Depth Map Estimation from Light Field images

Alwi Husada[1] and Yuriy Anisimov[2]

[1] a_husada15@cs.uni-kl.de
[2] yuriy.anisimov@dfki.de

Abstract. *This report summarize a framework that estimates depth from light field camera based on a paper with the title "Accurate Depth Map Estimation from a Lanslet Light Field Camera" by Jeon et al.[9]. As it is explained in the original paper, the framework consist of four steps: (1) Constructing cost volume. In this step, phase shift method is used, since it enables sub-pixels displacement in narrow baseline of stereo correspondence. (2) Cost aggregation is then applied to the computed cost-volume to reduce noises. (3) Multi-label optimization is used to propagate accurate disparity labels to the weak regions by using graph cut algorithm and weighted median filter. And to enhance computed disparity map (4) iterative refinement is used by applying quadratic polynomial interpolation to the computed depth from previous steps. In this project, we implement the framework using C++.*

Keywords: depthmap, lightfield

1 Introduction

Depth estimation is a set of techniques, aiming to obtain representation of a spatial structure of a scene. In other words, there are techniques to obtain depth information of scene in 2D planes. Extensive works have been done in this fields where many methods and setup with excellent results have been proposed especially in stereo vision techniques [3]. However, some of those techniques are not suitable for light field images.

Light fields can be described as a set of light rays, that travelling in every direction from every point in space [11,6]. It can be parametrized as 3D coordinates as ray positions as well as 2D rays directions. The parameters can be reduces to 4D space due to redundant information caused by the radiance along the ray remain constant. This 4D light field is currently used in many research and applications. Light field camera captures not only two-dimensional representation of scene but also directional light information. This allows capabilities of post-captures parameters adjustment such as aperture size and focus. There are many ways to capture light field images such as camera arrays, micro-lens array, code masks, objective lens array and gantry-based camera systems [6]. In this project we will focus on micro-lens array as it is implemented in commercial light field camera (i.e. Lytro [2] and Raytrix [5]).

According to Alam *et al.*[6] in their paper, micro-lens based camera suffer from low spatial resolution issues. It is caused by the sharing mechanism of its imaging sensor to capture both angular and spatial information. Hence there is trade-off between spatial and angular resolution. Another limitation of micro-lens based is narrow baseline. Because of this issue the existing stereo matching algorithm cannot produce satisfying result for light field images. Thus, algorithm for stereo matching with narrow baseline is required. Jeon *et al.*[9] proposed the algorithm for estimating depth from light field images. The algorithm will be used as main source for this project. The objective of this project is to implement proposed algorithm in C/C++ code.

The main method of the proposed algorithm by Jeon *et al.* is phase shift theorem. It is used to estimated sub-pixels shift of sub-aperture images to get sub-pixels accuracy in narrow baseline. The cost volume is computed by shifting sub-aperture images (except center view of sub-aperture image) at different sub-pixel locations, then compute similarity measurement between shifted sub-aperture images and center view of sub-aperture image. To reduce noisy image, the edge-preserving filter is then applied in each cost volume slice. In this project, we experiment with three different edge-preserving filters. those are guided filter (as implemented in original paper), domain transform filter and bilateral filter. Multi-label optimization is then applied to the estimated depth to decrease oversmoothing in the edge region. In the final step, quadratic polynomial interpolation is used to enhance the estimated depth map. Details methods are explained in the next section.

2 Framework

In this section, the proposed depth estimation algorithm is described. As mentioned earlier, the method used in this report is originally based on cost-volume for stereo camera. However, in order to accommodate small baseline between sub-aperture images in light field, three modifications have been made. First, phase shift algorithm was applied directly to sub-aperture images that enables point correspondence in narrow baseline at sub-pixel accuracy. Second, weight terms based on vertical or horizontal deviation between sub-aperture image pairs is defined. It is used for effectively aggregate gradient costs. Last, confidence matching correspondences are included in label optimization.

2.1 Cost Volume construction

As it was previously mentioned, phase shift algorithm is used for sub-pixel displacement. In phase shift theorem, sub-pixel displacement is shifting image I by $\Delta x \in R^2$ that can be represented in 2D Fourier transform as:

$$\mathcal{F}\{I(x + \Delta x)\} = \mathcal{F}\{I(x)\}exp^{2\pi i \Delta x}, \tag{1}$$

where $\mathcal{F}\{.\}$ is the discrete Fourier transform. Thus, the sub-pixel shifted image $I'(x)$ can be obtained using Inverse Fourier transform as per equation below:

$$I'(x) = I(x + \Delta x) = \mathcal{F}^{-1}\{\mathcal{F}\{I(x)\}exp^{2\pi i \Delta x}\}, \tag{2}$$

where $\mathcal{F}^{-1}\{.\}$ is the inverse discrete Fourier transform. This algorithm shifts the entire sub-aperture image instead of local patches. It is intended to localized the artifact caused by periodicity at the boundary of the image within a width less than two pixels, which is insignificant to the final depth computation.

Matching sub-aperture images is done by using two complimentary costs, which are sum of absolute differences (SAD) and sum of gradient differences (GRAD). Sum of absolute differences C_A is the difference between shifted sub-aperture images and the center view of sub-aperture image. Then it followed by truncating the value with the robust function τ_1. The sum of absolute difference C_A can be defined as a function x and label l :

$$C_A(x, l) = \sum_{s \in V} \sum_{x \in R_x} min(|I(s_c, x) - I(s, x + \Delta x(s, l))|, \tau_1) \tag{3}$$

where R_x is rectangular region with x as a center, τ_1 is a truncation value for robust function, V contains coordinate pixel st except of center image s_c and the Δx is 2D shift vector defined in the following equation:

$$\Delta x(s, l) = kl(s - s_c) \tag{4}$$

where l is label and k is pixels unit of the label. Δx is linearly increases as the angular deviation of the sub-aperture images and the center images increases. Another costs is sum of gradient differences

(GRAD).It is defined as follows:

$$\begin{aligned} C_G(x,l) = \sum_{s\in V}\sum_{x\in R_x} & \beta(s)min(|I_x(s_c,x) - I_x(s,x+\Delta x(s,l))|,\tau_2) \\ & +(1-\beta(s))min(|I_y(s_c,x) - I_y(s,x+\Delta x(s,l))|,\tau_2) \end{aligned} \tag{5}$$

where $|I_x(s_c,x)-I_x(s,x+\Delta x(s,l))|$ is the differences between x-directional gradient of sub-aperture image I and $|I_y(s_c,x)-I_y(s,x+\Delta x(s,l))|$ denotes as the differences in y-directional gradient. $\beta(s)$ is calculated based on sub-aperture images coordinates, to determine the relative importance between the two gradients. $\beta(s)$ is defined as follows:

$$\beta(s) = \frac{|s-s_c|}{|s-s_c|+|t-t_c|} \tag{6}$$

Accordingly, The cost volume C can be defined as follows:

$$C(x,l) = \alpha C_a(x,l) + (1-\alpha)C_G(x,l) \tag{7}$$

where $\alpha \in [0,1]$ tunes the relative importance between sum of absolute difference and sum of gradient difference costs.

2.2 Cost Aggregation

To remove noisy effect from cost volume step cost aggregation is applied. The main idea of cost aggregation here is to filter each slice of cost volume with edge-preserving filter to remove unreliable matches. There are some edge-preserving filters available such as Bilateral filter, Domain Transform and Guided filter. In the original implementation, Guided Filter by He *et al.*[8] is used. The central view of sub-aperture image is used for image guide. From the filtered cost volume C', then a disparity map l_a can be determined using winner-takes-all strategy as:

$$l_a = arg\,min\ C'(x,l) \tag{8}$$

As it was mentioned earlier, in this project we experiment with three difference edge-preserving filters. In addition to guided filter, we also implement domain filter and bilateral filter.

Bilateral filter is filtering method by combining domain and range filters to achieve smoothness in weak texture as in traditional domain filters and preserve high texture scene such as edges, lines or corners. It works by weight averaging the neighborhood pixel values based on theirs distance in both domain and range [13]. Whereas, the main idea of domain transform filter is that if an 2D RGB image manifold in 5D space (x,y,r,g,b) can be transformed to lower dimension and still preserve the distance among the pixels, then many spatially-invariant filters in this new space is edge preserving [7]. In this project we only use recursive filter (RF) of domain transform filter. The comparison of the result with those different filters can be seen in Fig.(3)

2.3 Multi-label optimization

Graph Cut.[10] Multi-label optimization is performed from the point of view of energy minimization by using graph-cut. The main idea is to construct specific graph for energy function. Then energy minimization can be performed by using min-cut/max-flow algorithm such that the minimum cut on the graph also minimize the energy. The following is the energy function to be minimized:

$$\begin{aligned} l_a = arg\,min\ \sum_x C'(x,l(x)) + \lambda_1 \sum_{x\in I} \parallel l(x) - l_a(x) \parallel + \\ \lambda_2 \sum_{x\in M} \parallel l(x) - l_c(x) \parallel + \lambda_3 \sum_{x'\in N_x} \parallel l(x) - l(x') \parallel \end{aligned} \tag{9}$$

Energy function equation (9) has four terms: $C'(x, l(x))$ is matching cost reliability, $\parallel l(x) - l_a(x) \parallel$ is data fidelity, $\parallel l(x) - l_c(x) \parallel$ is confident matching cost and $\parallel l(x) - l(x') \parallel$ is local smoothness. Note that in this project we do not implement confident matching cost because at it is stated in the original paper that even without confident matching, the proposed algorithm still produce reliable disparity map.

Weighted Median Filter.[12] After performing graph cut optimization then weighted median filter is applied to the computed depth. It is used if the estimated disparity map from previous steps is still noisy. In (unweighted) Median filter, value of pixel is replaced by the median value of its neighbors. It treats each neighbors equally and this can remove thin structures and rounding sharp corners. Those problems can be overcame by weighted median filter. In weighted median filter, pixels are weighted in the local histograms by an image. As per mentioned in the original paper, any edge-preserving filter can be used as weight in here.

2.4 Iterative Refinement

Iterative refinement is post-processing step to enhance existing disparity map. it is adopted from paper by Yang et al. [14]. The author claimed that by applying this algorithm can enhance spatial resolution up to 100x. In this steps, a new cost volume is built based on computed disparity map from previous steps. The square different is selected for constructing new cost volume since quadratic polynomial interpolation is used for sub-pixel estimation in later step.The construction of new cost volume can be formulated as follow:

$$\hat{C}(x, l_r) = min(\eta * L, (d - l_a(x, y))^2) \tag{10}$$

where d is potential depth candidate, l_a is computed depth from previous steps, L is search range and η is constant parameter. This cost function can help to preserve sub-pixel accuracy from previous computed depth.

In the original paper [14], bilateral filtering is then applied to each slice of the cost volume. However in this implementation we use weighted median filter as suggested by Jeon et al.[9]. Finally, to reduce discontinuities, sub-pixel estimation algorithm is used.This algorithm is based on quadratic polynomial interpolation. It approximate the cost function between three discrete depth candidate: $C(l_r), C(l_+), C(l_-)$. Then, a non-discrete disparity l^* can be obtained via:

$$l^* = l_r - \frac{C(l_+) - C(l_-)}{2(C(l_+) + C(l_-) - 2C(l_r))} \tag{11}$$

where l_+ (cost slice plus 1), and l_- (cost slice minus 1) are adjacent cost slices of l_r. For better result this procedure is applied iteratively. As per mentioned here[9], four iteration is enough to get satisfying result.

3 Results and Implementation

In this project we implement the proposed algorithm by Jeon et. al. [9] in C++. We utilize armadillo [1] and OpenCV [4] as libraries. To evaluate our implementation we use Lytro [2] datasets provided by Jeon et. al. together with MATLAB implementation from this website [3]. A machine with Intel i7 2 GHz CPU and 8 GB RAM was used for running the computation. The original code in MATLAB required 15 minutes for the Lytro datasets. However, our implementation in C++ requires longer to run: 19 minutes. The cost volume construction step required longest time to run compare to the other steps. Table below summarize running time at each step:

[3] https://sites.google.com/site/hgjeoncv/home/depthfromlf_cvpr15

Steps	**Matlab**	**C++**
Cost Volume	404.45 seconds	1070.87 seconds
Cost Aggregation	239.23 seconds	5.62 seconds
Graph Cut	30.76 seconds	64.8189 seconds
WMF	44.77 seconds	5.22 seconds
Iterative Refinement	184.05 seconds	18.41 seconds
TOTAL running time	**903.26 seconds**	**1164.94 seconds**

For the evaluation, the parameter values are selected equal to parameters that are stated in the paper. They might be varied depends on the datasets. The selected parameters values can be seen in Fig.(1).

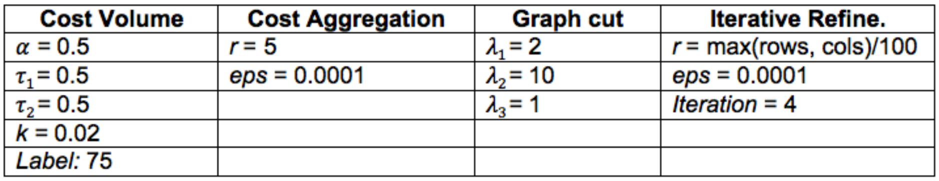
The comparison between C++ and MATLAB results at different steps can be seen in Fig.(2). We compute the means square different among them to evaluate quantitatively. As it was mentioned in section 2.2, we applied three different edge-preserving filter at cost aggregation step namely guided filter, domain transform filter and bilateral filter. The comparison among them is shown in Fig.(3). There is not much difference among them, however based on the Fig.(3) domain filter shows slightly better compare the other two.

4 Conclusion

A method for estimating disparity for light field images was proposed by Jeon et. al. [9]. Based on the results that mentioned in the original paper, this method out performed three existing methods. It show that sub-pixel shift in frequency domain by using phase shift theorem is effective for depth estimation in narrow baseline. Furthermore, the adaptive aggregation of the gradient costs and confidence cost by matching correspondence enhanced the depth map accuracy. The main drawback of this algorithm is the running time especially in cost volume computation. But it is expected that the speed can be significantly increased by parallelizing using GPU.

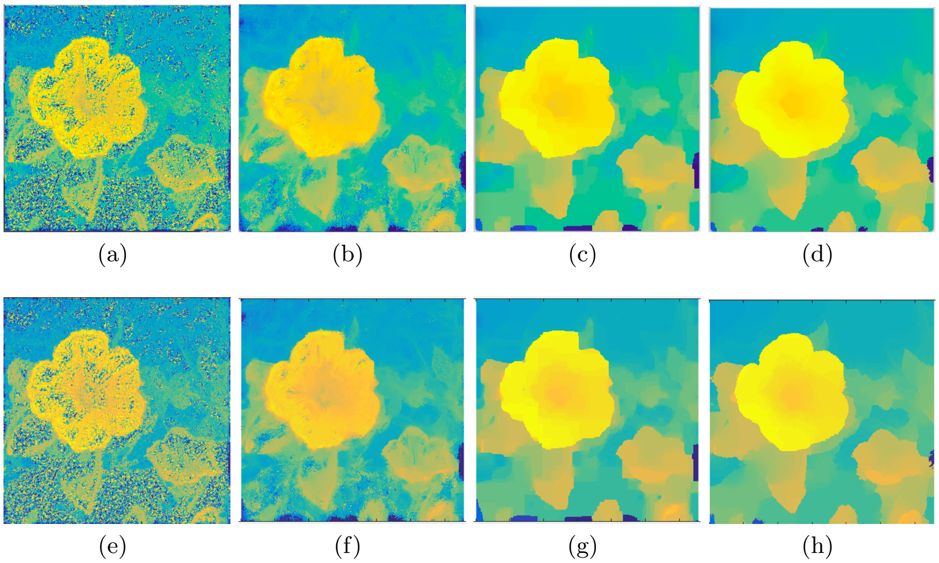
References

1. Armadillo. http://arma.sourceforge.net. Accessed: 2017-03-12.
2. Lytro. https://www.lytro.com/imaging. Accessed: 2017-03-12.
3. Middlebury stereo benchmark. http://vision.middlebury.edu/stereo/eval3/. Accessed: 2017-03-11.
4. Open cv. http://opencv.org. Accessed: 2017-03-12.
5. Raytrix. https://www.raytrix.de. Accessed: 2017-03-12.
6. M. Zeshan Alam and Bahadir K. Gunturk. Hybrid light field imaging for improved spatial resolution and depth range. *CoRR*, abs/1611.05008, 2016.
7. Eduardo SL Gastal and Manuel M Oliveira. Domain transform for edge-aware image and video processing. In *ACM Transactions on Graphics (ToG)*, volume 30, page 69. ACM, 2011.
8. K. He, J. Sun, and X. Tang. Guided image filtering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 35(6):1397–1409, June 2013.
9. H. G. Jeon, J. Park, G. Choe, J. Park, Y. Bok, Y. W. Tai, and I. S. Kweon. Accurate depth map estimation from a lenslet light field camera. In *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1547–1555, June 2015.
10. Vladimir Kolmogorov and Ramin Zabih. Multi-camera scene reconstruction via graph cuts. In *European conference on computer vision*, pages 82–96. Springer, 2002.
11. Marc Levoy. Light fields and computational imaging. *Computer*, 39(8):46–55, 2006.
12. Z. Ma, K. He, Y. Wei, J. Sun, and E. Wu. Constant time weighted median filtering for stereo matching and beyond. In *2013 IEEE International Conference on Computer Vision*, pages 49–56, Dec 2013.
13. C. Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Sixth International Conference on Computer Vision (IEEE Cat. No.98CH36271)*, pages 839–846, Jan 1998.
14. Q. Yang, R. Yang, J. Davis, and D. Nister. Spatial-depth super resolution for range images. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007.



Cost Volume	Cost Aggregation	Graph cut	Iterative Refine.
$\alpha = 0.5$	$r = 5$	$\lambda_1 = 2$	r = max(rows, cols)/100
$\tau_1 = 0.5$	*eps* = 0.0001	$\lambda_2 = 10$	*eps* = 0.0001
$\tau_2 = 0.5$		$\lambda_3 = 1$	*Iteration* = 4
$k = 0.02$			
Label: 75			

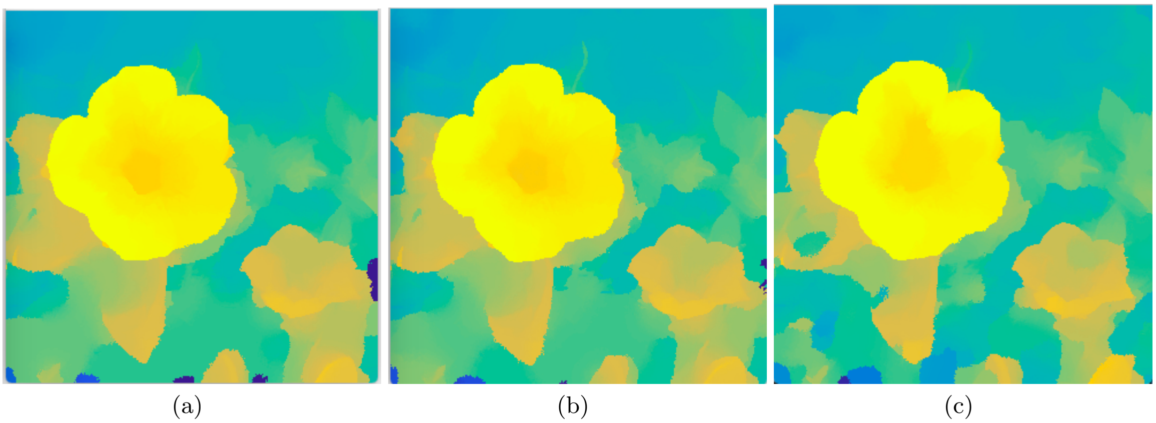
Fig. 1. Selected parameter values at different step of the framework



(a) (b) (c) (d)

(e) (f) (g) (h)

Fig. 2. Estimated disparity map at different step of the framework and also comparison of original MATLAB with C++ results. in the first row (upper (a-d)) show C++ result. Second row (lower (e-h)) show MATLAB result. (a and e) based on the initial cost volume -**MSD= 0.026507**, (b and f) after weighted median filter refinement -**MSD= 0.001259**, (c and g) after multi-label optimization -**MSD= 0.000885** and (d and h) final results, after iterative refinement -**MSD= 0.001566**.



(a) (b) (c)

Fig. 3. comparison of results from three different edge-preserving filter in cost aggregation step. (a) Guided filter, (b) Domain transform (RF) and (c) Bilateral filter. There is not much difference among them. But (b) shows little bit better since it removes holes produce in (a)