# A prototype of a real-time single lens 3D camera

J.P. Lüke, J.G. Marichal-Hernández, F. Rosa and J.M. Rodríguez-Ramos
Department of Fundamental and Applied Physics, Electronics and Systems
University of La Laguna
Avenida Francisco Sánchez s/n, 38200, La Laguna (España)
{jpluke, jmariher, frosa, jmramos}@ull.es

***Abstract*—This paper describes a prototype of the CAFADIS camera. The prototype consists of three parts: acquisition, processing and display. The images are captured with a plenoptic camera obtaining multiview information of the scene used afterwards to estimate depth information. The plenoptic images are processed with novel algorithms running on a multi-GPU computer that achieves real time performance. Finally the obtained depth and all-in-focus images are displayed on an autostereoscopic display.**

***Plenoptic camera; 3D reconstruction; graphics processing units; real-time; lightfield***

## I. INTRODUCTION

The 3D reconstruction has been a very active research field for many years. We deal with the problem obtaining 3D information of a scene from a set of 2D views. This can be achieved using several sensors, generally two or more cameras. The matching between the image points of the different views has to be estimated and some geometric calculations have to be done to obtain depth information.

The CAFADIS camera combines several elements in order to obtain the 3D information of a scene in real time. The multiple views of the scene are obtained using a plenoptic camera configuration [1, 2]. This allows acquiring several 2D images of the scene from different view points with a single lens and single body device.

The captured plenoptic image must be processed in order to extract the depth information. This is done with novel processing techniques to obtain super-resolved depth and all-in-focus images [3]. Such methods solve one of the main drawbacks of the plenoptic camera system which consists in the low spatial resolution of the final images [2,4]. This makes it possible to achieve, with sensors currently on the market, final images of acceptable resolution for today video standards. However, since the required computational power is high in order to achieve real time performance, our algorithms run on massively parallel hardware, namely graphics processing units (GPUs) [5].

The implementation is done on a multi-GPU computer using CUDA [6] and splitting the problem in parts that are scheduled on several GPUs [7]. The resulting color+depth image stream can be displayed on an autostereoscopic Philips WOWvx 3DTV.

The rest of this paper is organized as follows: In section II, some ideas about the image acquisition configuration will be given. Section III covers the processing part, and section IV, discusses the displaying part. Some results will be shown in section V. Finally, some concluding remarks can be found in section VI.

## II. ACQUISITION

The CAFADIS camera uses a plenoptic camera configuration for single exposure multiple views acquisition. A plenoptic camera consists basically in a modification of a conventional camera, where a microlenses array has been inserted at a distance $F$ in front of the sensor. Each one of those tiny microlenses has a focal length of $F$. Such a lens configuration allows for the capture of the radiance information of the scene on the sensor [2]. Each microlens samples a set of ray directions at a single spatial point on the main lens. In order to reach the maximum angular resolution the *f*-number of the microlenses should be the same as the *f*-number of the main lens. The lightfield, $L$, inside the camera can be parameterized with two planes as shown in figure 1. The first plane is related to the main lens and the second one is related to the microlens array so that $L(\boldsymbol{x},\boldsymbol{u})$ represents the light traveling through the main lens at $\mathbf{u}=(u_1,u_2)$ and hitting the microlens plane at $\mathbf{x}=(x_1,x_2)$.

The plenoptic camera captures a sampled version of $L(\boldsymbol{x},\boldsymbol{u})$ which can be used to synthesize photographs focused at different depths [2, 4]. The microlens array behind the main lens redirects the light that impinges with a certain direction to the adequate position on the sensor. Therefore each pixel behind a microlens corresponds to a direction of the incoming light rays. This fact leads to a spatial-angular resolution tradeoff because part of the sensor is now used for angular sampling instead of spatial sampling [8].

The prototype is built on an Imperx-2M30-LC camera modified as previously described. This camera has a sensor with a resolution of 1600×1200 pixels and a maximum frame rate of 33 fps. The data is transferred to the computer through a Camera Link interface connected to a DALSA Coreco x64-Full frame grabber.

A microlens array of 116×87 was placed in front of the sensor so that each microlens covers 13.7 pixels. The resulting plenoptic image is rectified and interpolated for obtaining an integer number of pixels behind the microlenses. The size of the resulting lightfield is then 116×87×11×11, which is

This work was funded by national R&D Program (Project AYA-13075) of the Ministry of Science and Innovation, by the Agencia Canaria de Investigación, Innovación y Sociedad de la Información (ACIISI) and by the European Regional Development Fund (ERDF).

equivalent to 11×11 images from different view points, each of 116×87
pixels.[8]

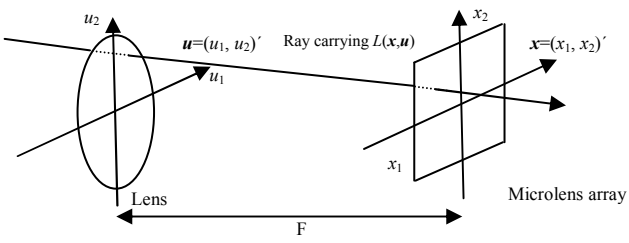


Figure 1. Two plane parameterization of the lightfield (adapted from [2]).

## III. PROCESSING

In this section some ideas about the processing algorithms will be given. The processing step consists in synthesizing photographs focused at different depths, which is known as focal stack transform. Then a focus estimator is used to obtain a depth map with the belief propagation algorithm [9].

One of the drawbacks of the plenoptic camera configuration is the low spatial resolution of the final images [2, 4]. This problem has been addressed using super-resolution techniques, which consist in recovering a high resolution image from multiple low resolution images from a given scene. Such techniques have been recently developed for plenoptic images obtaining considerably higher spatial resolution in the final images [3, 10, 11]. In this case, the ideas developed in [3] will be used to construct a super-resolved depth and all-in-focus image from the captured plenoptic images.

The depth reconstruction is based on the super-resolution focal stack transform which is used to obtain the color and variance focal stacks. The second is then used as cost function for the belief propagation algorithm in order to obtain a depth map of the scene. The color focal stack and the depth map are finally used to construct an all-in-focus image.

### A. Super-resolved focal stack transform

The first part of the algorithm is the super-resolved focal stack transform [3]. It consists on the synthesis of several photographs focused at different depths using the plenoptic images. In order to synthesize a photograph focused at a depth $\alpha F$ the following integral has to be computed:

$$\mathcal{P}_\alpha[L](\mathbf{x}) = \frac{1}{\alpha^2 F^2}\int L\left(\mathbf{u}\left(1-\frac{1}{\alpha}\right)+\frac{\mathbf{x}}{\alpha},\mathbf{u}\right)d\mathbf{u}. \qquad (1)$$

Then the focal stack transform is defined as $\mathcal{S}[\boldsymbol{L}](\mathbf{x},\alpha)=\mathcal{P}_\alpha[\boldsymbol{L}](\alpha\mathbf{x})$, that is the set of such photographs computed at all depths.

However, since plenoptic images are a discrete version of $L$ the integral in (1) has to be discretized and $L$ has to be interpolated obtaining the discrete focal stack transform. Assuming that the plenoptic camera has $n \times n$ microlenses and each one generates an image of $m \times m$ pixels the output images of conventional focal stack algorithms will be a set of images of size $n \times n$ [2, 4]. This is because the integral in (1) is only computed for discrete $\mathbf{x}$ coordinates for which exists a sample on the axis in the discrete version of $L$. However, the discrete set of $\mathbf{x}$ coordinates for which the integral can be computed with the rays captured in the plenoptic image can be extended with fractional values that meet certain conditions. Assuming integer coordinates for the sampled versions of $\mathbf{x}$ and $\mathbf{u}$, the discrete integral can be computed for fractional coordinates of $\mathbf{x}$ using only the data placed at integer coordinate position of the sampled version of $L$. This gives the discrete super-resolution focal stack transform that generates final images with more spatial resolution than conventional algorithms. The discrete photography integral in (1) can be written for the discrete super-resolved focal stack transform as follows:

$$\mathcal{P}_\alpha[\boldsymbol{L}](\mathbf{x}) = \frac{1}{|\boldsymbol{U}|}\sum_{\boldsymbol{u}\in\boldsymbol{U}}\boldsymbol{L}\left(\mathbf{u}\left(1-\frac{1}{\alpha}\right)+\frac{\mathbf{x}}{\alpha},\mathbf{u}\right). \qquad (2)$$

with:

$$\boldsymbol{U} = \left\{\boldsymbol{u}\in\boldsymbol{Z} \mid \mathbf{u}\left(1-\frac{1}{\alpha}\right)+\frac{\mathbf{x}}{\alpha}\in\boldsymbol{Z}\right\} \qquad (3)$$

The condition in (3) restricts the set of values of x and $\alpha$ for which the integral can be computed to a discrete set of values. However, since under this condition $\mathbf{x}$ also can take fractional values the resolution of the final image will be increased. Another restriction that is applied in this case is that $|U| \geq 2$, because for computing a variance focal stack more than two rays for each pixel are needed.

Restricting the collection of slopes to $\{\alpha = \Delta u/\Delta x |$ $0<|\Delta x|<s$, $\Delta u=s$, $\Delta u$, $\Delta x\in Z$ with $s$ a prime number$\}$ the generated super-resolved focal stack has $\Delta u\ n - \Delta u + 1$ pixels in each image dimension with $n=2r+1$ and $m=2s+1$. That gives a final resolution of approximately the sensor resolution divided by 4 and $m$-3 planes can be generated. More detailed explanations of the algorithm can be found in [3].

### B. Belief propagation algorithm

The belief propagation algorithm is a well known algorithm used in multi-stereo [9]. It is used to obtain a depth map from the super-resolved focal stack. The input of this algorithm is a cost function that has to be minimized and smoothed. In this case, the variance focal stack is used [3]. The variance focal stack is based on the photoconsistency assumption which states that the radiance of rays from a 3D point over all directions is approximately the same. It consists in computing the variance of the rays instead of the average in (2). If the point is in focus the rays from all directions should be similar and the variance is expected to be small, but if the point is out of focus the value of the variance is expected to be high.

The belief propagation algorithm also introduces the assumption of smooth surfaces. The energy function to be minimized is composed of a data term $E_d$ and a smoothness term $E_s$, $E = E_d+\lambda E_s$, where the parameter $\lambda$ measures the

relative importance of each term. The data term is the sum of the per-pixel data costs, $E_d = \Sigma_p\ c_p(d)$, where $c_p(d)$ is the variance measure for pixel $\boldsymbol{p}$ in the super-resolved image and $d$ is a specific depth, that is related to a distance plane in the focal stack. The smoothness term can be written as $E_s = \Sigma_{p,q}\ V_{pq}(d_p, d_q)$ where $\boldsymbol{p}$ and $\boldsymbol{q}$ are two neighboring pixels. $V_{pq}(d_p, d_q)$ is defined as:

$$V_{pq}(d_p, d_q) = \begin{cases} 0 & if\ d_p = d_q \\ 1 & \text{otherwise} \end{cases} \qquad (4)$$

The energy function $E$ is optimized using the hierarchical belief propagation approach. The bipartite graph approach was also used to save computing time [9]. The algorithm consists in an iterative message passing process. The messages depend on the 4-connected neighbors and are updated with the following rule:

$$M^{i}_{\mathrm{p}\to\mathrm{q}}(d_{\mathrm{q}}) = \min_{d_{\mathrm{p}}} \begin{pmatrix} c_{\mathrm{p}}(d_{\mathrm{p}}) + \mu \sum\limits_{\mathrm{s}\in N(\mathrm{p})} M^{i-1}_{\mathrm{s}\to\mathrm{p}}(d_{\mathrm{p}}) \\ -M^{i-1}_{\mathrm{q}\to\mathrm{p}}(d_{\mathrm{p}}) + \lambda \cdot V_{\mathrm{pq}}(d_{\mathrm{p}}, d_{\mathrm{q}}) \end{pmatrix} \qquad (5)$$

$N(\boldsymbol{p})$ is the four-connected neighborhood of the pixel $\boldsymbol{p}$, $\mu \in (0,1]$ and $M^{i}_{p\to q}(d_q)$ is the message passed from pixel $\boldsymbol{p}$ to pixel $\boldsymbol{q}$ for disparity $d_q$ at iteration $i$. Once this process has finished, the belief function is used to obtain a depth map. It is then straightforward to use the color focal stack and the depth map to obtain an all-in-focus image.

## IV. DISPLAY

The displaying of the obtained 3D information is done with autostereoscopic Philips WOWvx 3DTVs. These displays can operate in 2D mode showing conventional images or in 3D mode showing 3D scenes. In order to display the obtained color+depth stream the format conventions specified in [12] have to be accomplished. Under Windows this can be done using the API provided by Philips. However, under Linux interfacing with the TV has required some programming.

Each 3D frame is provided with a header that contains some control information related with that frame. If the display properties are not going to be modified for a while this header can be generated once and used for multiple frames. The frames are displayed on the 3DTV in full screen programming the X Window system library. The header is then attached in the upper left corner of each frame using a pixmap. Optionally an interpolation of the input can be performed in order to fit it to the size of the 3D display.

## V. RESULTS

The whole algorithm was implemented on a multi-GPU computer. The computer was provided with 2 Intel Xeon X5560 processors, 16 Gb RAM memory, 2 NVidia Tesla C1060 devices, 1 NVidia GeForce GTX 285 graphics card and a DALSA Coreco x64-Full frame grabber. The operating system used was Debian GNU/Linux 5.0 and the GPUs were programmed using NVidia CUDA 2.2. The ideas presented in [6] were used to obtain a multi-GPU implementation of the super-resolved focal stack transform and the belief propagation algorithms. A plenoptic image which is captured by the plenoptic camera and read from the frame grabber is split into 3 individual horizontal stripes except for small overlapping zones. Then each stripe is assigned to a GPU. Each GPU performs the bayer filtering and image rectification of its part previous to the computation of the focal stack and the belief propagation. Finally, the all-in-focus image is constructed. Each part is processed separately. There is some inter-GPU communication in the iterative part of the belief propagation algorithm, since messages from the frontiers have to be propagated to the contiguous part on other GPU. The scheduling scheme over the GPUs is essentially the same as that used in [7]. Each GPU has a CPU thread associated with. In order to obtain maximum performance the CPU threads should be executed on separate CPUs. For this purpose a `pthread_set_affinity_np` function call was used at beginning of each thread. This guarantees that the execution times are not seriously influenced by the Linux scheduler. Also some barriers were used for thread synchronization. Finally the partial results are recomposed in host memory and displayed on a Philips WOWvx 3DTV using the X Window library. The appeareance of the whole system is shown in figure 2.


Figure 2. The CAFADIS camera. (A) Result of 3D reconstruction shown as color+depth image on a Philips 3D display. (B) The CAFADIS camera recording the scene.

### A. *Results of the algorithm.*

The size of the plenoptic image used is 1276x957 pixels. The resulting lightfield is 116x87x11x11 pixels. The final image size obtained is 580x435 pixels with a depth resolution of 8 levels. A lightfield captured with the CAFADIS camera is shown in figure 3.

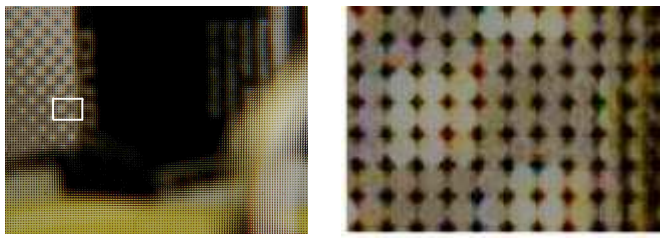

Figure 3. Lightfield data captured with the CAFADIS camera (left). Detalied view of the image in the white square (right)

Figure 4 shows the resulting all-in-focus image and depth map.

The current prototype of the CAFADIS camera works only for small scenes. Scenes of bigger size can be captured using microlenses with smaller focal length. A video showing the current system working in real-time with a small scene can be found at [13].

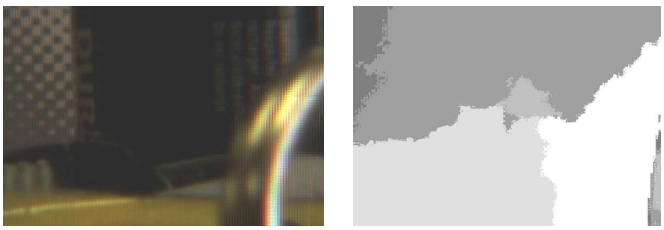

Figure 4. All-in-focus image and depth map obtained with the algorithms for the CAFADIS camera. Left: Obtained all-in-focus image. Right: Distance map.

### B. *Performance of the implementation.*

In this section the real time performance of the algorithm will be evaluated. The number of iterations in the BP algorithm influences the execution time of the system. In figure 5 the execution times and frame rates for different number of iterations are shown.

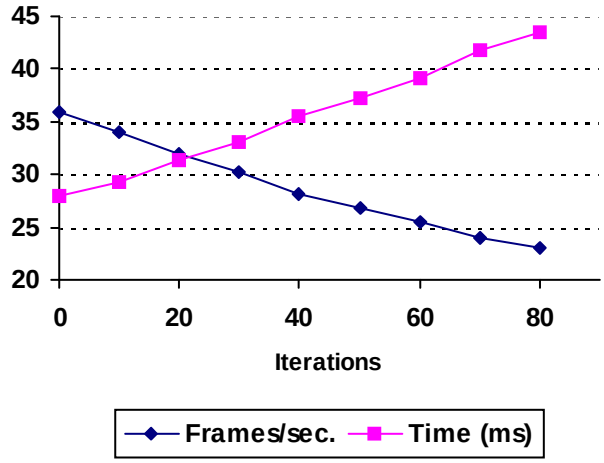


Figure 5. Frame rates and execution times versus the number of iterations of the belief propagation step.

We chose to use 50 iterations achieving about 27 fps, which is enough to feed our display.

## VI. CONCLUSIONS AND FUTURE WORK

A first working prototype of the CAFADIS camera has been presented. Several capture and processing techniques were integrated generating a color+depth stream that can be used as input for an autostereoscopic display in real time. Although the current prototype works only for small scenes we expect that the integration of these techniques on a better and faster camera will give appealing results while maintaining the ability to perform in real-time thanks to massivelly parallel hardware.

In the future, the system will be tested with bigger scenes using another optical configuration for the construction of the plenoptic camera. The output format could be adapted to other 3D displaying hardware.

## REFERENCES


[1] E. Adelson and J. Wang, “Single lens stereo with plenoptic camera”, IEEE transactions on pattern analysis and machine intelligence, vol. 14, nº2, p. 99, 1992

[2] R. Ng, “Fourier slice photography,” in Proceedings of the International Conference on Computer Graphics and Interactive Techniques (SIGGRAPH ’05), pp. 735–744, Los Angeles, Calif. USA, 2005.

[3] F. Pérez Nava and J. P. Lüke, “Simultaneous estimation of superresolved depth and all-in-focus images from a plenoptic camera,” in Proceedings of 3DTV-Conference 2009, Potsdam, Germany, May 2009.

[4] J.G. Marichal-Hernandez, J..P. Luke, F. Rosa, F. Perez Nava and J.M. Rodriguez-Ramos , "Fast approximate focal stack transform," in Proceedings of 3DTV-Conference 2009, Potsdam, Germany, May 2009.

[5] J. D. Owens, D. Luebke, N. Govindaraju, et al., “A survey of general-purpose computation on graphics hardware,” Computer Graphics Forum, vol. 26, no. 1, pp. 80–113, 2007.

[6] NVidia Corporation, “nVidia CUDA Programming Guide”.

[7] J. P. Lüke, F. Pérez Nava, J. G. Marichal-Hernández, J. M. Rodríguez-Ramos and F. Rosa “Near Real-Time Estimation of Super-Resolved Depth and All-In-Focus Images from a Plenoptic Camera Using Graphics Processing Units” in International Journal of Digital Multimedia Broadcasting, vol. 2010, Article ID 942037.

[8] T. Georgiev, K.C. Zeng, B. Curless, D. Salesin, S. Nayar, and C. Intwala. “Spatio-Angular resolution tradeoff in Integral Photography”. Proceedings of Eurographics Symposium on Rendering., 2006

[9] P.F. Felzenszwalb, D.P. Huttenlocker, “Efficient Belief Propagation for Early Vision”, Computer Vision and Pattern Recognition, 2004. CVPR 2004, vol. 1, pp. I-261-I-268, 2004

[10] A. Lumsdaine, T. Georgiev, “Full resolution lightfield rendereing”. Adobe Tech Report, January 2008.

[11] T.E. Bishop, S. Zanetti, and P. Favaro, “Light Field Superresolution”. First IEEE International Conference on Computational Photography, 2009.

[12] Philips 3D Solutions, “3D Interface Specifications, White Paper “, February 2008.

[13] http://pejeverde.lct.ull.es/cafadis/3dsa