Fall 2020 CS6476 Final Project: Stereo Correspondence

Chen Zhang
GT username: czhang613

1. Introduction

This project aims at finding the disparity map between two views. The disparity map using simple algorithm, and a-expansion algorithm along with various penalty models are examined in this project. It is clear that the a-expansion algorithm is powerful when dealing with occluded edges. In piano case, the best disparity map is achieved under truncated squared distance model. Different images show distinct optimization with the use of a-expansion algorithm.

2. Related work

Simple correspondence method searches for a patch around each pixel on the same epipolar line, compares the left the right image, and picks the position with minimum cost like sum of squared differences (SSD). However, this method has issues to find the best matches around edges and occluded areas [Deng, Kim].

Novel methods formulate stereo correspondence as an energy minimization problem. In this framework, we look for the labeling f that minimizes the energy [Boykov]

$$E(f) = E_{smooth}(f) + E_{data}(f)$$

in which E_{smooth} measures the extent to which f is not piecewise smooth while E_{data} measures the disagreement between f and the observed data. Boykov and coworkers brought up the energies of the form:

$$E(f) = \sum_{p \in P} D_p(f_p) + \gamma \sum_{p,q \in N} V_{p,q}(f_p, f_q)$$

where p, q in N are nearby pixels, and Vp,q measures the interaction penalty. Dp is the measurement of data term disagreement like SSD or sum of absolute differences (SAD). This kind of energy term can be optimized using graph cuts.

In this project I apply the a-expansion algorithm and optimize the disparity map especially around occluded edges. The a-expansion algorithm is a fast route towards energy minimization. Multiple interaction penalty functions are being examined.

3. Method

1) Simple algorithm

Here we assume the image provided use calibrated camaras and the epipolar line for each pixel is just it's horizontal line. Here I use the piano-perfect image from middlebury sample dataset[middlebury]. Other images are included in the discussion session later as comparison. Ground truth is provided by the dataset, and it's converted to png format through an online converter[pfm2png].

A simple method calculating the disparity is to do a horizontal search based on SSD. The SSD is defined by the following, as stated in the project document:

$$S_{tx}(r,\ c) = \sum_{j=0}^{tplRows-1} \sum_{i=0}^{tplCols-1} \left[t(j,\ i) - x\left(r + j - \frac{tplRows}{2},\ c + i - \frac{tplCols}{2} \right) \right]^2$$

First, I create a 3-dimentional array with the first two dimension as the image width and height, and the third dimension as the steps it moved. Then I apply a filter of a certain window size at each pixel, for the purpose of counting all pixels around. Last step is to search the minimum difference between each layer (3rd dimension) and the other image.

The metric I use to evaluate the disparity map is two values:1) percentage of correctly calculated pixels and 2) average of SAD. It is worth mentioning that in the correct percentage part, pixels within 2 disparity are considered as correct. The average of SAD is calculated by SAD over total number of pixels. The higher correct percentage, the lower averaged SAD, the better the disparity mapping.

A range of window size and depth values are tested to achieve best disparity map. The optimal window size for piano-perfect images is 15, and the best depth is 200 (see Figure 1). The plotted intensity is the calculated disparity value. Note that different images have distinct optimal window size and depth values.

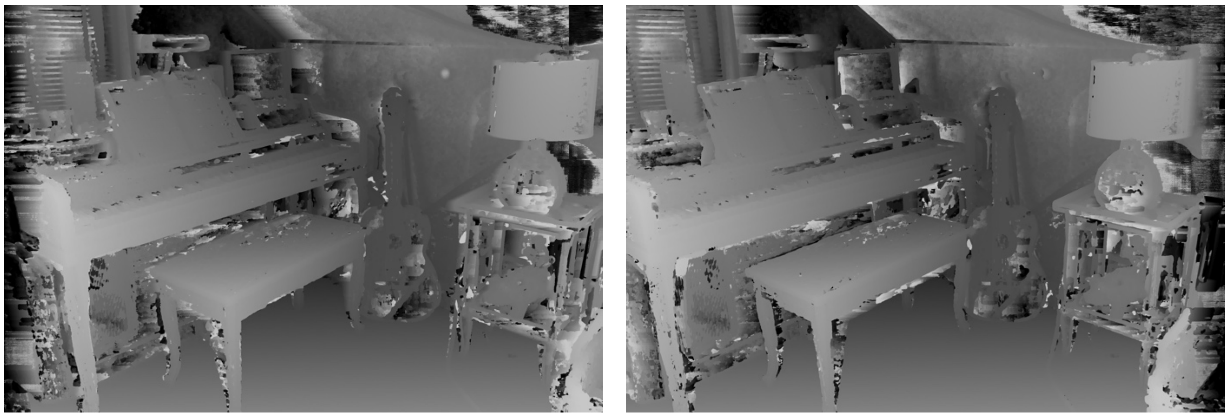
Figure 1. Simple disparity. The left image is mapping left view to right view, and the right image is mapping right view to left view.

2) Advanced algorithms

Advanced algorithms consider both smoothness and data disagreement. Data disagreement term is represented by the SSD calculated in simple disparity. Now there are a few discontinuity preserving interaction penalty functions that might be interesting. The simplest one is just the penalty equals to the absolute distance between labels of neighbor pixels.

However, a discontinuity preserving interaction term should have a bound on the largest possible penalty[Boykov], which prevents over-penalizing sharp jumps[Kim]. Examples include truncated absolute distance and truncated squared distance. These piecewise smooth models encourage labelings consisting of several regions where pixels in the same region have similar labels.

Another type of discontinuity preserving interaction function is Potts model. T is 1 if its argument is true, and otherwise 0. This piecewise constant model encourages labelings consisting of several regions where pixels in the same region have equal labels.
The abovementioned four types of functions are shown below. V stands for the penalty function of two labels alpha and beta. K is the cut-off value for truncated distances in (2) and (3). K is the factor in potts model (4)[Boykov].

(1) Absolute distance: $V(\alpha,\beta) = |\alpha - \beta|$
(2) Truncated absolute distance: $V(\alpha,\beta) = min(K, |\alpha - \beta|)$
(3) truncated squared distance: $V(\alpha,\beta) = min(K, |\alpha - \beta|^2)$
(4) potts model: $V(\alpha,\beta) = K \cdot T(\alpha \neq \beta)$

The a-expansion algorithm described in Boykov paper can be represented in the following:

```
1.  Start with an arbitrary labeling $f$
2.  Set success := 0
3.  For each label $\alpha \in \mathcal{L}$
    3.1.  Find $\hat{f} = \arg\min E(f')$ among $f'$ within one $\alpha$-expansion of $f$
    3.2.  If $E(\hat{f}) < E(f)$, set $f$ := $\hat{f}$ and success := 1
4.  If success = 1 goto 2
5.  Return $f$
```

Maxflow-Mincut algorithm in graph cuts is widely used to minimize energy functions of E(f) as mentioned above. So graph cuts is used to efficiently find f-hat for the key part of each algorithm in Step 3.1. This algorithm allows a large number of pixels to change their labels simultaneously. Note that here we start with not arbitrary labeling, but the label generated from simple disparity. In this study, I adopt the package pymaxflow[Pymaxflow] to calculate the graph cuts and compare the 4 penalty functions in the result. Even with a single cycle of alpha expansion the algorithm provides really good ‘smoothing’ of the disparity map. Due to the high computational cost, this study only use 1-cycle a-expansion and compare the result.

4. Experiment

Recall that out energy function is in the following format:

$$E(f) = \sum_{p \in P} D_p(f_p) + \gamma \sum_{p,q \in N} V_{p,q}(f_p, f_q)$$

Here gamma is a weighing factor of how much penalty is added with regard to the data disagreement term. Various gamma values are examined in all 4 methods. It’s worth mentioning that, for simplicity, only the left to right mapping is used.

1) Absolute distance

The following table shows the result of different factor gamma in the energy function. The column ‘simple’ refers to the simple disparity algorithm in section 2 as a comparison. Absolute distance shows that the factor of 20 gives best matching. See the table below and Figure 2e.

gamma	simple	0.5	1	5	10	20
Correct Perc.	50.35	54.51	55.45	57.62	58.64	58.69
Aver. of SAD	24.89	21.05	20.44	18.47	17.36	15.92

2) Truncated absolute distance
The following table shows the result of 3 different cutoff values (K) with factor gamma from 0.5 to 20. It looks like that the gamma = 20 and K = 50 gives best matching. See the table below and Figure 2f.

	gamma	simple	0.5	1	5	10	20
K = 10	Correct Perc.	50.35	53.98	55.13	57.74	58.85	58.81
	Aver. of SAD	24.89	21.36	20.78	18.84	17.81	17.24
K = 30	Correct Perc.		54.54	55.50	58.57	58.39	58.60
	Aver. of SAD		20.98	20.30	17.65	17.05	16.38
K = 50	Correct Perc.		54.64	55.58	57.88	58.52	58.51
	Aver. of SAD		21.03	20.21	17.51	16.94	15.75

3) Truncated squared distance
Under truncated squared distance model, I set the K as 100, 900 and 2500 to test
The following table shows the result of 3 different cutoff values (K) with factor gamma from 0.5 to 20. It looks like that the gamma = 20 and K = 2500 gives best matching.
Although gamma = 5 and K = 100 or 2500 seems to provide higher correct percentage or lower averaged SAD, the performance around the occluded edge area still has glitches. See Figure 2g.

	gamma	simple	0.5	1	5	10	20
K = 100	Correct Perc.	50.35	57.11	58.35	60.00	59.47	58.95
	Aver. of SAD	24.89	19.08	18.28	16.60	15.86	15.27
K = 900	Correct Perc.		57.27	57.85	58.85	58.98	58.30
	Aver. of SAD		18.49	17.66	14.36	14.02	14.14
K = 2500	Correct Perc.		56.59	57.11	59.84	59.46	58.78
	Aver. of SAD		19.07	17.34	13.84	13.82	14.06

4) Potts model
Potts model is an important discontinuity preserving function, although it's a very simple type of smoothness penalty. Here I test factors from 0.5 to 100. factor 100 provides the best result with regard to occluded edges, even if it's not optimal for correct percentage. The occluded edges are not as good as other penalty models, but it still can work with areas on the top right corner where a much consistent disparity is observed, compared to simple disparity algorithm. See the table below and Figure 2h.

factor	simple	0.5	1	5	10	20	100
Correct Perc.	50.35	51.93	52.15	53.83	54.48	54.82	48.00
Aver. of SAD	24.89	22.31	22.27	21.28	20.89	20.36	18.57

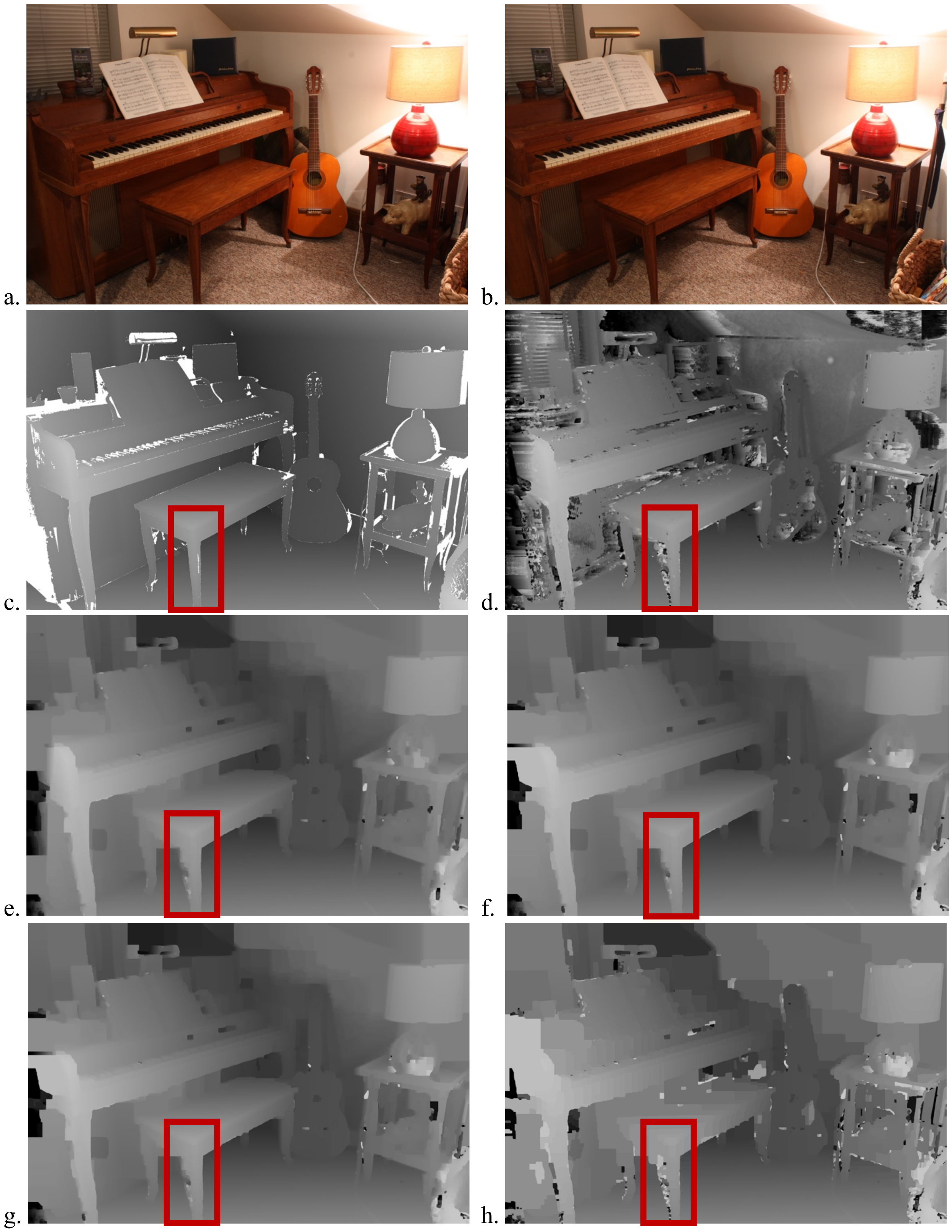

Figure 2. Result comparison. a) left view; b) right view; c) ground truth; d) simple disparity; e) absolute distance; f) truncated absolute distance; g) truncated squared distance; h) potts model. A representative occluded edge area is highlighted in red rectangle.

5) Other images

Each image has its own best window size and depth. For example, the images with less edges and unique features require larger window size and higher depth, like the flower image. The a-expansion algorithm also provides huge improvement in umbrella(Figure 3 top). However, it is not working so well for images like flower with fine features (Figure 3 bottom). The a-expansion cannot restore the little house behind the flowers, the edges of flowers are blurry. I think this is related to ratio of data disagreement and neighbor penalty. Note that other images are using a resizing of one fourth of the original image for faster computation.

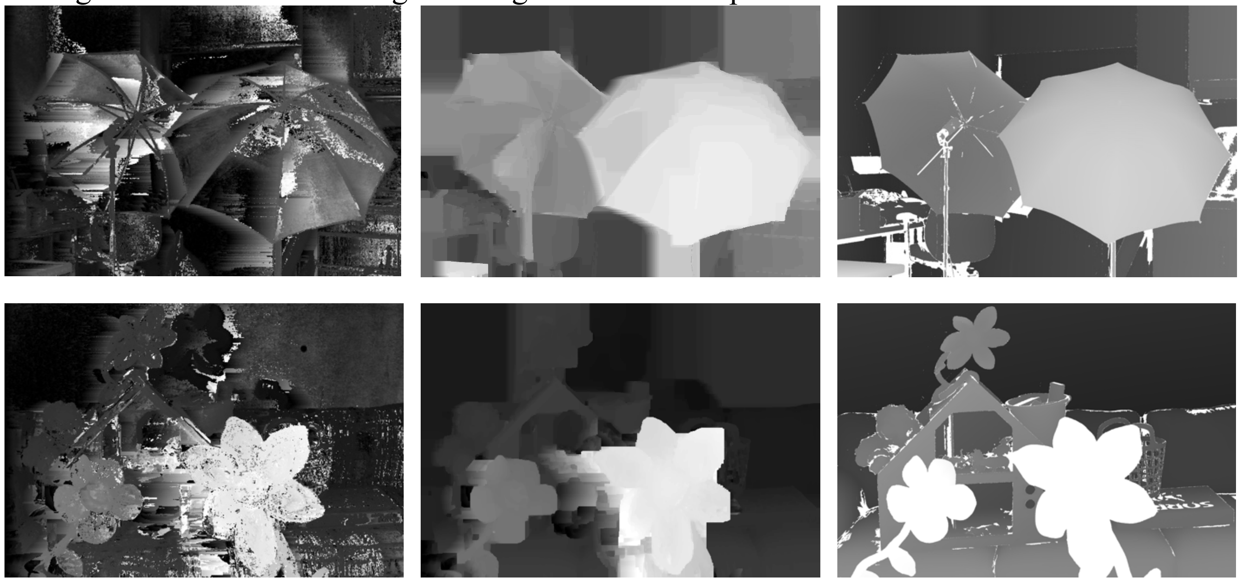

Figure 3. Disparity map for other images. Top is umbrella, bottom is flower.

5. References

1) [Boykov] Y. Boykov, O. Veksler and R. Zabih, Fast approximate energy minimization via graph cuts, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2001, VOL. 23, PAGE 1222.
2) [Deng] Y. Deng, Q. Yang, X. Lin and X. Tang, Stereo correspondence with occlusion handling in a symmetric patch-based graph-cuts model, IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, 2003, VOL. 29, PAGE 1068.
3) [Kim] J. Kim, V. Kolmogorov, and R. Zabih, Visual Correspondence Using Energy Minimization and Mutual Information, Proceedings of the Ninth IEEE International Conference on Computer Vision, 2003 IEEE.
4) [pfm2png] https://convertio.co/pfm-png/
5) [middlebury] http://vision.middlebury.edu/stereo/data/scenes2014/
6) [Pymaxflow] http://pmneila.github.io/PyMaxflow/tutorial.html