Stereo Correspondence with Occlusions using Graph Cuts

EE368 Final Project

Matt Stevens
mslf@stanford.edu

Zuozhen Liu
zliu2@stanford.edu

I. Introduction and Motivation

Given two stereo images of a scene, it is possible to recover a 3D understanding of the scene. This is the primary way that the human visual system estimates depth. This process is useful in applications like robotics, where depth sensors may be expensive but a pair of cameras is relatively cheap. In order to construct depth maps from stereo images, we need to first solve the stereo correspondence problem. The stereo correspondence problem has traditionally been one of the most studied topics in image processing and computer vision[1] but it is still an area of active research.

Graph cut algorithms proposed by Boykov et. al.[2] represent a framework that reduces energy minimization problems to network flow problems. Since a number of problems in computer vision can be formulated as some form of energy minimization, graph cut algorithms have a wide range of applications such as: image restoration, stereo correspondence and image segmentation.

In this project, we combined our interests to implement a graph cut algorithm for stereo correspondence[3] and performed evaluation against a baseline algorithm using normalized cross correlation (NCC) across a variety of metrics. Specifically, we investigated on the effectiveness of labeling disparities and handling occlusions for the graph cut algorithm. We used a pre-aligned stereo image dataset with ground truth disparities from Middlebury College to benchmark performance[4].

II. Related Work

After decades of active research, an enormous amount of different methods and techniques have been proposed to solve the stereo correspondence problem[5]. By and large, these methods can be categorized into two main approaches: local methods and global methods.

Local methods tend to consider a small window around each target pixel and encode certain smoothing constraints into a cost function computed over the entire window area. By minimizing the cost function, a best disparity value is selected for the given target pixel. Despite various optimized methods proposed such as Adaptive Support Weight [6], Slanted Window[7], local methods still face the limitation of handling occlusion in matching process. Our baseline NCC method is also a naive local method.

Global methods are different from local approaches in that the smoothness and occlusion constraint can be directly encoded in a global energy function of a given disparity map. The goal is to use various optimization techniques to achieve a disparity map that minimizes the global energy function. An exact minimization of the energy function is NP-complete. However, with certain selection of a smoothness term, the minimization can be computed efficiently via Dynamic Programming[8].

Graph cut algorithms also belong to global methods and can be applied to optimizing the energy function in a more generalized way. In recent years, variations of the original graph cut algorithm[3] have been proposed to either improve performance or runtime such as LogCut[9]. However, alpha-expansion, a core step of graph cut, remains widely adopted as the optimization engine for later algorithms. Therefore, the goal of this project is to study the fundamentals of a class of graph cut algorithms and develop insights on how the algorithm handles occlusion, which remains a challenge in local methods.

III. Graph Cut

In this section, we will explain the workflow of graph cut algorithm in more detail. First step is to reformulate the problem as an energy minimization problem on a Markov Random Field (MRF). Next step is to iteratively perform alpha-expansion by using a min-cut minimization to achieve an optimal labeling configuration. An overview of the workflow is summarized in the figure below.

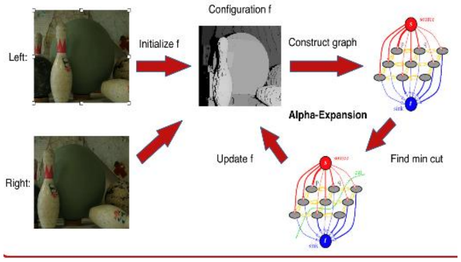


Fig. 1. **Workflow diagram**

A. *Markov Random Field*

If we consider each pixel in the reference image as a node in the MRF, we can take its unique disparity value to be its label. Now since a MRF only has a finite set of labels, we need to select a discrete set of disparity values to consider. In our implementation, we computed the minimum and maximum disparity values d_min, d_max from the given ground truth disparity map for every input image pair and took [d_min, d_max] as the interval for all possible disparity values.
Besides defining the label set, we also need to select a neighboring system for MRF and our implementation adopted a 4-neighbor system. With a complete definition of the MRF for stereo correspondence, our goal is to find a best labeling configuration for the MRF that minimizes certain energy function.

B. *Energy Function*

Denote the set of possible labeling configuration as:

$$A = \{(p, q) \mid p_y = q_y, 0 \leq q_x - p_x < k\}$$

For any unique disparity configuration f on the entire image, the energy function is defined as follows:

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

- The data term represents the cost of intensity difference between corresponding pixels.
- The occlusion term imposes a constant penalty cost for occluded pixels
- The smoothness term adds a constraint to make neighbor pixels have similar disparities. Specifically, there is a constant cost if one labeling assignment is in the configuration and its neighbor pixel's assignment with the same disparity value is not in the configuration.

C. *Alpha-Expansion*

We first define alpha-expansion as the set of all possible assignments for next update. The constraint is that this set is a union of previous active assignments and assignments with disparity alpha. In other words, at every iteration, we could only either cancel active assignments or make new assignments with disparity alpha. The algorithm would iterate until no new assignments can lower the energy function.

After initializing a unique configuration, we can then iteratively refine our configuration by performing alpha-expansion to minimize the global energy function. We perform an alpha expansion for every value of alpha. We continue to cycle through values of alpha until the model no longer changes, meaning that the algorithm has arrived at its local optimum, and we terminate. The question is then, how to find a configuration that minimizes the energy function within an alpha-expansion in step 3.1. This leads to our discussion on the min-cut graph.

D. *Min-cut Graph*

At each iteration, we can define the following notation:

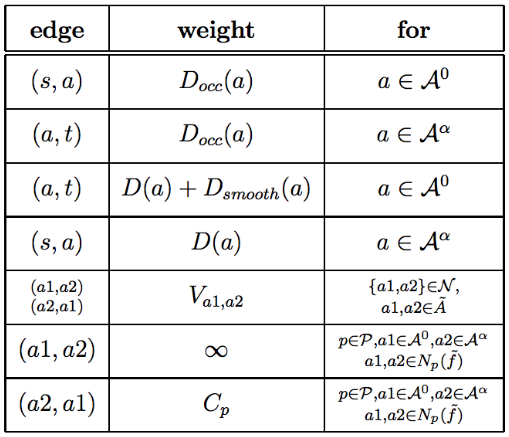
TABLE I. **COST MODEL TERMINOLOGY**

A^0	set of active assignments in current configuraiton
A^α	set of potential assignments with disparity alpha
$D(a)$	data cost of an assignment
$D_{occ}(a)$	occlusion cost of an assignment
$D_{smooth}(a)$	smoothness cost of an assignment
V	smoothness constant
C_p	occlusion constant

The steps to construct a min-cut graph is decomposed into:

1. Create a source and a sink node.
2. Insert each correspondence assignment in set as a node into the graph
3. Insert edges and its weight between two nodes into the graph based on the table below

TABLE II. **EDGE WEIGHTS**



edge	weight	for
(s, a)	$D_{occ}(a)$	$a \in \mathcal{A}^0$
(a, t)	$D_{occ}(a)$	$a \in \mathcal{A}^\alpha$
(a, t)	$D(a) + D_{smooth}(a)$	$a \in \mathcal{A}^0$
(s, a)	$D(a)$	$a \in \mathcal{A}^\alpha$
$(a1{,}a2)$ $(a2{,}a1)$	$V_{a1,a2}$	$\{a1{,}a2\}\in\mathcal{N}$, $a1{,}a2\in\tilde{A}$
$(a1, a2)$	∞	$p\in\mathcal{P}{,}a1\in\mathcal{A}^0{,}a2\in\mathcal{A}^\alpha$ $a1{,}a2\in N_p(\tilde{f})$
$(a2, a1)$	C_p	$p\in\mathcal{P}{,}a1\in\mathcal{A}^0{,}a2\in\mathcal{A}^\alpha$ $a1{,}a2\in N_p(\tilde{f})$

A detailed proof of how a min-cut in this graph is equivalent to optimal configuration that minimizes the energy can be found in [3].

E. *Update Configuration*

Once we have constructed the graph discussed above, we would then be able to invoke a network flow routine to compute the min-cut on the graph. From the min-cut, we can retrieve new active assignments and update the configuration accordingly.

IV. EXPERIMENTAL RESULTS

For evaluation, a disparity dataset containing images and ground truth disparity values was obtained from the Middlebury Stereo Vision Page. Occlusions in the ground truth disparity map were not provided and were calculated based on physical principles. A pixel that corresponds to a pixel outside the other image's boundary is labeled as occluded. A pixel that corresponds to a pixel with a different disparity (with a threshold of 2) is also labeled as an occlusion. These ground truth disparity maps allow us to objectively evaluate the performance of the stereo algorithms presented here. Results of our stereo algorithms are shown in Fig. 2.

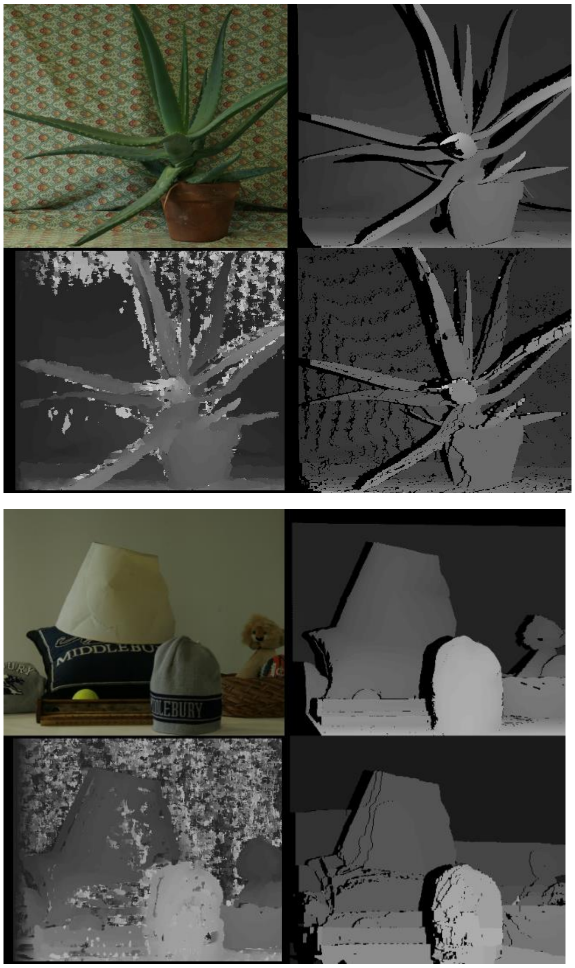


Fig. 2. **Disparity algorithm results.** Upper left: source image, upper right: ground truth, lower left: NCC algorithm, lower right: graph cut algorithm

For the normalized cross correlation algorithm, a quick visual comparison of the algorithmically generated disparity maps to the ground truth reveals that NCC captures the general sense of the scene, but also introduces a significant amount of noise. Noise occurs when the template match is too weak to trigger a response that stands out. This occurs especially on smooth surfaces like in the second set of images, leading to a great deal of noise in these regions. In the first image, the repetitive pattern means that multiple locations in the image will produce a response from the template, leading to the bimodal response that is observed. However, considering the simple nature of the algorithm, the results are reasonable.

Analyzing the graph cut disparities reveals three main traits: the graph cut algorithm approximates the values well, but it introduces sharp jumps in disparity value, and it also introduces thin lines of occluded pixels. These artifacts are somewhat inherent to the formulation of the algorithm. As for the lines, there is a strict constraint that corresponding pixels must correspond only with each other. This means that at the boundary of some region of constant disparity, either the right or left side must be occluded. Short of loosening the problem constraints, there is no real way to solve this issue. The sharp jumps are a result of the smoothing term in the model, which encourages piecewise-constant regions, and thus sharp boundaries at the edges of these regions. A reduction in the smoothing parameter would decrease these sharp jumps, but also increase the level of noise from mislabeled disparities.

Although the qualitative results of our graph cut implementation look promising, it is important to look at quantitative results as well. We evaluated our graph cut algorithm and our baseline algorithm on a number of metrics, looking for two things - how accurately does the algorithm label disparities, and how accurately does it label occlusions. Labeling occlusions is a binary classification problem and can be assessed using a number of metrics. We measured false positive rate and false negative rate, as in [3]. For overall accuracy, we measured "Gross Errors", the percent of pixels that do not match within a certain threshold, also used in [3]. In addition, we measured bias and R^2 coefficient over pixels unoccluded in both test and ground truth images to measure how accurately the algorithms predict the disparity and what effect outliers have.

In both accuracy and occlusion labeling, the normalized cross-correlation algorithm performed quite poorly, as is expected of an algorithm with no filtering or smoothness constraints.

Our implementation of the graph cut algorithm did not perform as well as the original algorithm we intended to adapt in terms of overall accuracy, getting a significantly lower score in terms of gross error percentage. It is worth noting that the dataset used for testing contains many difficult images, with untextured regions and occlusions. The dataset used for testing the reference algorithm was highly textured with plenty of visual cues for judging disparity. So some amount of the poor relative performance of our graph cut algorithm is likely due to a more difficult test set. And our version of grab cut missed fewer occlusions, so it did improve in one area.

Below the results of our benchmarks are summarized. The following are mean values with and standard deviation across our dataset for the NCC method, our graph cut method (GC) and the graph cut method of [3] (GC Ref.):

TABLE III. **BENCHMARK RESULTS**

	Gross Errors	**False Neg. Rate**	**False Pos. Rate**	**Bias (px)**	**R^2**
NCC	72±5%	68±12%	3.7±0.8%	0.8±3.8	0.25±0.53
GC	16±12%	26±17%	4.8±3.7%	-1.0±1.8	0.76±0.31
GC Ref.[a]	1.9%	42.6%	1.1%	-	-

[a] Results measured on a different dataset, see [3]

The positive bias of the NCC algorithm indicates that it is mislabeling values as high disparities. This is supported by a visual inspection of the test images. and the negative bias of the graph cut algorithm indicates that is not detecting values with high disparities, or the foreground of the scene. This interpretation is also supported by the first test image shown earlier. It may be that the very most foreground elements tend to protrude sharply, and the smoothing of the algorithm is clipping these high disparities to lower values.

The graph cut algorithm was highly variable in its performance, which suggests that there is more work to be done in tuning the algorithm to suit all scenarios. In the best cases, the algorithm had excellent performance, on par with the standard, however in many cases there were large errors, particularly for scenes with either complicated or ambiguous geometry.

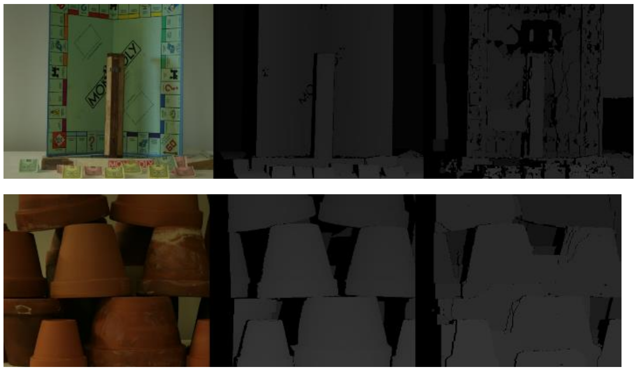


Fig. 3. **Modes of failure.** Left: original, middle: ground truth, right: graph cut results. The combination of ambiguous geometry, untextured surfaces, and occlusion leads to large errors. The image above features untextured surfaces on the wall and the board, as well as an occluding object in the middle and many occlusions in the foreground. The image below features ambiguous features whose depth cannot necessarily be resolved by stereo correspondences.

V. Evaluation of System Parameters

As mentioned in section III, the graph cut algorithm balances three different cost functions, one for pixel matching, one for occlusion, and one for smoothing. There are two parameters to control the relative importance of these three functions: C_p and V. C_p represents the cost of occluding a pixel, and V represents the cost of a discontinuity in disparity values. Both of these parameters are scaled to the pixel matching cost function, which in turn depends on natural image statistics. The original authors parameterized both C_p and V in terms of a different cost, lambda, keeping their relationship fixed, close to one to one. In our work, we investigated the tradeoff between smoothing, occlusion error, and pixel matching. Fig. 4 shows the relationship between the importance of occlusions and smoothing and the error of the algorithm. If the occlusion cost is too low, then the entire image will be labeled as an occlusion, leading to multiple types of errors. If too much emphasis is placed on avoiding occlusions, the algorithm will attempt to label occluded pixels and introduce errors. Similarly, if too little smoothing is applied, the disparity labels will be noisy. And if too much smoothing is applied, diminishing returns set in, and the results start to look visually less appealing.

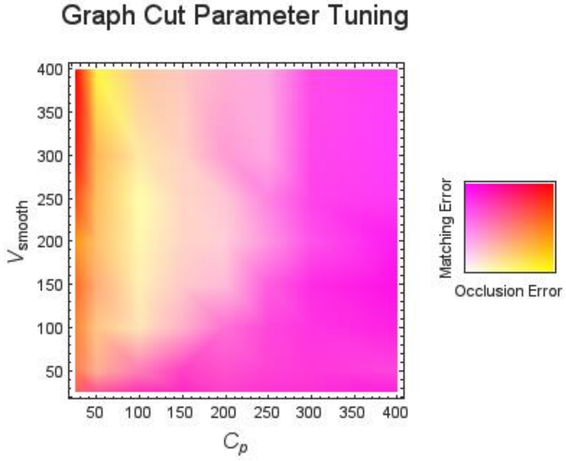


Fig. 4. Parameter tuning for graph cut - white areas are desirable and colored areas contain errors

C_p of 200 and V of 200 were chosen to be an optimal trade-off between mislabeling occlusions and mislabeling disparities.

The NCC algorithm has one parameter, window size. The larger the window size, the more regularization is applied, and the smoother the image. However, a large window size also makes it difficult to detect sharp edges where different parts of the template may match differently. We found an optimal window size to be 9 pixels.

VI. Comparison to Alternative Approaches

In 2002, graph cut algorithms were the state of the art for the stereo correspondence problem [1], but they have since been supplanted by methods involving convolutional neural networks. However, MRF methods, the same basis underlying graph cuts, still have state of the art performance [11].

Graph cuts, as a global optimization method, are much more powerful than local methods. That was illustrated in the extreme by our two algorithms, the NCC algorithm which is purely local, and the graph cut algorithm which is global. However, the graph cut algorithm took roughly 20 times longer to run than the NCC algorithm. Even with performance optimizations, it is a slow technique, and is ill-suited to many applications of stereo imaging like robotics.

VII. Discussion, Limitations, and Future Work

An important area of future work would be better quantitative analysis of performance. The data used for analysis in [1] and [3], the primary sources for this paper, was only available with a left disparity map, and not a right one, meaning it was not possible to calculate occlusions given this data. Furthermore, the method for calculating occluded and texture-less regions in not explicitly mentioned in the sources. The authors of [1] provide an SDK for testing, but there was not enough time in the scope of the project to integrate with this SDK. A best effort was made to quantify our performance in a way that would be relatable to other algorithms, however there is still work to be done.

In 2002, graph cut algorithms were the state of the art for the stereo correspondence problem [1], but they have since been

supplanted by methods involving convolutional neural networks. However, MRF methods, the same basis underlying graph cuts, still have state of the art performance [11]. But if we accept that our goal is not state of the art performance, there are still techniques to improve results within the realm of graph cuts.

The graph cut optimization framework is distinct from the cost functions used in the model, and improved cost functions can give better results. The pixel matching cost E_{data} can have a significant effect on the performance of the algorithm [12]. One such cost function is described by Birchfield and Tomasi, and is invariant to image sampling [13], which gives better matching results near sharp edges in images where intensity values may differ greatly. The algorithm works by examining the neighboring pixels of corresponding pixels to determine how quickly the image is varying, and to lower the dissimilarity when the image intensity is rapidly changing. This matching technique was implemented in our system, but was found not to improve results. However, this matching cost uses absolute error instead of squared error, so the smoothing and occlusion parameters had to be adjusted, and it is possible that they were not adjusted optimally. In the future, more sophisticated cost functions such as NCC or cost functions using filters could be investigated for potential improvements [12].

REFERENCES

[1] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms,” Int’l J. Computer Vision, 2002.

[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 (PAMI), vol 26, no.9, pp 1124-1137, Sept 2004.

[3] V. Kolmogorov and R. Zabih, "Computing visual correspondence with occlusions using graph cuts", In ICCV, volume II, pages 508–515, 2001.

[4] Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nešić, N., Wang, X., & Westling, P. (n.d.). High-Resolution Stereo Datasets with Subpixel--Accurate Ground Truth. Lecture Notes in Computer Science Pattern Recognition, 31-42.

[5] M. Bleyer and C. Breiteneder, “Stereo Matching - State-of-the-Art and Research Challenges,” in Advanced Topics in Computer Vision. Springer, 2013, pp. 143–179

[6] K.-J. Yoon and I.-S. Kweon. “Locally Adaptive Support-Weight Approach for Visual Correspondence Search”. In CVPR, pp.924–931, 2005.

[7] M. Bleyer, C. Rhemann, and C. Rother, “Patchmatch Stereo—Stereo Matching with Slanted Support Windows,” Proc. British Machine Vision Conf., 2011

[8] I. J. Cox, S. L. Hingorani, S. B. Rao, and B. M. Maggs, " A maximum likelihood stereo algorithm.", in CVIU, 63(3):542–567, 1996.

[9] Victor Lempitsky, Carsten Rother, and Andrew Blake, “Logcut-efficient graph cut optimization for markov random fields,” in Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on. IEEE, 2007, pp. 1–8.

[10] J. Zbontar and Y. LeCun. “Stereo Matching by Training a Convolutional Neural Network to Compare Image Patches”, in CVPR, 1510.05970, 2015.

[11] M. G. Mozerov and J. V. D. Weijer, “Accurate Stereo Matching by Two-Step Energy Minimization,” *IEEE Transactions on Image Processing*, pp. 1153-1163

[12] H. Hirschmuller and D. Scharstein. “Evaluation of cost functions for stereo matching.” In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2007).*

[13] S. Birchfield and C. Tomasi, “A Pixel Dissimilarity Measure That Is Insensitive to Image Sampling”, In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 401-406, 1998.

APPENDIX - WORK BREAKDOWN

Data Acquisition – Matt

Error Metrics – Zuozhen

Baseline NCC Algorithm – Matt

Graph Cut Model – Zuozhen

Max Flow Integration – Matt

Poster Design – 50/50

Final Report – 50/50