In 24th Portuguese Conference on Pattern Recognition (RecPad 2018)
Coimbra, Portugal, October 2018

Camera Adaptation for Deep Depth from Light Fields

Diogo Filipe Baptista Portela
diogo.b.portela@gmail.com
Nuno Barroso Monteiro
nmonteiro@isr.tecnico.ulisboa.pt
José António Gaspar
jag@isr.tecnico.ulisboa.pt

Institute for Systems and Robotics
Instituto Superior Técnico
University of Lisbon, Portugal

Abstract

Plenoptic cameras image a 3D point by discriminating light rays contributions towards various viewpoints. They allow developing depth estimation methods, such as depth from focus as found in the deep neural network DDFFNet by Hazirbas et al. The training of the DDFFNet has implicit a specific camera geometry, defined by the microlens array and the configuration (zoom and focusing) of the main lens. In this paper we augment the network application range by accepting larger input disparity ranges that can be obtained by different configurations or cameras. The proposed methodology involves converting a field of view and a depth range into the settings where the DDFFNet has been trained. The conversion of the input data is based in the estimation of gradients (structure tensor) on the light field. Results show that depth estimation is possible for various cameras while using the originally trained DDFFNet.

1 Introduction

The last years have seen a rise in the study and improvement of pleonoptic cameras since the first model was developed by Ng [4], in 2006. The use of these cameras allow to obtain, from a single shot, what is called a light field, an array of multiple scene views named viewpoints, as if an array of conventional pinhole cameras were used. A light field can be digitally refocused after it has been captured, as demonstrated by Ng *et al*. [3], and used to achieve depth reconstruction, as shown by Tao *et al*. [5].

Hazirbas *et al*. [1] presented *Deep Depth From Focus Network* (DDFFNet), a Convolutional Neural Network that outputs a disparity map from a focal stack. As any neural network it requires intense training, and while it may lead to good test results, it may also result in an inability to perform well under inputs with characteristics outside its training scope. In this paper we deal with the network's inability to correctly reconstruct datasets with disparity ranges outside its scope. These ranges can vary widely depending either on the camera's zoom or focus or its physical characteristics, such its baseline.

Although the usual approach to solve this problem lying on fine tuning, that is not always possible when dealing with new data, due to constrains such as few number of examples, time or computational power.

The method presented here tries to enlarge the application range by obtaining a new light field by backprojecting the original, transforming it and finally reprojecting the result into an array of cameras identical to Hazirbas'.

2 From light fields to the Deep Depth From Focus Network

We can describe a light ray using the pixel it hits, in the form of a light field $\mathcal{L}(i,j,k,l)$, where (k,l) indicates the viewpoint's index within the array, while (k,l) indicates the pixel in the viewpoint.

We focus on a disparity $\frac{\partial i}{\partial k} = \frac{\partial j}{\partial l} = \alpha$ by performing shearing, this is translating each viewpoint by an amount proportional to its distance from the array's center, $\mathcal{L}_\alpha(i,j,k_\alpha,l_\alpha) = \mathcal{L}(i,j,k+\alpha(i_{center}-i),l+\alpha(j_{center}-j))$, and summing them. By stacking multiple images, each focused at a different disparity, we obtain what is called a focal stack.

The DDFFNet takes a focal stack of 10 images, each focused at linearly spaced disparities, to produce a disparity map as output. The network was trained for a depth range of [0.5, 7] meters, meaning, by the camera parameters, that the input focal stacks cover disparities between [0.02, 0.28] pixels. Datasets with ranges outside this scope are incorrectly reconstructed. Retraining is not always possible, so we propose simulating a camera as the one used in the training process by transforming the new light field in one similar to the ones captured for training the DDFFNet.

3 Ground Truth based Camera Adaptation

Considering a plenoptic camera as an array of pinhole cameras, its field of view can be bounded by the envelope of all cameras' fields of view (pyramids). This envelope is not much wider than the central viewpoint's pyramid, because usually baselines are very small.

We will transform the new dataset, as a point cloud, from the original trunk of pyramid to one similar to the DDFFNet camera, forming a new light field similar to the one the latter would captured. As example, we used the dataset *Cotton*, Figure 1(f), present in the 4D Light Field Benchmark [2], which its disparity range is outside the ones used for training the network.

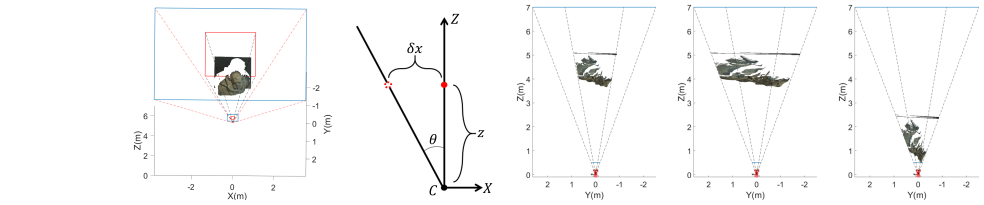
Backprojection In the first step we backproject the center viewpoint, resulting in a 3D point cloud, as in Figure 1(a). Besides the camera's intrinsic parameters, we need a depth estimation for each pixel, obtain through the ground truth or through some depth estimation method, such as the structure tensor, explained in detail in section 4. With depth Z we compute the other 3D coordinates, (X,Y), using the backprojection model $[X\ Y\ Z\ 1]^T = [C^T\ 1]^T + [Z.D^T\ 0]^T$ where $C = -P_{1:3}^{-1} \cdot P_4$ represents the optical center, $D = P_{1:3}^{-1} \cdot [u \quad v \quad 1]^T$ is the optical ray's direction for a given (u,v) pixel and P_i the projection matrix's i^{th} column.

***FOV* rotation and scaling** We obtain the two model's fields of view by backprojecting the image corners, Figure 1(a) with Benchmark and Hazirbas' in red and blue, respectively. To align their centers, the point cloud is rotated along X and Y around the optical center. For each, the rotation angle can be computed backprojecting both cameras' principal points to a depth z, Figure 1(b) where the red and white dot are the Benchmark and Hazirbas' projection, respectively. We conclude that $\theta = tan^{-1}(\delta x/z)$. The other dimension's angle can be computed in an analogous form.

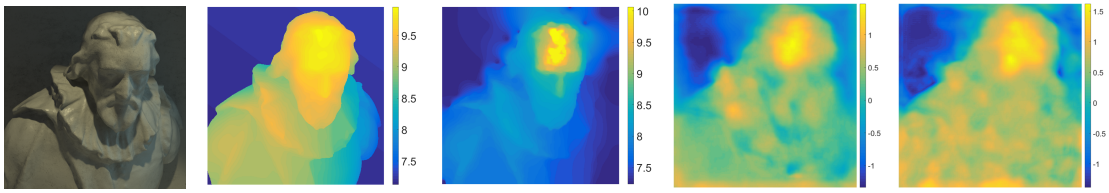
To match the field of views vertex angles, we scaled X and Y, by the same factor to avoid distortion. However, due to their different shapes (Benchmark's is a square while Hazirbas' a rectangle), the scaling factor is the one that scales the point cloud so that it matches Hazirbas' smaller side, that is, the ratio between their maximum Y, for the same depth. The result of this scaling can be visualized in Figures 1(c) and 1(d).

Depth scaling We have now to scale the point cloud so that its depth lies in network's trained range. However, inspecting Figure 2(b) of the supplementary material available in [1], we conclude that using a smaller range will result in a more well distributed set of focused depths. Thus the range used was [0.5, 2.5] meters. Through an affine transformation we can force the depth to fall within a range $[z_1,z_2]$ by solving the linear system, $z_{min}a + b = z_1$ and $z_{max}a + b = z_2$, with z_{max}, z_{min} the original maximum and minimum depths, respectively. To compensate for this, the X and Y are multiplied, $(x_{new},y_{new}) = \frac{z_{new}}{z} \cdot (x,y)$, for each point. See Figure 1(e).

Reprojection The final step is to project the point cloud to a camera array with the same intrinsic and extrinsic parameters as the camera used in the DDFFNet. This results in the new light field which will then be refocused and used to create the input focal stack.



(a) Original point cloud (b) *FOV* rotation (c) Before *FOV* Scaling (d) After *FOV* Scaling (e) After depth Scaling



(f) Central viewpoint (g) Ground truth (h) STDI (i) STDI + DDFF (j) STII + DDFF

Figure 1: Depth reconstruction from a light field. Light field central viewpoint (f) and ground truth data (a, g) from the dataset [2]. Ground truth based camera adaptation, point cloud transformations (b - e). Camera adaptation based on the structure tensor (h - j).

4 Structure Tensor based Camera Adaptation

In real cases 3D point clouds are not available. We propose obtaining an initial depth estimation to construct the point cloud.

In a light field, slices made by fixing (i,k) or (j,l) will result in an epipolar image. The disparity of a feature will translate as the slope of a epipolar line in these images, being parallel to the gradient direction such that $\frac{\partial i}{\partial k} = -\frac{\nabla_i \mathcal{L}}{\nabla_k \mathcal{L}}$, as in Figure 2. By measuring the gradient in those images we can extract a depth estimation.

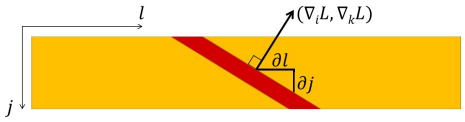


Figure 2: Epipolar plane. Depth information can be obtained from the gradient.

For this we need to obtain each pixel structure tensor, $S(k,l)$, a matrix derived from the gradient that will give its predominant direction in that pixel.

Let $I^{ik}_{(.)}(j,l)$ be the value of $\nabla_{(.)}L$ calculated at (j,l), for a horizontal epipolar image, calculated using a Sobel operator. For each pixel the local structure tensor, $S_0(j,l)$, is computed as

$$S_0(j,l) = \begin{bmatrix} I^{ik}_j(j,l)^2 & I^{ik}_j(j,l) I^{ik}_l(j,l) \\ I^{ik}_l(j,l) I^{ik}_j(j,l) & I^{ik}_l(j,l)^2 \end{bmatrix}. \quad (1)$$

Values are then averaged along j, resulting in a 1D array $S^{ik}_e(l)$. This process is repeated for every horizontal and vertical epipolar image.

By computing $S(k,l) = \sum_i \sum_j S^{ik}_e(l) + S^{jl}_e(k)$, where $S^{jl}_e(k)$ represents the average 1D array for vertical epipolar images, we obtain the value of the final structure tensor.

In a structure tensor matrix, computing the eigenvector corresponding to the greatest eigenvalue, λ_1, gives the predominant gradient direction. The relation between both eigenvalues allow for a confidence level on the gradient obtained. Such measure was defined as $\lambda_1 - \lambda_2$, with cases below a given threshold discarded.

After computing the structure array, its eigenvectors are calculated and filtered, and from them the gradient directions are computed. These are then used to compute the disparity for each pixel. This disparity map is converted to depth and used to construct the point cloud.

To deal with areas of low gradient being discarded, we propose two strategies. Constructing the point cloud as is and inpaint each viewpoint in intensities, or inpainting the disparity map, and then projecting a full point cloud.

5 Experimental results

Given the focal stack input, the network was used to obtained a new point cloud to be transformed using the inverse transformation of each step, in reverse order. Each point is projected to a camera identical to the Benchmark central viewpoint, forming a depth map, then converted to disparity and compared to the ground truth.

The proposed method was evaluated on the Benchmark's [2] Training Set, the same used by Hazirbas to evaluate the network performance after retraining. In Table 1 are the numerical results, as defined in [1]. Pre refer to results using the untransformed datasets. GT concerns the ground truth based approach. STDI and STII refer to structure tensor methods complemented with disparity or intensity inpainting, respectively. As a qualitative analysis, the disparity map obtained by each method is presented in Figure 1, along with the ground truth and central viewpoint.

Method	Pre	Retrain	GT	STDI	STII+DDFF	STDI+DDFF
Disparity MSE	0.7741	0.19	0.3002	0.7383	0.5378	0.3392
Disparity RMS	0.8709	0.42	0.5463	0.7934	0.7227	0.5765
Depth MSE	0.9395	—	0.2934	1.1063	0.6499	0.3104
Depth RMS	0.7233	—	0.4220	0.6950	0.5958	0.4332

Table 1: Retrain and ground truth (GT) vs structure tensor methods.

Analyzing the results we conclude that applying the network directly on the 4D Benchmark datasets produces an MSE in depth of almost 1 meter, rendering it almost useless. However, by applying the proposed method through the Structure Tensor + Disparity Inpainting technique we reduce that error by more than two thirds (≈67%).

6 Conclusions

In this paper we presented a method to overcome the small disparity range limitation of the DDFFNet without resorting to retrain. The datasets used in this proof of concept were restricted to the benchmark [2], however, other datasets can be transformed to a valid DDFFNet input, provided the intrinsic parameters of the camera are known. Comparing to a full retraining approach, the proposed method provides a faster, more versatile and adapting approach at the cost of loosing some accuracy.

Acknowledgements

Work partially supported by the FCT project UID / EEA / 50009 / 2013.

References

[1] C. Hazirbas, L. Leal-Taixé, and D. Cramers. Deep depth from focus, November 2017. arXiv:1704.01085v2.

[2] K. Honauer, O. Johannsen, D. Kondermann, and B. Goldluecke. A dataset and evaluation methodology for depth estimation on 4d light fields. In *ACCV*, 2016.

[3] R. Ng. Fourier slice photography. *ACM Transactions on Graphics*, 24:735–744, 2005.

[4] R. Ng, M. Levoy, M. Bredif, G. Duval, M. Horowitz, and P. Hanrahan. Light field photography with a hand-held plenoptic camera. *Comput. Sci. Dept., Stanford Univ., Tech. Rep.*, 2004.

[5] M. Tao, P. Srinivasa, J. Malik, S. Rusinkiewicz, and R. Ramamoorthi. Depth from shading, defocus, and correspondence using light-field angular coherence. *CVPR*, June 2015.