Depth Estimation of Light Field Images by Optical Flow

Marcus Pan
mpanj@stanford.edu

Abstract

Light field images present a rich, compact, dataset of multiple images of a scene from slightly shifted viewpoints. I explore the use of an optical flow algorithm on light field images to estimate depth. The optical flow algorithm is simple to implement as does not require finding correspondences between images. Further, I explore the use of image priors and modern optimization techniques to form a better estimate. Where camera parameters are available, I estimate the metric depth of objects in a scene.

I compare my the results of my algorithm with the results from a focal stack algorithm, the depth map produced by Lytro, and with ground truth from a synthetic dataset. The results show that it performs better than the basic depth from f algorithm, and gives comparable results to the Lytro map and ground truth in areas where the image has sufficient features.

1. Introduction

Depth estimation of 2D images requires a set of at least 2 images, and usually involves a feature matching algorithm. This problem can be simplified by using a light field camera, that embeds multiple images from slightly shifted viewpoints into a single sensor, thus eliminating the need for calibration of multiple cameras.

Further, using an optical flow (OF) algorithm to estimate depth removes the need to solve the feature matching problem, which can be tricky for images with regular patterns. The optical flow is the ratio of the pixel shift to its corresponding viewpoint shift, and be calculated only with image gradients.

With the optical flow estimate, the metric depth can be derived if certain camera parameters are obtained. Without those parameters, the relative depth can still be visualized.

2. Related Work

Adelson *et al*. was one of the early researchers in this field who described a design for a plenoptic camera [1]. This camera had a microlens array in front of the sensor, that would split light coming from different angles onto different pixels. Since then, companies such as *Lytro* and *Raytrix* have made consumer light field cameras readily available.

Adelson *et al*. proposed a basic OF algorithm for depth estimation. OF algorithms have also been applied to estimate depth from video [5]. Others have researched other techniques such as feature matching, focal stack analysis, or combinations of them [6]. These techniques tend to perform well for the specific kinds of pictures, that have features that suit the algorithm.

My approach combines Adelson's OF algorithm, with an image prior. I also estimate metric depth when camera parameters are available. When not available, Williams *et al*. [7] and Bok *et al*. [2] have described calibration techniques to recover these parameters.

3. Methods

3.1. Optical Flow

determine 'pixel velocities', or pixel shifts from subsequent video frames. For the light field image, the reference variable is viewpoint shift and not time, and hence optical flow measures the pixel shift over the viewpoint shift.

The basic equations in [1] for optical flow estimation are rederived in greater detail here. Let $I(x, y, v_x, v_y)$ be the pixel (x, y) at viewpoint (v_x, v_y) of a light field image. We take the center view as the reference and find the average optical flow with respect to all other viewpoints. We scale the viewpoint and pixel baselines to have increments of 1. Let a viewpoint be shifted by ϵ in the α direction:

$$\Delta_{v_x} = \epsilon \cos\alpha, \Delta_{v_y} = \epsilon \sin\alpha$$

Let the optical flow at a pixel, h be

$$h = \frac{\Delta_x}{\Delta_{v_x}} = \frac{\Delta_y}{\Delta_{v_y}}$$

The ratio is equivalent in the x and y directions by similarity of triangles. A shift in viewpoint is thus related to a shift in pixel by h:

$$I(x, y, v_x, v_y) = I(x - h\epsilon\cos\alpha, y - h\epsilon\sin\alpha, \\ v_x + \epsilon\cos\alpha, v_y + \epsilon\sin\alpha)$$

We introduce the following notation and definition for partial derivatives:

$$I_x = I(x+1, y, v_x, v_y) - I(x, y, v_x, v_y)$$
$$I_{v_x} = \frac{I(x, y, v_x + k, v_y) - I(x, y, v_x, v_y)}{k}$$

I_y, I_{v_y} are similarly defined. Thus we linearize the shifted I about (x, y, v_x, v_y):

$$\begin{aligned} I(x - h\epsilon\cos\alpha, y - h\epsilon\sin\alpha, v_x + \epsilon\cos\alpha, v_y + \epsilon\sin\alpha) \\ \approx I(x, y, v_x, v_y) - I_x h\epsilon\cos\alpha - \\ I_y h\epsilon\sin\alpha + I_{v_x}\epsilon\cos\alpha + I_y\epsilon\sin\alpha \end{aligned}$$

To smoothen our estimate of h at $I(i, j, k, l)$, we estimate it for a small 4D patch P of $((2n+1)\times(2n+1))$ neighboring pixels, and all other viewpoints. Summing over the patch P is defined as follows:

$$\sum_P I(x, y, v_x, v_y) = \sum_{x=i-n}^{i+n} \sum_{y=j-n}^{j+n} \sum_{v_x \neq k} \sum_{v_y \neq l} I(x, y, v_x, v_y)$$

We then apply the Taylor linearization to simplify the error defined as:

$$\begin{aligned} E = \int_0^{2\pi} \sum_P (I(x, y, v_x, v_y) - I(x - h\epsilon\cos\alpha, y - h\epsilon\sin\alpha, \\ v_x + \epsilon\cos\alpha, v_y + \epsilon\sin\alpha))^2 d\alpha \\ = \int_0^{2\pi} \sum_P (-I_x h\epsilon\cos\alpha - I_y h\epsilon\sin\alpha + \\ I_{v_x}\epsilon\cos\alpha + I_y\epsilon\sin\alpha)^2 d\alpha \end{aligned}$$

Differentiating with respect to h and setting to 0 gives the following expression for the optimal h:

$$\begin{aligned} \hat{h} &= \arg\min_h E \\ &= \frac{\sum_P I_x I_{v_x} + I_y I_{v_y}}{\sum_P I_x^2 + I_y^2} \end{aligned}$$

For faster computation of $\hat{h}$ the summation over P can be computed as multiplication with a kernel K in the frequency domain:

$$\text{Let } K = \mathbf{1} \in \mathbb{R}^{2n+1 \times 2n+1}$$
$$\hat{h} = \frac{\mathcal{F}^{-1}\Big(\mathcal{F}(K)\mathcal{F}(I_x I_{v_x} + I_y I_{v_y})\Big)}{\mathcal{F}^{-1}\Big(\mathcal{F}(K)\mathcal{F}(I_x^2 + I_y^2)\Big)}$$

3.2. Regularization

Let H be the vector of $\hat{h}$ values at every pixel. We define a confidence weight, C for our H estimate based on the pixel gradients:

$$C = diag((D_x H)^2 + (D_y H)^2)$$

where D is the differential operator. We estimate a smooth value of H, H_s, by minimizing a total variation loss:

$$L(H_s) = \frac{1}{2}||C(H_s - H)||_2^2 + \lambda||DH_s||_1$$

Since the L1 norm is nonconvex, we use the ADMM algorithm [3], and reformulate the minimization problem as:

$$\begin{aligned} \min_{H_s} \quad & \frac{1}{2}||C(H_s - H)||_2^2 + ||z||_1 \\ \text{subject to} \quad & DH_s - z = 0 \end{aligned}$$

The augmented loss function is thus:

$$\begin{aligned} L_\rho(H_s, z, u) = \frac{1}{2}||C(H_s - H)||_2^2 + ||z||_1 + \\ y^T(DH_s - z) + \frac{\rho}{2}||DH_s - z||_2^2 \end{aligned}$$

The ADMM update rules are as follows:

$$\begin{aligned} H_s^{(0)} &= zeros(W, H) \\ z^{(0)} &= zeros(W, H, 2) \\ u^{(0)} &= zeros(W, H, 2) \\ H_s^{(k+1)} &= \arg\min_{H_s} \frac{1}{2}||C(H_s - H)||_2^2 + \\ &\quad \frac{\rho}{2}||DH_s - z^{(k)}||_2^2 \\ &= (C^T C + \rho D^T D)^{-1}(C^T C H + \rho D^T(z^{(k)} - u^{(k)}) \\ z^{(k+1)} &= \arg\min_z ||z||_1 + \frac{\rho}{2}||DH_s^{(k+1)} - z + u^{(k)}||_2^2 \\ &= S_{\frac{\lambda}{\rho}}(DH_s^{(k+1)} + u^{(k)}) \end{aligned}$$

$$\text{where } S_K(v) = \begin{cases} v - k & v > k \\ 0 & |v| < k \\ v + k & v < -k \end{cases}$$

$$u^{(k+1)} = u^{(k)} + DH_s^{(k+1)} - z^{(k+1)}$$

3.3. Metric Depth Estimation

Since h is defined with integer pixel and viewpoint shifts, the following camera parameters are needed to estimate the metric depth:
W: sensor width
N: no. of pixels along sensor width

b: baseline between viewpoint shifts
f: focal length of lens
S_f: focal plane of lens

Given the relative h, we derive the metric h_m:

$$h = \frac{\Delta_x}{\Delta_{v_x}}$$
$$h_m = h\frac{W/N}{b}$$

Extending [1], the formula for depth d given optical flow h_m can be derived as follows:

$$\frac{1}{d} = h_m(\frac{1}{f} - \frac{1}{S_f}) + \frac{1}{S_f}$$
$$d = \frac{fS_f}{h_m(S_f - f) + f}$$
$$\approx \frac{fS_f}{h_m S_f + f}$$

If no camera parameters are available, the relative depth is estimated as follows:

$$d_r = \frac{1}{h - min(h) + 1}$$

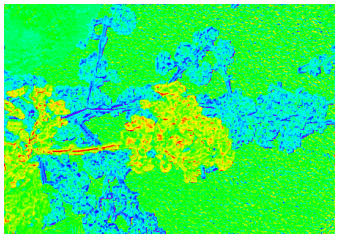
4. Results

The algorithm was tested with images taken with light field images taken with a Lytro Illum camera, and with from a synthetic dataset [**?**]. It was run in MATLAB on my personal laptop, and took about 50 s to generate the depth map. The command line tool provided by Lytro takes about 30 s. The color map for the results shown is as follows:

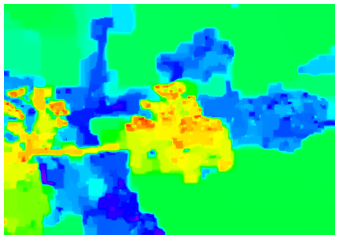
4.1. Flowers

The following shows results of flowers against a sky, taken with a Lytro Illum:

scene



unregularized OF depth

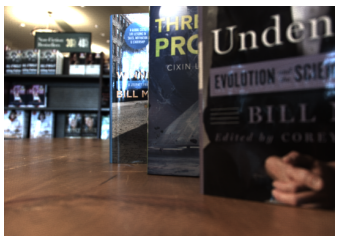


regularized OF depth

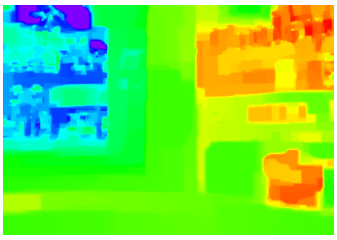
The results show how the total variation prior removes noise from the raw depth estimate. The depth of the blue sky however is inaccurate, as there simply aren't enough features.

4.2. Books

The following shows a scene with books at different depths, and a comparison with the focal stack algorithm as implemented in the homework.

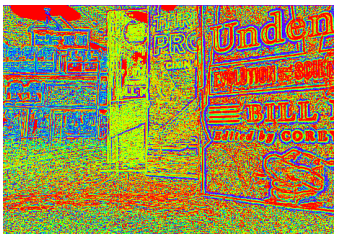


scene



regularized OF depth

unregularized OF depth

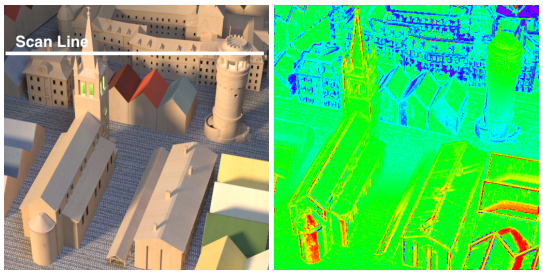


unregularized focal stack depth

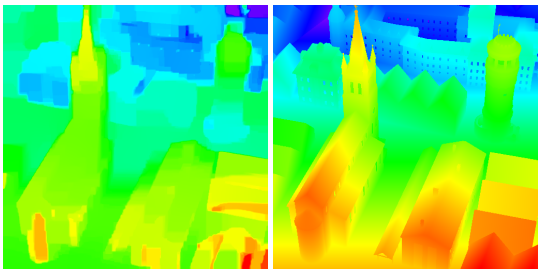
The nearest book in the regularized OF results shows how the TV prior fills in the dark cover with estimates from the feature-rich title text. Even in its unregularized form, the OF results is a lot less noisy than the focal stack algorithm. The focal stack algorithm also produces inaccurate artifacts at the lines of the image, such as the blue edges for the nearest book.

4.3. Town

The following shows the results on an image provided by a dataset from HCI, Heidelberg University and the University of Konstanz [4]. The dataset provided the ground truth depth, and the camera parameters required to estimate metric depth.

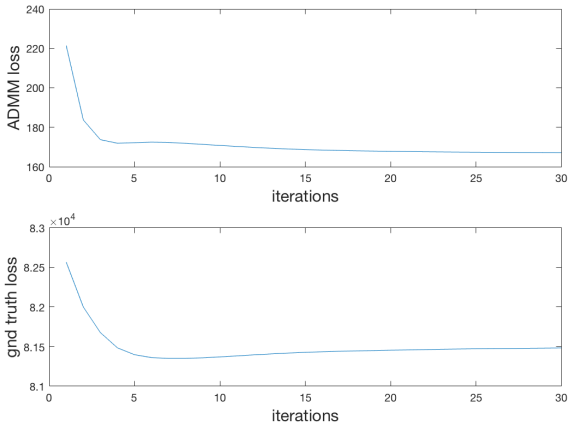


scene unregularized OF depth

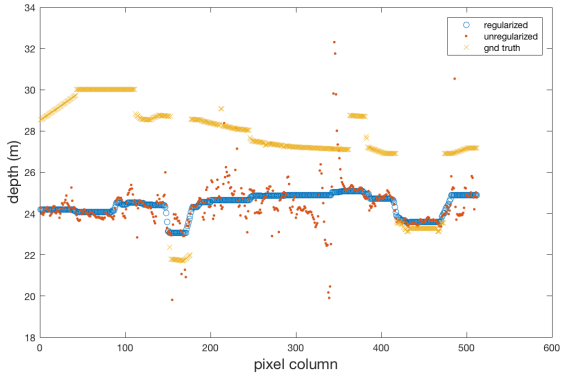


regularized OF depth ground truth

The OF algorithm performs well where there are sufficient features. For areas without features, $h \approx 0$, and the depth is pegged to the depth to the focal depth (green color). The following plots show the quantitative difference from the ground truth. The estimated depth along the scan line is plotted against the ground truth depth.



error plots



metric depth at scan line

The ground truth error is defined as the sum squared difference betweeen the estimated depth and ground truth optical flow. The ADMM error decreases with each iteration as expected, but the ground truth tapers slightly upward at

the end. This could be due to over-distortion by the ADMM algorithm.

The metric depth at the scan line shows that the scale of the depth is correct. The depth estimation for the 2 nearest towers is reasonably accurate. However it is off for the distant background. The h values could be too low, making the estimate pegged to the focal plane at 24 m. The regularizer does a good job at smoothening out the noisy initial estimate.

4.4. People

The following shows a scene with people in a hallway, and compares the results with the depth map provided by Lytro:

scene

regularized OF depth

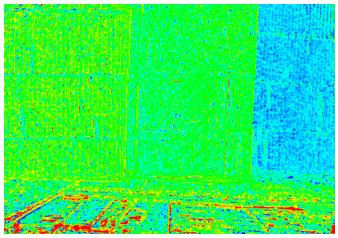
lytro depth

The OF algorithm is able to resolve the people at different depths, and the far end of the hallway. However, the Lytro map does better at low-feature areas, such the dark sweater of the closest passerby.

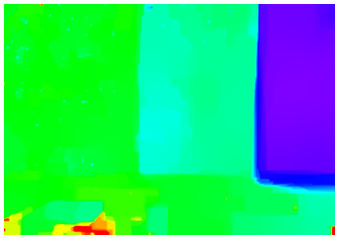
4.5. Simple Books

The following scene shows 3 books wrapped in newspaper for better features:

scene



unregularized OF depth



regularized OF depth

lytro depth

The regularized output captures the 3 planar depths of the books. However, the floor is over-smoothed, and does not show the linear distance to the camera. The Lytro depth map is able to capture this.

5. Conclusion

The optical flow algorithm works decently well with light field images. It is simple and efficient to implement in the frequency domain. This makes it preferable to the focal stack algorithm. The unregularized results are noisy, but still better than that of the focal stack algorithm.

With a weighted confidence and TV prior smoothening, it is able provide good estimates for feature-rich regions. If the features are sparse but evenly scattered in the image, the TV prior can help to fill in the feature-less areas.

However, for completely feature-less regions, such as the sky, it performs poorly. The Lytro depth map is more robust.

To improve this aspect of the algorithm, more sophisticated priors would be needed. For instance, one could segment the image, and estimate the depth of each segment individually. For sections, where the estimate is unconfident, visual recognition algorithms could be applied to learn what those segments are. These algorithms rely on surrounding regions and color, and thus won't be limited if the local region is devoid of features.

This project also explored the accuracy of the metric depth estimation, when camera parameters are available. The estimates weren't extremely accurate, and more testing needs to be done with real datasets. For this, camera calibration would need to be done.

The optical flow algorithm combined with image priors and camera calibration, has the potential to provide robust and accurate depth estimation. This makes the light field camera a powerful sensor for applications in robotics, VR, automation, and many other fields.

References

[1] E. H. Adelson and J. Y. A. Wang. Single lens stereo with a plenoptic camera. *IEEE Trans. Pattern Anal. Mach. Intell.*, 14(2):99–106, Feb. 1992.

[2] Y. Bok, H. G. Jeon, and I. S. Kweon. Geometric calibration of micro-lens-based light field cameras using line features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(2):287–300, Feb 2017.

[3] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Found. Trends Mach. Learn.*, 3(1):1–122, Jan. 2011.

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

[5] R. Ranftl, V. Vineet, Q. Chen, and V. Koltun. Dense monocular depth estimation in complex dynamic scenes. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4058–4066, June 2016.

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

[7] S. B. Williams, O. Pizarro, and D. G. Dansereau. Decoding, calibration and rectification for lenselet-based plenoptic cameras. *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 00:1027–1034, 2013.