Final project Report

Negar Nejatishahidin
Department of Computer Science
George Mason University
nnejatis@gmu.edu

Abstract

The goal of this work is to implement the belief Propagation algorithm[2] to generate the disparity map and evaluate the results with some available algorithms. Although lots of effort has been done in this area, there is still a large gap between the ground truth and the state of the art results. The earlier works more focus on the fundamental image correspondences and stereo geometries. In this report, I will go over several stereo matching algorithms. Also, I will talk about my model and the results in compere to Open CV and other basic models.

1. Introduction

The goal of this work is to implement a Belief Propagation to be able to generate a depth map To find the structure of the environment depth-maps are essential. These maps are later used for a variety of applications such as augmented reality, remote sensing, and 3D scene understanding. For example, generating the 3D structure of an area using single or multiple images has been an interesting topic since early vision. One way to generate these depth maps is by using stereo matching. This technique mainly depends on finding correspondence pixels from two viewpoints, the process is both not easy and expensive. In this project, I focused more on the disparity map. The depth map can easily be calculated from the disparity map.

Some methods in this area search for maximum match score or minimum error over a small region, typically using variants of cross-correlation or robust rank metrics like [1]. Some other methods are based on Gradient-Based Optimization like [3]. This is one of the methods that I have compared my results with. In these sort of methods, they minimize a functional, typically the sum of squared differences, over a small region. These two methods are called local methods. The method that I have used is called Belief Propagation[2] which solve disparities via message passing in a belief network which is a global method. Also, in recent years some deep learning methods are used to improve these disparity maps, like [4]. In this model, they train a network by treating the problem as multi-class classification, where the classes are all possible disparities. Also, some recent papers are published in which the solution is constructed as a hybrid between classical Stereo Vision techniques and deep learning approaches [5].

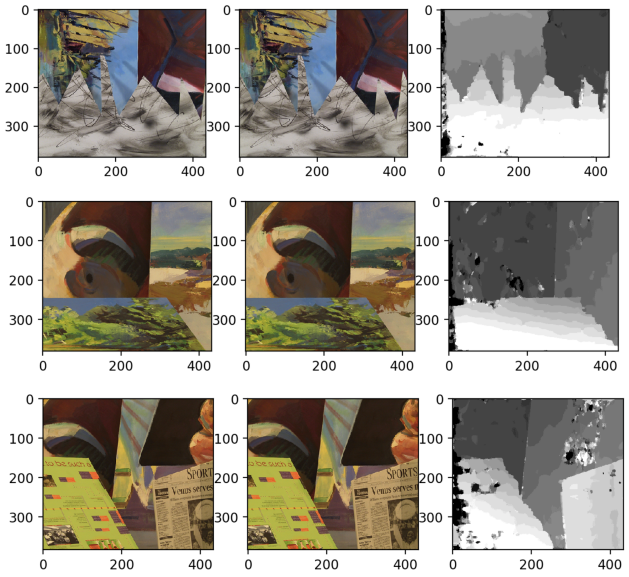

Figure 1. 3 different output with iterations=100 and t=15. From left to right, left image, wright image and disparity.

2. Theory

The problem of recovering an accurate disparity map L can be seen as an energy minimization problem,

$$E(f) = \Sigma_{(p,q)\in N} V(f_p, f_q) + \Sigma_{p\in\rho} D_p(f_p)$$

where $f_p 0, 1, 2, ..., d_m ax$ is the disparity value of pixel p, P is the set of pixels in the image, N is the set of undirected edges in the four-connected image grid graph, $D_p(f_p)$ is

called data cost which is the cost of assigning label-disparity f_p to pixel p, and $V(f_p f_q)$ is called discontinuity cost which is the cost of assigning labels f_p and f_q to two neighbor pixels.

I am going to find the disparity map that minimizes the above energy corresponds to the maximum a posteriori (MAP) estimation problem for an appropriately defined Markov random field (MRF). I have used the min-sum Loopy Belief Propagation method.

2.1. Data Cost

I have defined the data cost, but how it is calculated? The data cost is calculated through the following formula :

$$D_p(f_p) = min(||I_l(y,x)I_r(y,xf_p)||,)$$

Where denotes a truncation value. The truncation is necessary to make the data cost robust to occlusion and artifacts that violate the brightness constancy assumption. The output would be a 3-dimensional array in which the height is y, the width is x, and the depth is the data cost of label L.

2.2. Energy function

The quality of labeling is given by an energy function which is the first equation. The labels of the pixels which are neighbors would be similar else there are near an edge, in this case, the cost would be assigned using $V(f_p, f_q)$.

2.3. Update Massages

In this section The massages of each nodes are updated based on the previous massages of neighbor nods. The updates are done based on the following formula:

$$m_{pq}^{t}\left(f_q\right) = \min_{f_p}\left(V\left(f_p, f_q\right) + D_p\left(f_p\right) + \sum_{s \in \mathcal{N}(p)\setminus q} m_{sp}^{t-1}\left(f_p\right)\right)$$

Also, I have normalized the messages between each iteration, otherwise, overflow or underflow likely to occur after enough message updates. In other words, if the iteration increase, you would probably get a black picture without normalization. In practice one usually normalizes the messages to sum to 1, so that :

$$\Sigma_{x,j} M_{i,j}(x_j) = 1$$

2.4. Compute Belief

After enough iterations, this series of conversations is likely to converge to a consensus that determines the marginal probabilities of all the variables. Estimated marginal probabilities are called beliefs.

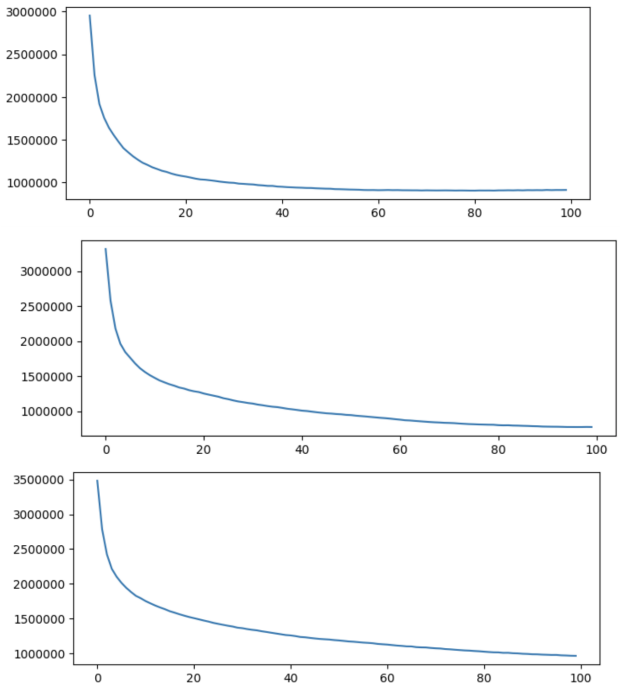

Figure 2. This is the plot of energy over iterations and sequence of images are same as figure 1.

After T iterations a belief vector is computed for each node which is the sum of the data cost and massages from all the neighbors. It will be computed as follow :

$$b_q(f_q) = D_p\left(f_p\right) + \sum_{p \in \mathcal{N}(q)} m_{pq}^{T}(f_p)$$

Finally, the label f_q that minimizes the belief individually at each node is selected.

2.5. Map labeling

In this section, I am just returning the labels for each pixel which is the best label from beliefs computed so far.

2.6. Stereo

The label map that I have mentioned would the disparity. Also, I have visualized the energy function through the iterations. You can see the image left and right and the computed disparity in figure 1. Also, the plot of the energy is shown in figure 4.1.

3. Data Set

I have used 2001 Stereo datasets with ground truth [6]. These datasets of piecewise planar scenes were created by Daniel Scharstein, Padma Ugbabe, and Rick Szeliski. Each

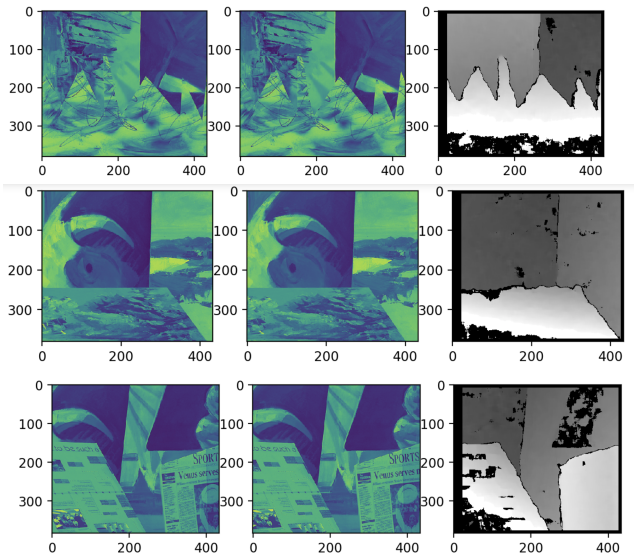

Figure 3. The result of OpenCV .

set contains 9 images (im0.ppm - im8.ppm) and ground-truth disparity maps for images 2 and 6 (disp2.pgm and disp6.pgm). Each ground-truth disparity map is scaled by a factor of 8. For example, a value of 100 in disp2.pgm means that the corresponding pixel in im6.ppm is 12.5 pixels to the left. [6]

4. Evaluation and Results

I have used the Open-CV stereo function to see the results. Previously they had Graph Cuts algorithm, but in the latest version, only "StereoBMis" is available. The results are shown in figure 3. I will compare the results with Open-CV stereo. I am also going to campare the results with the SSD and NCC methods that were implemented earlier.

To measure if the algorithms work well or not I have chosen 3 different evaluation matrix for comparing the results of 3 methods:

First: mean square error of the disparities with the ground truth (MDE). Second: median error relative to the rendered depth (Rel) Third : percentages of pixels with predicted depths falling within an interval

$$([\delta = |predictedtrue|/true]),$$

$$\delta = 1.25, 1.25^2, 1.25^3$$

You can see the results in figure 4.

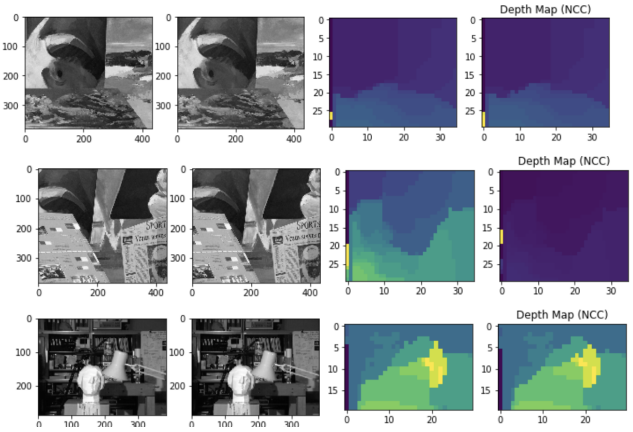

Figure 4. The result SSD and NCC. From left to right, left image, right image, SSD and NCC are shown(patch width size =45).

	MSE	Rel	< 1.25	$< 1.25^2$	$< 1.25^3$
SSD	0.2261	0.1173	243989	264630	288394
NCC	0.2246	0.1191	247552	268524	292519
Belef	0.1213	0.0722	267581	282692	296973
CV2	0.1089	0.0675	285416	299393	315558

The evaluation based of execution time is shown if figure 4.

	Speed(s)
SSD	15
NCC	95.5
Belef	37.5
CV2	1.03

4.1. Conclusion

After trying to get a good result with different data for 3 algorithms, I found that For algorithm 1, the depth would result good if the images' features can be unique according to SSD and NCC. However, I mainly work on the 4 simple images which non of them result good, However, the algorithm results good for the art image that we used for the homework. Using this idea, I increased the size of the patches to make the patch more unique. And the results dramatically increased. result are shown in figure 4.I have also added the result before increasing the patch width size in figure 5.

For the belief algorithm, I think that it does not work properly if the right image is shifted more than a few pixels. For example, it works perfectly for the simple images that you can see in figure 1, but it result not good for the art image of the homework. I have also shifted the second image to the left to get a better result which obviously results better. The result without and with the shift are shown in figure

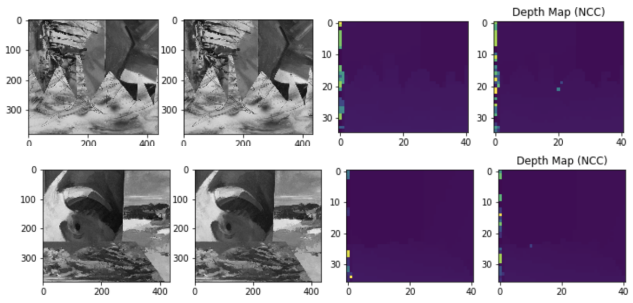

Figure 5. The result SSD and NCC before improvement. From left to right, left image, right image, SSD and NCC are shown(patch width size = 15).

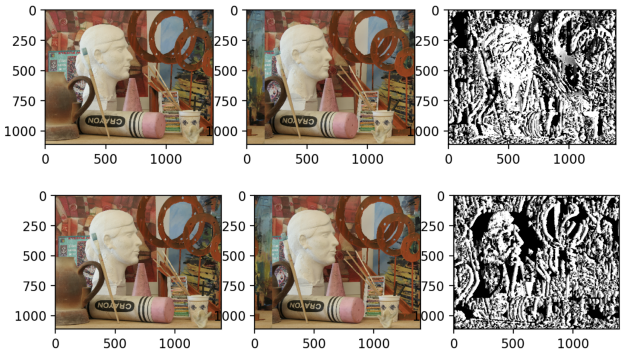

Figure 6. The result of theart image. In the top one the image is rolled. The button one is the original image

6. Also, I have added the energy plot which shows that the algorithm works for images with a little shift 7.

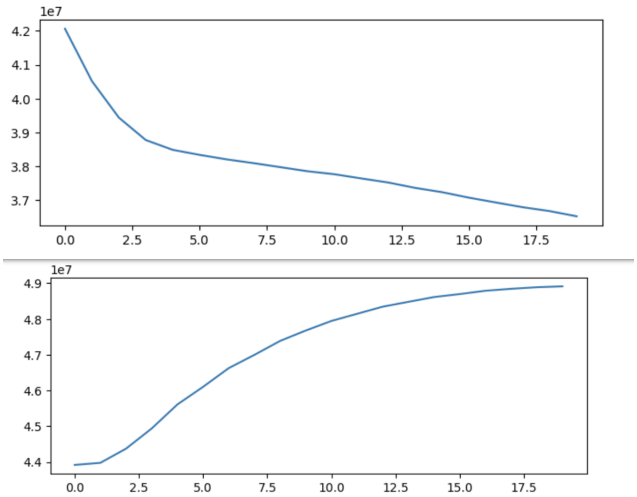

Figure 7. The energy functions. In the top one the image is rolled. The button one is for the original image.

References

[1] P. Aschwanden, W. Guggenbuhl, et al. Experimental results from a comparative study on correlation-type registration algorithms. *Robust computer vision*, pages 268–289, 1992.

[2] P. Felzenszwalb and D. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70, 07 2004.

[3] V. S. Kluth, G. W. Kunkel, and U. A. Rauhala. Global least squares matching. In *[Proceedings] IGARSS '92 International Geoscience and Remote Sensing Symposium*, volume 2, pages 1615–1618, 1992.

[4] W. Luo, A. G. Schwing, and R. Urtasun. Efficient deep learning for stereo matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[5] L. Puglia and C. Brick. Deep learning stereo vision at the edge, 2020.

[6] D. Scharstein, R. Szeliski, and R. Zabih. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. In *Proceedings IEEE Workshop on Stereo and Multi-Baseline Vision (SMBV 2001)*, pages 131–140, 2001.