

Concluding, the position of a point, $[x\ y\ z]^T$ can be obtained by estimating the gradients $\frac{\partial k}{\partial i}$ and $\frac{\partial l}{\partial j}$, applying equation (9) to calculate the depth z, and finally using equation (6) to obtain the (x,y) coordinates.

B. The proposed algorithm

In this section we describe the complete reconstruction algorithm, starting from a light field in the image space and resulting in a depth map. Prior to running the proposed algorithm, it is assumed that one runs the decoding of raw images to a light field image and that the plenoptic camera is calibrated [10].

The complete reconstruction algorithm encompasses three main steps, namely disparity estimation, disparity regularization and conversion to metric structure.

1) Disparity Estimation: A light field in the image space, $L_{img}(i,j,k,l)$ can be represented as a hypercube, where each slice with (i,j) constant is a viewpoint image. These images are similar in their properties, except each was taken from a slightly different position, indicated by their (i,j) coordinates.

Slices made by fixing (i,k) or (j,l) are vertical or horizontal epipolar plane images (EPIs), respectively (e.g. figure IV-B1). EPIs are very interesting for the purposes of reconstruction because they show the effect of parallax. An EPI is composed of the same row of pixels taken from a row of viewpoints, and stacked on top of each other [5]. This way, they show how the captured image shifts as the center of projection (of a viewpoint) pans left and right (or up and down).

Within these EPIs, the gradient of the image will be extracted. The direction perpendicular to this one represents

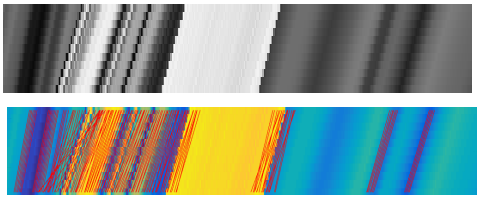


Fig. 1. Above: An epipolar plane image, taken from a synthetic light field, made using Matlab's VR tools. Below: The same EPI, with a different color scheme, and red lines marking detected gradients. The color scheme still represents image intensities, but it was chosen to make the gradients more visible.

the direction with least change in an image, which, in most cases, corresponds to a direction along which all the pixels correspond to the same real-world feature. Obtaining the gradient $\frac{\partial k}{\partial i}$ or $\frac{\partial l}{\partial j}$ that matches this direction is the first step in the algorithm.

These EPIs show the same line from all viewpoints, arranged side by side. Figure IV-B1 shows a typical EPI. In it several edges can be seen. These edges correspond to areas of the image with large gradients, which allow for useful epipolar gradient information to be extracted.

The gradient on the EPI is calculated using a structure tensor. The values of I_i, I_k, representing the gradient of the image along i and k, are calculated using a Sobel operator. The local structure tensor, S_0, is then formed for each pixel in the EPI according to

$$S_0 = \begin{bmatrix} I_i^2 & I_i I_k \\ I_i I_k & I_k^2 \end{bmatrix} \quad . \tag{10}$$

A typical structure tensor is a convolution of a window function with this S_0 matrix.

The final goal of the first part of this algorithm is to obtain a structure tensor $S(k,l)$ for every viewpoint pixel k,l, that is, a 2D array of structure tensors, indexed by viewpoint pixel. The problem of occlusion will not be considered, and whenever a real-world feature gets occluded, errors will occur.

At first, the matrix S_0 associated with every EPI will be averaged along i, eliminating this variable, and producing a 1D vector of structure tensors, hereby referred to as S_e. This matrix S_e will be calculated for every possible EPI, both vertical and horizontal, and on all color channels. The value of S will result of adding up every one of these S_e vectors on the location in S that they refer to. This method will naturally give preference to stronger gradients, preventing areas with weak gradients dominated by noise from disturbing the final results.

From the structure tensor, it is possible to extract the gradient direction across the EPI, and also, from the difference between the eigenvalues, a confidence measure, telling us how accurate the disparity estimation is in a given location [13].

For example, in figure 1, a structure tensor like S_e is produced for the EPI of figure IV-B1, and lines were traced along the edges detected where the confidence measure is above a certain threshold. As can be seen, most of them appear to match, although some of them are completely incorrect. However, by adding up the values of S_e obtained for every EPI in the line or row of S they correspond to, most errors should be decreased. The results of applying the algorithm on the full light field image from which figure IV-B1 was taken will be discussed in section V-A.

2) Disparity Regularization: In areas where the confidence measure does not meet a certain treshold, the disparity values were ignored. To have data pertaining to the whole viewpoint area, the in-painting algorithm published by John D'Errico in [14] was used (method number 4). To deal with the noise present in the results, a total variation method by Gabriel Peyre [15] is used. Both of these methods were applied to the resulting disparity map, a 2D image.

The total-variation method used finds the values y that minimize a function of the form $E(x,y) + \lambda J(y)$, where $E(x,y)$ measures the difference between the original image x and y, $J(y)$ is the total variation of y, and λ is the regularization parameter. The regularization method used, was based on the one defined by Chambolle in [16]. In that article, they define $E(x,y) = \|y-x\|^2/2$, and $J(y) = \|\nabla y\|$, where the gradient ∇y is defined with a forward difference method.

3) Metric Structure: The disparity estimation and regularization steps organize disparities into an image format, a dense disparity map. Consequently, one may obtain the resulting point cloud as a depth map represented also as an image in the coordinates of the viewpoint images, i.e. (k,l). Converting the disparity map into a depth map, or point cloud map, involves computing the location $[x\ y\ z]^t$ for every pair of values (k,l), for a fixed pair (i,j), as detailed in section IV-A.

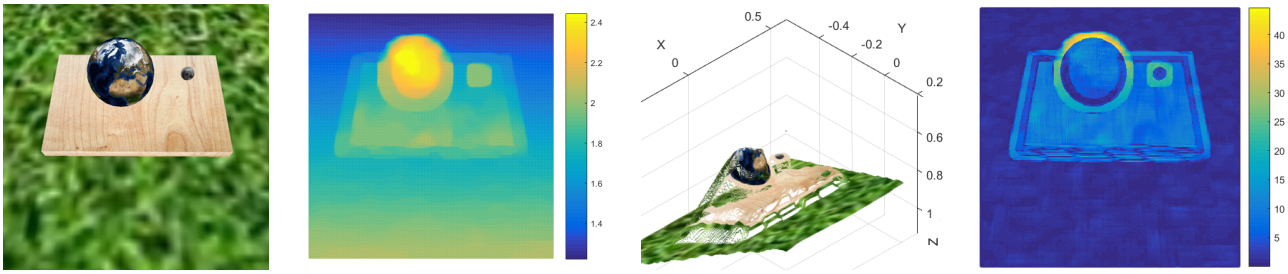
V. Experiments

Several tests were conducted to test the proposed methodology, both with synthetic images, as well as with real images taken by a plenoptic camera.

A. Synthetic Data

A synthetic scenario was created in order to create synthetic 4D light field images. In our case, the scene was built in the Virtual Reality Markup Language (VRML) and the images were captured using the Matlab Virtual Reality (VR) toolbox. Given the VRML scenario, an array of images is captured, and then arranged into a light field. The intrinsics matrix is obtained using J. Y. Bouguet's calibration toolbox [17], since Matlab VR tools do not clarify what the intrinsics are when an image is captured.

1) VRML Scene: The VRML scene is composed of a horizontal plane (i.e. normal to the y direction, considered as up) with a grassy texture upon which rests a box with a wooden texture. The center of the coordinate system corresponds to the center of the box, which has dimensions $0.6 \times 0.04 \times 0.4$ in the x, y and z directions, respectively. Floating above this box are two spheres. One sphere is textured to look like the Earth, and has its center at coordinates $(-0.05, 0.15, 0)$ and radius 0.1. The other is textured with the moon's surface, centered on $(0.15, 0.15, 0)$ and radius 0.02.



(a) Central viewpoint (b) Disparity (c) Reconstruction (d) Relative reconstruction error

Fig. 2. Synthetic light field depth reconstruction. Central viewpoint image, size 378×378 pixels, of a 11×11 set of viewpoint images (a). Disparity map in pixel (b). Reconstructed point cloud in metric units (c). Relative depth reconstruction error in percentage (d).

2) Capturing an image: The synthetic light field images were made by combining several viewpoint pictures, captured from positions forming an array. The distance between each center of projection is the same in both directions, namely $d_x = d_y = 0.004$.

The intrinsic matrix for each viewpoint was obtained by performing a calibration routine in Matlab [17], after taking several pictures of a checkerboard within this virtual world. Note that Matlab tends to represent vectors in row form, whereas we represent them in column form. Because of this, the resulting intrinsics matrix needs to be transposed to become equivalent to K. This is the H matrix that is obtained, by applying equation (5):

$$\mathbf{H} = \begin{bmatrix} 0.0040 & 0 & 0 & 0 & -0.024 \\ 0 & 0.0040 & 0 & 0 & -0.024 \\ 0 & 0 & 0.0029 & 0 & -0.55 \\ 0 & 0 & 0 & 0.0029 & -0.55 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix} . \quad (11)$$

3) Results: A light field was captured in the described settings. Figure 2(a) represents the central viewpoint of this light field. The measured disparity values are also represented in figure 2(b). The (x, y, z) coordinates corresponding to each point were calculated as described in the previous section, and all the points were arranged into a point cloud, which is represented from some points of view in figure 2(c).

The accuracy of the algorithm was evaluated by comparing with the ground truth obtained from the information used to define the scene in the VRML file. The results are presented in Figure 2(d). The relative error is below 10% for 88% of the image, and the RMS of the error is 0.0574, where the globe has a diameter of 0.2. If the grass background plane is ignored, the RMS becomes 0.0841.

The resulting point cloud approximates well the scene, especially large planar features such as the grassy plane. However, more detailed areas are significantly distorted. Notably, areas where occlusion occurs, such as in the edges of the spheres, "ramps" seem to form, connecting the sphere to the grassy background. This is a result of averaging in the i or j direction, which averages structure tensors of two distinct features, significantly separated in space.

B. Real Data

The light field image used, captured by a Lytro model F01, is represented in figure 3(a). The scene is composed of two checkerboard textured cubes, stacked on top of each other. Behind the cubes, to the left, is a checkerboard, squares are $35mm$ in size, and to the right is a computer monitor. As expected, in the real light field image there are problems such as outer viewpoints being darker than central ones, having too low contrast edges, dead pixels, and other general imaging noise.

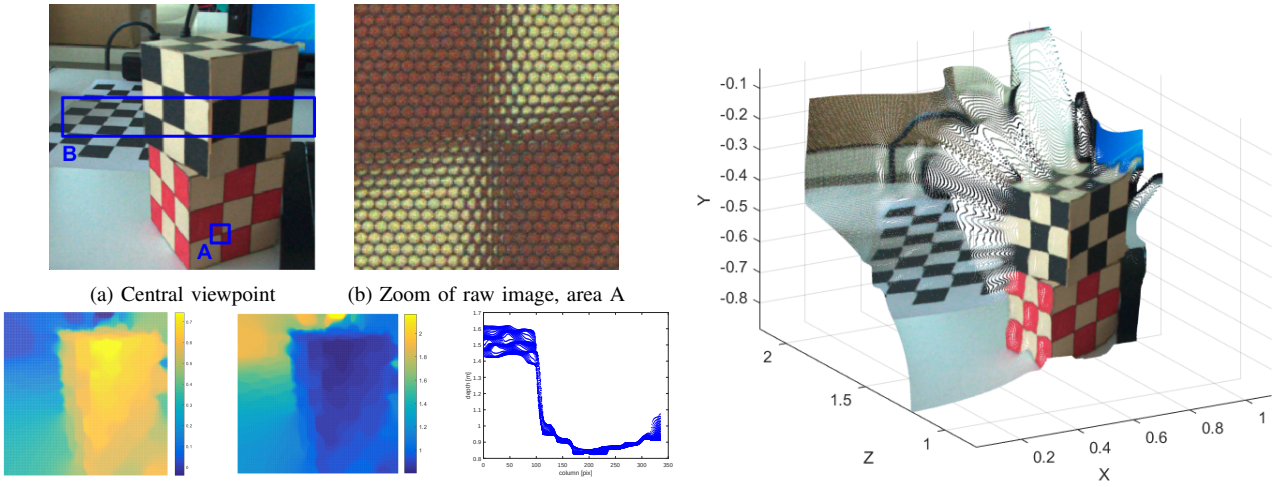
The decoding of raw images to a light field in the image space is a process automated by the toolbox of Dansereau et al. [10] which works perfectly well with our cameras. The intrinsics calibration required some repetitions of the process, namely to add novel calibration images with checkerboard patterns. The value for the confidence measure cut-off threshold, and the regularization parameter were adjusted to improve the results. A median filter was also applied to the image to remove dead pixels. In addition, a Gaussian filter with $\sigma = 3$ was applied on the structure tensor to smooth the observed noise.

In order to avoid darker viewpoints, due e.g. to vignetting, a "ring" composed of the outermost viewpoints was rejected, and the remaining ones were normalized so that all have 0 as minimum intensity value, and 1 as maximum.

The resulting depth map is presented in figure 3. It correctly displays lower depth values in regions of the scene that are closer, such as the cube, and higher in regions that are further, such as the monitor. Moreover, basic geometry is preserved.

To help visualize the reconstruction, a region from the central viewpoint, marked in figure 3(a), was extracted from the point cloud and viewed from above in figure 3(e). This figure allows to distinguish between the chess pattern on the background and the two faces of the cube on the foreground.

In order to quantify the reconstruction error, points have been extracted from the point cloud corresponding to the three visible faces of the cube at the top (black squares). The points from each face were then fitted into a plane using a least squares method. The RMS of the distances between each point and the fitted plane was then obtained, and divided by the length of the side of the cube in the point cloud. The measurement errors were 3.3%, 1.4% and 1.1% for the left,



(a) Central viewpoint (b) Zoom of raw image, area A

(c) Estimated disparity (d) Estimated depth (e) Depth for pixels in B

(f) Reconstructed point cloud

Fig. 3. Real light-field image depth reconstruction. The central viewpoint (a), size 378×379 pixels, of a 11×11 viewpoints set, has superimposed two rectangular areas, A and B, to support the explanation of the next sub-figures. The raw image zoomed in area A shows the imaging effect of the micro-lenses (b). The estimated disparities are regularized on the complete image area (c). Depth is computed given the disparities at all pixel locations (d). The depth values for the pixels in area B are shown in (e) sorted by column number. Using the intrinsic parameters a point cloud is generated from the depths estimated for all the pixels (f).

right and top faces, respectively.

VI. Conclusion

In this article, a dense methodology for reconstructing a scene from a light field image was presented. The proposed methodology is quite successful at reconstructing 3D data, structure, despite some distortion being visible. Geometry is mostly kept in a recognizable form after reconstruction, and in the synthetic images, the dimensions of the objects were the same as defined in the scene VRML file.

In the case of the real image, as expected, the calibration uncertainty affects the results. As with the calibration of standard, pin hole like cameras, care has to be taken about setting the calibration data, i.e. the poses of the checkerboard patterns shown to the camera. Nevertheless, we found it feasible to create coherent scene reconstructions. The results obtained are encouraging.

As future work, we consider repeating the calibration multiple times, while indicating the user positions of the checkerboard likely to help calibration.

Acknowledgment

This work was supported by the Portuguese Foundation for Science and Technology (FCT) project [grant numbers UID/EEA/50009/2013, and PD/BD/105778/2014].

References

[1] R. Ng, M. Levoy, M. Brédif, G. Duval, M. Horowitz, and P. Hanrahan, "Light field photography with a hand-held plenoptic camera," *Computer Science Technical Report CSTR*, vol. 2, no. 11, pp. 1–11, 2005.

[2] M. Levoy and P. Hanrahan, "Light field rendering," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 31–42.

[3] S. J. Gortler, R. Grzeszczuk, R. Szeliski, and M. F. Cohen, "The lumigraph," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 43–54.

[4] E. H. Adelson and J. R. Bergen, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of Technology, 1991.

[5] R. C. Bolles, H. H. Baker, and D. H. Marimont, "Epipolar-plane image analysis: An approach to determining structure from motion," *International Journal of Computer Vision*, vol. 1, no. 1, pp. 7–55, 1987.

[6] D. Dansereau and L. Bruton, "Gradient-based depth estimation from 4d light fields," in *Circuits and Systems, 2004. ISCAS'04. Proceedings of the 2004 International Symposium on*, vol. 3. IEEE, 2004, pp. III–549.

[7] E. H. Adelson and J. Y. A. Wang, "Single lens stereo with a plenoptic camera," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 2, pp. 99–106, 1992.

[8] S. Wanner and B. Goldluecke, "Variational light field analysis for disparity estimation and super-resolution," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 3, pp. 606–619, 2014.

[9] N. B. Monteiro, J. P. Barreto, and J. Gaspar, "Dense lightfield disparity estimation using total variation regularization," in *International Conference Image Analysis and Recognition*. Springer, 2016, pp. 462–469.

[10] D. G. Dansereau, O. Pizarro, and S. B. Williams, "Decoding, calibration and rectification for lenselet-based plenoptic cameras," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 1027–1034.

[11] B. Wilburn, N. Joshi, V. Vaish, E.-V. Talvala, E. Antunez, A. Barth, A. Adams, M. Horowitz, and M. Levoy, "High performance imaging using large camera arrays," in *ACM Transactions on Graphics (TOG)*, vol. 24, no. 3. ACM, 2005, pp. 765–776.

[12] S. Wanner, S. Meister, and B. Goldluecke, "Datasets and benchmarks for densely sampled 4d light fields." in *VMV*. Citeseer, 2013, pp. 225–226.

[13] J. Bigun, "Optimal orientation detection of linear symmetry," 1987.

[14] D'Errico. inpaint_nans. Matlab Central File Exchange. Accessed: 2017-03-30.

[15] G. Peyre. Toolbox sparse optimization. Matlab Central File Exchange. Accessed: 2017-03-30.

[16] A. Chambolle, "An algorithm for total variation minimization and applications," *Journal of Mathematical Imaging and Vision*, vol. 20, pp. 89–97, 2004.

[17] J. Y. Bouguet. (2008) Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. Accessed: 2017-03-30.