Georgia Tech's CS6476 Final Project: Stereo Correspondence

Trevor Davidson
tdavidson30@gatech.edu

December 7th, 2023

Introduction

All cameras operate as two-dimensional sensors: real-world scenes are projected onto a plane, then captured and displayed as arrays of pixels. Although this can capture details effectively on the horizontal and vertical axes, information about depth is completely lost. Without prior information about the scene and camera, the image features can exist at any point along the ray from connects the pixel and the camera sensor. However, with stereo correspondence techniques, combining an additional similar image with the original image allows algorithms and computers to determine relative depth of objects in a scene. These algorithms have been the subject of significant research with various approaches [1].

The purpose of this work is to implement a basic stereo matching algorithm using the Sum of Squared Differences cost function and an advanced energy minimization algorithm, comparing the results against each other and ground truth images.

Recent Work

Hamzah and Ibrahim [1] cover a variety of approaches to stereo correspondence. Although not preferred methods relative to graph cut and energy minimization algorithms, feature-based matching techniques are relatively simple to implement. However, these applications generally have low accuracy. Since they require correlated points in the second image, feature-based methods struggle with occluded areas or features. Additionally, textureless areas do not provide enough information for a confident match and struggle to accurately pinpoint changes [1].

Feature-based implementations are generally local methods: there are no constraints on the disparity values related to other parts of the image. In recent years, global methods have become significantly more prominent due to their higher accuracy or efficiency [1]. Often these methods use graph cuts or belief propagation. For example, Lu [2] proposed an improved graph cut algorithm using census transform to optimize the data term in the associated energy function. Their approach is an application of the method described by Boykov [3] where simple template matching can be used to create hard foreground/background constraints. The constraints can then be used to create the weights of edges between the pixels and the source and sink nodes before applying the minimum cost cut. Kolmogrov [4] worked with a similar graph cut technique, but also applied four different smoothness terms (data, smoothness, occlusion, and uniqueness) to improve accuracy in occluded areas [1, 4].

Dynamic programming is another global technique. Using a predefined cost function, the core dynamic programming algorithm attempts to traverse horizontal scan lines, resulting in polynomial complexity [1]. However, recent methods have been developed using minimum spanning trees (MST). Huang and Zhang used MST with belief aggregation and belief propagation. Starting with an initial disparity belief, an initial, hybrid MST is determined based on disparity and color distance using two sequential passes, with greater aggregated belief for pixels with closer color and disparity to their neighbors. This is followed by calculating an MST only determined by color distance and using belief propagation to refine the disparity map [5]. Veksler combined both dynamic programming and MST to improve vertical consistency between scanlines by programmatically traversing through the tree created from a 4-connected grid of all pixels, improving over scanline-based methods [6].

Methods

Stereo block matching was implemented using the methodology described in [7] with the formula described in the assignment description, shown in Figure 1. For window at pixel ji, S_{tx} is calculated along the horizontal scanline. The disparity is then found at the, $S_{tx}(r)$ value along the scanline. This is done for every row to create three disparity maps: left to right, right to left, and a centered disparity that is the matching points between the two. For this analysis, all windows were set to 5 rows and 5 columns. The calculated disparity values were also bounded between 0 and a specific upper bound for each dataset.

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

Figure 1. Sum of Squared Differences Equation

For an energy minimization algorithm, an MST-traversing dynamic programming algorithm was developed. The tested formulation is inspired by Veksler [6]. Grayscale images were used for this section to reduce computation time.

First, we need to define the MST. As defined in Veksler, “Let $G' = (V, E')$ be a graph with vertices V consisting of all the image nodes and edges E' consisting of all the edges between the nearest neighbouring pixels, that is E' is simply the standard 4 connected grid. For each pair of neighbouring pixels p and q, assign weights $v_{pq}|I(p) - I(q)|$to the edge between p and q. Construct the minimum spanning tree of G' and let that tree to be the tree structure for our algorithm”. This tree is named the “minimum intensity tree”, or MID tree [6].

Once the tree is formed, a root node $r \in V$ is chosen. The minimum energy is independent of the particular choice of r [6], so the top left corner of the image is chosen. The disparity map d is also initially defined as all zeros.

From here, there are two main departures from Veskler. Veskler specifies that the first pass through the tree begins by evaluating for the optimal disparity assignment and energy at leaf nodes and progressing up the tree. The equations for this are specified in section 2.2, equations 3 and 4 [6], and shown below:

$$E_v\left(d_{p(v)}\right) = \min_{d_v \in D}\left(m(d_v) + s\left(d_v, d_{p(v)}\right) + \sum_{w \in C_v} E_w(d_v)\right)$$

$$L_r^* = arg \min_{d_r \in D}\left(m(d_r) + \sum_{w \in C_v} E_w(d_r)\right)$$

Where m is defined as follows:

$$m\left(d_p\right) = \min\left\{\left|L(p) - R\left(p - d_p\right)\right|, \tau\right\}$$

Where $L(p)$ is the pixel intensity in the left image and $R(p)$ is the pixel intensity in the right image. τ is used to make $m\left(d_p\right)$ more robust to outliers [6].

However, this analysis adjusts the energy equation for this pass to ignore the $s(d_v, d_{p(v)})$ term for only the first pass. This is because a naïve initial guess is implemented: the smoothing term would then use this as part of the energy calculation and influence the future map dramatically. This term is re-added for the second pass, back down the MST. Second, the smoothness penalty in the energy calculation is done in Veskler in the following way:

$$s_P(p_d, q_d) = \begin{cases} 0, & if\ d_p = d_q \\ w_{pq}, & otherwise \end{cases}$$

This was chosen to reduce the complexity of the calculation, since the original energy call only allows for the smoothness to be related to the parent term [6]. However, this choice disregards the possibility that the added point could be closer in disparity to its children than its parents. This analysis modifies that assumption accordingly and evaluates the smoothness penalty as such:

$$s_{penalty}(p_d, q_d) = \sum_{q \in C_v, P_v} \begin{cases} 0, & if\ d_p = d_q \\ w_{pq}, & otherwise \end{cases}$$

This allows the smoothness penalty to apply more significantly if the disparity does not agree with any child or parent and less significantly if it agrees with both sets.

Evaluation of the generated disparity maps was done with the BMP measure, as described by Merrouche [8], which was lifted from the Middlebury stereo benchmark [9]. This is calculated as follows:

$$BMP = \frac{1}{N}\sum_{(x,y)} \varepsilon(x, y)$$

Where $\varepsilon(x, y)$ is a binary function defined as:

$$\varepsilon(x, y) = \begin{cases} 1, & |D_t(x, y) - D_e(x, y)| > \delta \\ 0, & |D_t(x, y) - D_e(x, y)| \leq \delta \end{cases}$$

and δ is set to one pixel [8, 9].

Results

The images analyzed in this report were also from the Middlebury dataset, specifically the map, sawtooth, and cones datasets [10]. Figures 2-4 show the disparity maps produced each algorithm and the ground truth from that dataset. Figure 5 shows the centered disparity image for the Map dataset, which shows the occlusions for both the left and right SSD disparity maps.

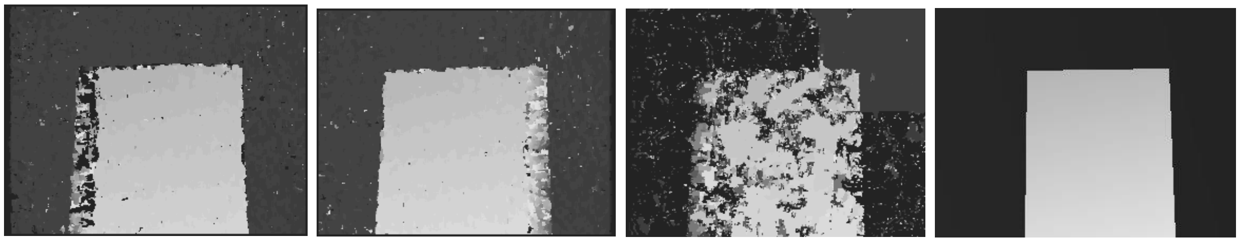
Figure 2. Results from algorithms on the Map dataset. Left to Right: Left Disparity Map (SSD), Right Disparity Map (SSD), Energy MST Disparity Map, Ground Truth (left)

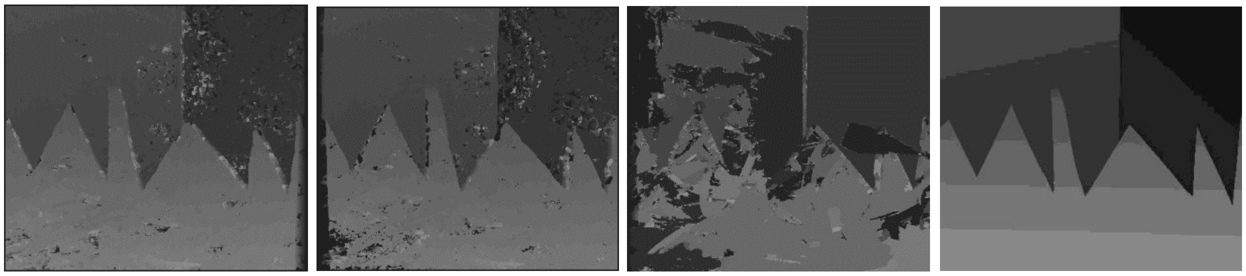
Figure 3. Results from algorithms on the Sawtooth dataset. Left to Right: Left Disparity Map (SSD), Right Disparity Map (SSD), Energy MST Disparity Map, Ground Truth (left)

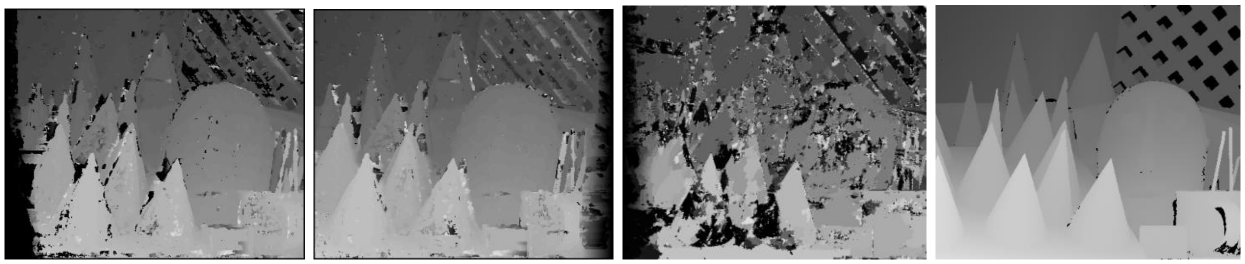
Figure 4. Results from algorithms on Cones dataset. Left to Right: Left Disparity Map (SSD), Right Disparity Map (SSD), Energy MST Disparity Map, Ground Truth (left)

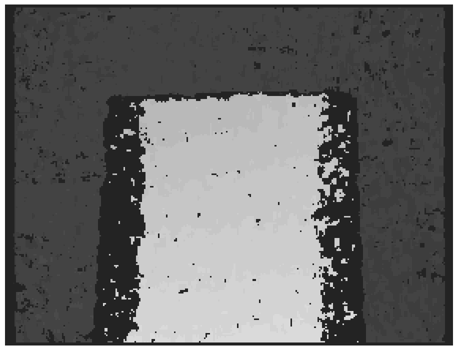
Figure 5. Combined (centered) disparity map for the Map Middlebury dataset using SSD

Dataset	*Left Disparity Map (SSD)*	*Right Disparity Map (SSD)*	*Energy MST*
Map ($w_{pq} = 42, \tau = 21, d_{max} = 22$)	67.353	61.918	38.714
Sawtooth ($w_{pq} = 42, \tau = 21, d_{max} = 30$)	57.500	53.361	70.559
Cones ($w_{pq} = 25, \tau = 21, d_{max} = 60$)	96.547	94.553	93.839

Table 1. BMP Error % for all used datasets with Energy MST parameters

As expected, the SSD models struggle with occlusions and low-texture regions despite using color images. These are particularly evident in the cones, where the areas just to the left of the cones show low disparity values, and in the map, where the left edge disparity is very badly captured.

However, both SSD images display more smoothness than the Energy MST algorithm. This is especially evident in the cones dataset but present in each: large swaths of poorly calculated disparity exist. Additionally, the BMP values shown in Table 1 are far too high and suggest the coded algorithm is not functioning correctly; the results from [8] show single digit error rates (albeit with better algorithms) when compared to the ground truth images. These issues are likely due to varying intensities on the surface of and around the cones. Intensities and additional features assist a matching algorithm like SSD, but they can confuse the Energy MST algorithm, which only compares individual pixels instead of feature windows. A way to combat this in future iterations would be to use a sliding window in the Energy MST algorithm and improve the integration of the smoothness term.

There is one success with the Energy MST algorithm: similar surfaces have very consistent disparity values (excepting the obvious holes/issues). The BMP error and resulting image from the map dataset shows this well: areas that are approximately correct are a very consistent depth. This suggests that the energy function is applying the smoothing and data terms correctly in some instances and the model may only require refinement of parameters. With additional algorithmic refinement to increase processing speed, the color images may improve the model's efficiency.

Conclusions

A dynamic programming algorithm relying on an MST is proposed. Results generated using stereo pairs from the Middlebury datasets show the Energy MST dynamic programming algorithm shows some minor promise in key areas, but requires greater refurbishment in a number of areas. Future work on this algorithm should reexamine many parts of the MST algorithm, refining the smoothness term in the energy minimization, parameters for the data and smoothness terms, and the initialization of the image.

References

[1] Hamzah, Rostam & Ibrahim, Haidi. (2015). Literature Survey on Stereo Vision Disparity Map Algorithms. Journal of Sensors. 2016. 1-23. 10.1155/2016/8742920.

[2] Baoli Lu, Liang Sun, Lina Yu, Xiaoli Dong, An improved graph cut algorithm in stereo matching, Displays, Volume 69, 2021, 102052, ISSN 0141-9382, https://doi.org/10.1016/j.displa.2021.102052. (https://www.sciencedirect.com/science/article/pii/S0141938221000627)

[3] Boykov, Y., Funka-Lea, G. Graph Cuts and Efficient N-D Image Segmentation. Int J Comput Vision 70, 109–131 (2006). https://doi.org/10.1007/s11263-006-7934-5

[4] 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

[5] Xiaoming Huang, Yu-Jin Zhang, An O(1) disparity refinement method for stereo matching,Pattern Recognition, Volume 55, 2016, Pages 198-206, ISSN 0031-3203, https://doi.org/10.1016/j.patcog.2016.01.025., (https://www.sciencedirect.com/science/article/pii/S0031320316000467)

[6] O. Veksler, "Stereo correspondence by dynamic programming on a tree," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), San Diego, CA, USA, 2005, pp. 384-390 vol. 2, doi: 10.1109/CVPR.2005.334.

[7] Kris Kitani, Stereo Matching 16-385 Computer Vision, Carnegie Mellon University, https://www.cs.cmu.edu/~16385/s17/Slides/13.2_Stereo_Matching.pdf

[8] Merrouche, Saad & Andrić, Milenko & Bondzulic, Boban & Bujaković, Dimitrije. (2020). Objective Image Quality Measures for Disparity Maps Evaluation. Electronics. 9. 10.3390/electronics9101625.

[9] Middlebury Stereo Evaluation – Version 3, https://vision.middlebury.edu/stereo/eval3/

[10] Middlebury Stereo Datasets, https://vision.middlebury.edu/stereo/data/