CS6476 : Computer Vision, Final Project – Stereo Correspondence

Kevin M. Lee, klee876@gatech.edu

OMSCS – Georgia Institute of Technology – Fall 2022

Introduction

The purpose of this project is to develop a disparity map given two calibrated stereo images. Disparity maps are simply a depiction of depth in images. We are born with the ability to perceive depth as our brain estimates distance with our two eyes. However, in computer images, a singular image solely contains information about pixel intensity / coloration, and does not inherently have depth information. Therefore, we need a pair of cameras with images from left and right angles to compute this data.

Figure 1: 2003 - Cones Dataset from Middlebury Stereo

Data Acquisition / Related Work

Input datasets are gathered from [Middlebury's Stereo Vision page] for testing our algorithms. I compiled the pairs of images, left, right, and ground truth for final evaluation of methods. For stereo images to be useful for disparity, images must first be rectified to where the epipolar lines are matched in the two images [1].

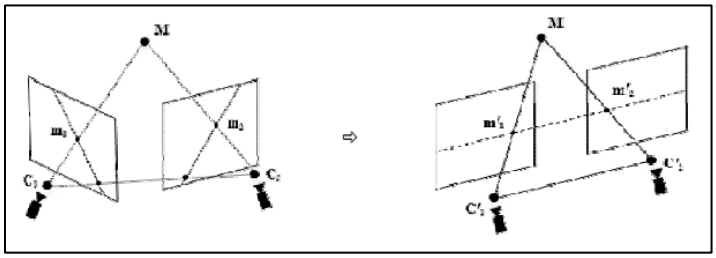


Figure 2: Stereo Camera Image Rectification and Epipolar Line

This Middlebury Stereo dataset is useful because the stereo system is already calibrated, hence images are rectified for us already. A simple disparity map can be determined by scanning along the epipolar line to find the best pixel match and computing the horizontal displacement between the two images known as disparity.

During researching state of the art methods for disparity maps, I learned about three common advanced algorithms for producing these disparity maps to handle occlusion and smoothing. These three algorithms are graph cuts [2][3], k-means image segmentation [4], and semi-global matching [5][6].

Graph Cuts

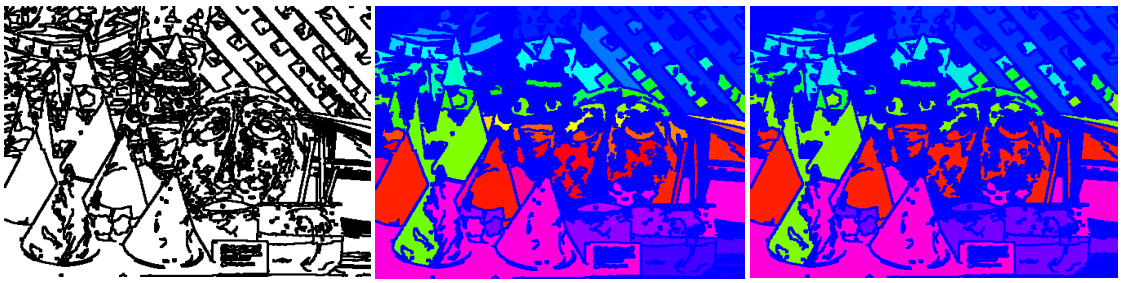
Graph cut consists of developing a graph containing vertices (nodes) and energies (directed edges). The vertices are simply the pixels in the image, and the energy is the sum of the data, occlusion, smoothness, and uniqueness energies as shown.

$$G = (V, E)\ where\ E(f) = E_{data} + E_{occ} + E_{smooth} + E_{unique}$$

From the constructed graph, segmentation is done by cutting the graph between the source and the sink at the maximum flow – equivalent to minimum cut – a theorem from Ford and Fulkerson [3]. In each configuration, α expansion is done to propagate active nodes [2]. The graph cuts are performed until a configuration contains all active nodes or until disparity depth has been reached forming segmentations equal to the maximum disparity level.

K-Means Image Segmentation

Another method involves K-means image segmentation which utilizes canny edges applied on the image to generate groups of pixels using connected component labeling. With these groupings, we apply k-means algorithm which tries to group pixels together based on distance and label coloration [4]. Then, semi-global matching is used to attribute regions of grouped labels with a



disparity value. Below are images generated using "cones" dataset from Middlebury Stereo depicting transition of image to canny edges to connected component labeling to k-means grouping.

Semi-Global Matching

The final method I researched was semi-global matching. The method begins with performing census transformations onto both left and right images. Census transformation is a image processing method to convert to bit-string based on surrounding pixels in a kernel [6]. The equation below is simply, pixels with greater intensities than the center are converted to 1s and lesser intensities to 0s. Then the bits are strung together as the value for the pixel in the census transform

$$C_T(x,y) = \bigotimes_{x \in \mathrm{N}} \begin{cases} 1, if\ \ I(x+i, y+j) \geq I(x_c, y_c) \\ 0, if\ \ I(x+i, y+j) < I(x_c, y_c) \end{cases}$$

255	170	10		1	1	0
140	90	50	⇨	1		0
80	75	100		0	0	1

Figure 3: Example of Census Transformation

In the example above, the center pixel would be written as 11010001. Afterwards a hamming distance is used for initial cost function [5]. Hamming distance is a bitwise operator that compares differences of two values.

$$Example\colon\ H_D(9,14) = \ H_D(1001, 1110) \rightarrow \ H_D(9,14) = 0111 \ \ \rightarrow \ \ H_D(9,14) = 3$$

For the values shown above, matching bits result in 0s and different bits result in 1s with the hamming distance being the summation of 1s in the bitwise computation.

$$C_{total}\big((x,y),d\big) = \sum_{(x,y)\,\in\, W} H_D(C_{T_L}(x,y), C_{T_R}(x - d_1 y))$$

The cost computation is done similar to a normal disparity map, but instead of comparing SAD values, it is a comparison of hamming distance to determine the best disparity between pixels in the left and right image [6]. The above equation shows this where H_D is the hamming distance between census value of a pixel in the left image compared with a displaced census value of a pixel in the right image on an epi-polar line.

$$E(D) = \sum_{p} C_{total}\big(p, D_p\big) + \sum_{q\,\in\, N_P} \begin{cases} P_1 * T, if\ |D_p - D_q = 1| \\ P_2 * T, if\ |D_p - D_q > 1| \end{cases} where\ T[]\ is\ a\ distribution$$

The energy of a disparity map is the summation of all matching costs with penalties, low and high, affecting neighboring pixels where the disparity is equal to 1 or greater than 1, respectively. Energy minimization is done with path traversing to penalize large shifts of disparity in the neighborhood of a pixel. The cost of traversing a path is defined by the following equation [7].

$$L_{r=direction}(p,d) = C(p,d) + min\begin{cases} L_r(p-r,d) \\ L_r(p-r,d-1)+P_1 \\ L_r(p-r,d+1)+P_1 \\ \min_i L_r(p-r,i)+P_2 \end{cases} - \min_i L_r(p-r,i)$$

The aggregate is the summation of all directional costs summed. Then the lowest aggregate sum is chosen returning an image matrix with disparity values.

Methods / Experiment

Prior to any algorithmic approaches to disparity maps, I created a simple disparity map using solely sum of squared differences as suggested by the project documentation. However, multiple sources suggest that sum of absolute differences is computationally faster and generates comparable results, so I migrated to SAD equation shown below instead [8].

$$SAD(x,y,d) = \sum_{(x,y)\in W} |I_L(x,y) - I_R(x-d,y)|$$

I started off trying to understand graph cut algorithm and k-means image segmentation algorithms but fell short. Energy implementation on maxflow graphs and maximum cuts formulas were very intensive [3]. K-means segmentation had vague instructions for reproducing leading me to semi-global matching in the end with not much time remaining [4].

For semi-global matching implementation, I followed the steps as listed in "Related Work – Semi-Global Matching" portion of this paper. Using the approach, I reduced the complexity of some portions of the algorithm for ease of processing on hardware. The census transform was done with a simple 5x5 kernel as compared to some advanced windows mentioned in a paper by Loghman and Kim [6]. Additionally, aggregate path sums were performed over 4 directions – up, down, left, right – as opposed to 16 directional computation done by Hirschmuller [9].

Results

The following four sets of data were downloaded from Middlebury Stereo, 2003 Cones, 2003 Teddy, 2005 Art, 2005 Moebius, using the smallest file sizes for computational speed. All images are roughly in the 350x475 ~ 166k pixels range. Important output images are compiled below.

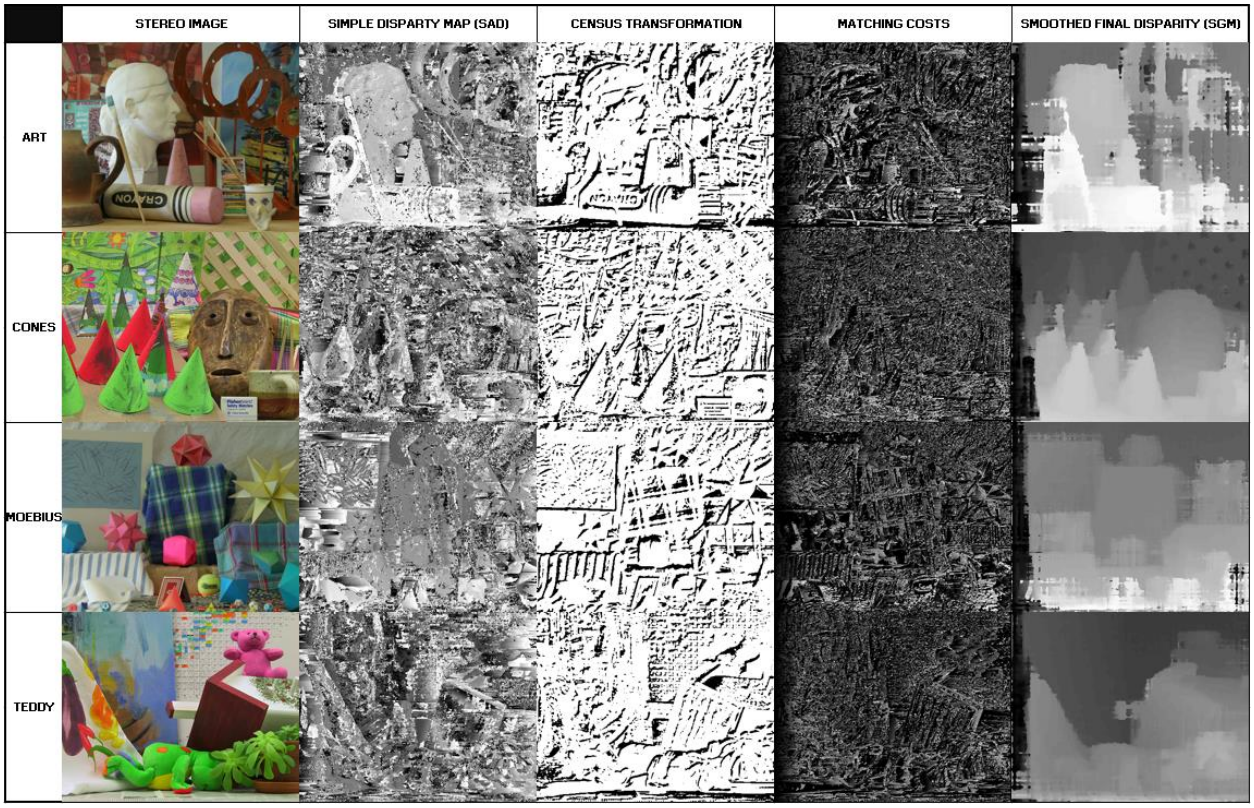


Figure 4: Stereo Image with Outputs

In the simple disparity maps generated using sum of absolute differences, we are generally able to see shapes and figures, but there is a lot of noise with the computation. We see a lot of small artifacts in this disparity map caused by SAD perhaps not detecting the correct best match, or occlusions between left and right images causing false values. Census transformation and matching costs are steps working towards semi-global matching final disparity map and both contain heavy edge information.

By computing path costs of neighboring pixels and aggregating across disparities, we generate a final disparity map that is then smoothed by a median blur to achieve the last column of images. We see black pixels on the left of matching costs and final disparity due to these images utilizing the left stereo image as baseline. This portion does not have a matching component on the right image, hence causes bad disparity computations. This disparity map upon visual inspection seems quite decent as it captures major components of the images and depicts their relative depths appropriately.

However, in evaluation the output final disparities are not as great as visual inspection. Below is a table of computed accuracies in comparison to the ground truth given by the datasets. These were computed by comparing generated disparity maps against ground truth maps using at 10% tolerance on disparity values.

$$Accuracy = \frac{\sum_P \begin{cases} 1, if\ D_P\ is\ between\ [0.90 * D_{GT}, 1.10 * D_{GT}] \\ 0, else \end{cases}}{H_{image} * W_{image}}$$

	SIMPLE DISPARTY MAP (SAD)	FINAL DISPARITY (SGM)	SMOOTHED FINAL DISPARITY (SGM)
ART	26.40%	28.64%	30.12%
CONES	22.03%	60.05%	86.04%
MOEBIUS	41.50%	67.68%	69.24%
TEDDY	22.68%	55.79%	86.39%

Figure 5: Evaluation of Output Disparity Maps

The simple disparity maps using SAD were very low in accuracy in comparison to the ground truth. Disparity maps generated using SGM algorithm improved drastically aside from the 'art' dataset. Finally, applying a median filter improved accuracy largely, possibly due to influencing the left black pixels. I believe the 'art' dataset computes with low accuracy since it contains the most occlusions of the datasets gathered. This is visually seem by streak like regions in the image that don't appear as often on the other disparity maps.

I previously mentioned that these computations were all done with left stereo image as the baseline. The right stereo baseline images are contained in the output folder, but were not shown since there were complications getting the SGM disparity map to compute correctly. The computations, I believe, should be the same, but was not working.

Conclusion

There is still a lot of improvements that can be made on my implementation of semi-global matching for disparity maps. For example, there is a parallax algorithm that helps fill in occluded / black regions in the disparity map [5]. Another shortcoming is the lack of variable implementation. During my research, I did not come across any algorithms to predict input parameters such as disparity levels, matching cost computation kernel size, or penalty costs for semi-global matching. Instead of tuning manually, implementing these should improve the accuracy as a systematic way to determine these input parameters using image information should be optimal.

References

[1] Kang, YS., Ho, YS. (2011). Efficient Stereo Image Rectification Method Using Horizontal Baseline. In: Ho, YS. (eds) Advances in Image and Video Technology. PSIVT 2011. Lecture Notes in Computer Science, vol 7087. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-25367-6_27

[2] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max- flow algorithms for energy minimization in vision," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 9, pp. 1124-1137, Sept. 2004, doi: 10.1109/TPAMI.2004.60.

[3] Vladimir Kolmogorov, Pascal Monasse, and Pauline Tan, *Kolmogorov and Zabih's Graph Cuts Stereo Matching Algorithm,* Image Processing On Line, 4 (2014), pp. 220-251. https://doi.org/10.5201/ipol.2014.97

[4] E. Ko and Y. -S. Ho, "Disparity Map estimation using semi-global matching based on image segmentation," *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1-4, doi: 10.1109/APSIPA.2016.7820850.

[5] Jian Zhang and Jing Huang 2021 J. Phys.: Conf. Ser. 2010 012037

[6] M. Loghman and J. Kim, "SGM-based dense disparity estimation using adaptive Census transform," *2013 International Conference on Connected Vehicles and Expo (ICCVE)*, 2013, pp. 592-597, doi: 10.1109/ICCVE.2013.6799860.

[7] Rostam Affendi Hamzah, Haidi Ibrahim, "Literature Survey on Stereo Vision Disparity Map Algorithms", Journal of Sensors, vol. 2016, Article ID 8742920, 23 pages, 2016. https://doi.org/10.1155/2016/8742920

[9] H. Hirschmuller, "Stereo Processing by Semiglobal Matching and Mutual Information," in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328-341, Feb. 2008, doi: 10.1109/TPAMI.2007.1166.