Project Report:
Stereo Matching using Graphcut-based Optimization

Seyed Abbas Sadat
Simon Fraser University
sas21@sfu.ca

Note: All the figures should be seen on screen in order to distinguish the gray level changes. Also for pixel coordinates (i, j) means row i and column j.

I. INTRODUCTION

In this project we developed a stereo matching system based on graph-cut optimization framework [1]. In stereo matching problem, 2 images taken from 2 cameras that are on the same horizontal line are to be matched pixel by pixel. Therefore, for each pixel on the left image, we should find its corresponding pixel on the right image. According to the assumptions, we know that if (i, j) is the coordinate of the pixel in the left image, the coordinate of the corresponding pixel in the right image will be $(i, j - d)$. The unknown variable d is called disparity and is defined in the image space. Disparity is inversely proportional to the depth of the pixel, i.e., the higher the disparity the closer the object. The disparity is not always observable using only 2 images because part of the scene can be occluded in either of the left or right images and hence no matched pixel can be found.

The stereo matching problem is formulated as: given a left and right image, find the most probable disparity image. This *maximum a posteriori* problem is often solved by finding a disparity image that minimizes some energy function. The energy function includes some terms that promote data fidelity (data terms) as well as other terms that support prior assumptions about the solution, e.g., smoothness (smoothness term). These energy functions are non-convex and finding a global optimum it very difficult. Instead, approaches that result in approximation solutions are preferred. One way to reach a local optimum in the minimization is to perform gradient descent. Graph-cuts offer a framework that enables us to generate a movement (step) of current solution that is steepest in terms of minimizing the energy function. By generating and performing these steps iteratively, we reach a local optimum.

In section II we briefly explain how to generate gradient descent moves for a general pixel labelling problem.

II. GRAPHCUT-BASED OPTIMIZATION

The energy function that we use for stereo matching has the following form:

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

Here $D_p(f_p)$ is the data term, i.e. cost of assigning label f_p to pixel p. This cost is calculated based on the input data (left and right images in stereo matching). The $V_{p,q}(f_p, f_q)$ term is the pairwise cost of two neighbours p, q having labels f_p, f_q respectively.

Assuming that we are at some arbitrary solution we generate an $\alpha\beta - swap$ move or $\alpha - expansion$ move. Next, if performing the new move reduces the energy function, we apply the move on the current solution and generate a new move again iteratively.

For computing the best move at each solution, i.e. a move with the largest reduction of energy function, we construct a graph network based on the current solution and the input images. The weights of the edges are carefully defined so that the cost of any cut in the created graph equals the cost of the specific move in the labelling problem (plus a constant). Therefore if we find the minimum cut of the graph, we can convert it to a move that reduces the energy function the most. The following sections describe $\alpha\beta - swap$ and $\alpha - expansion$ moves.

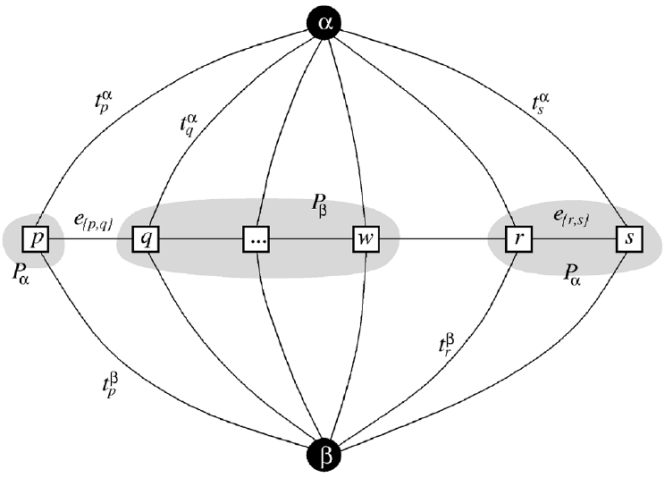


Fig. 1: The graph corresponding to a $\alpha\beta - swap$.

A. $\alpha\beta - swap$

In $\alpha\beta - swap$ move, subset of the pixel having label α can be changed to label β and vice versa. Therefore only the union of the pixel with α or β will change labels and

other pixel remain the same. Therefore in each cycle of the algorithm this move is computed for each possible label pair and then is applied on the current solution if it reduces the energy function. In order to find the best possible label change in an $\alpha\beta - swap$, the minimum graph-cut problem shown in figure 1 is solved:

The terminals are the labels α and β and the middle nodes are all the pixels with either α or β label. The weight of the terminal edges are the data terms $D_p(\alpha)$ or $D_p(\beta)$, i.e. the cost of assigning the terminal label to the pixel. The weight of the edges between middle nodes are the pairwise smoothness term $V_{p,q}(\alpha, \beta)$. Please see [1] for details.

B. $\alpha - expansion$

In $\alpha - expansion$ move, a subset of the pixel that do not have label α can be changed to label α and labels of the rest of the pixels remain unchanged. Therefore in each cycle of the algorithm this move is computed for each possible label and then is applied on the current solution if it reduces the energy function. In order to find the best possible label change in an $\alpha - expansion$, the minimum graph-cut problem shown in figure 2 is solved:

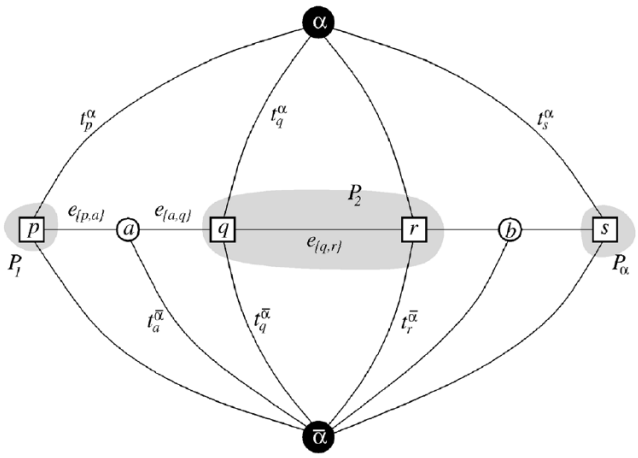


Fig. 2: The graph corresponding to a $\alpha - expansion$.

The terminals are the labels α and $\overline{\alpha}$. Selecting $\overline{\alpha}$ simply means that the pixel will keep its current label. The middle nodes are all the pixels with labels other than α. The weight of the terminal edges are the data terms $D_p(\alpha)$ or $D_p(f_p)$, i.e. the cost of assigning the terminal label to the pixel. Notice that $D_p(f_p)$ is used when $\overline{\alpha}$ is selected for pixel p and f_p is the current label of pixel p. The weight of the edges between middle nodes that with the same labels are the pairwise smoothness term $V_{p,q}(\alpha, f_p)$ in which f_p is the current label of the pair pixels. If the labels of the two neighbour pixels are different we add another axillary node with edges to both corresponding neighbour nodes with weights $V(\alpha, f_p)$ and $V(f_p, \alpha)$ and an edge to the terminal $\overline{\alpha}$ with weight $V(f_q, f_p)$. Please see [1] for details.

III. Stereo Matching

As mentioned earlier in stereo matching we want to find the most probable disparity image given the left and right input images. Therefore we minimize an energy function with data terms and smoothness terms involving the disparity labels. More specifically for data term we use the following:

$$D_{i,j}(f) = min\{|I(i,j) - I'(i,j-f)|, 200\}$$

where I and I' are the left and right image respectively and f is the proposed disparity for pixel (i, j). The constant (200) is to make sure that we are robust to noisy data and outliers. In addition to single pixel cost, we also use window based cost which is the mean of the above data term in a window of size w centred around the corresponding pixel. We will discuss the results in section IV.

For the smoothness term we use 2 different models. The first one is the Potts model as follow:

$$V_{p,q}(f_p, f_q) = 50 * T(f_p \neq f_q)$$

This means that if two neighbouring pixel are assigned different disparities, 50 units will be added to the energy function. This model tend to generate large segments with same disparity and is not appropriate for scenes that there is a gradient in depth. We show this in the experiments. For a better smoothness function which preserves linear depth changes we use the following smoothness term:

$$V_{p,q}(f_p, f_q) = min\{5 * |f_p - f_q|, 50\}$$

This smoothness term penalizes linearly as a function of the disparity difference. We need to cut off these penalties at some point to preserve the edges. We will discuss the results in the next section.

IV. Experiments and Results

For the experiments we used the publicly available Middlebury stereo dataset[1]. Here, we only present the results for 2 different images "Art" in Fig. 5 and "Wood1" in Fig. 6. These figures show the result for pixel level accuracy and subpixel accuracy. By subpixel accuracy we mean that the disparity could have 0.5 disparities. Therfore we can see that the gray level can be different very slightly from the results with pixel level accuracy because of finer resolution.

Next we examine the effect of smoothness model in the energy function. The results are shown in Figure 3. It can be seen that the Potts model does not allow smooth changes in disparity and large areas with single disparity is preferred. On the other hand, with linear model, a number of disparity changes can be recognized which results in a little smoother disparity image.

We also tested the effect of the size of the window that is used to aggregate the data term. Figure 4 demonstrate the results for 3 different window sizes. We can see that as the window size grows, the changes in disparity becomes smoother. This window act like a low pass filter that removes the high frequencies in the disparity image.

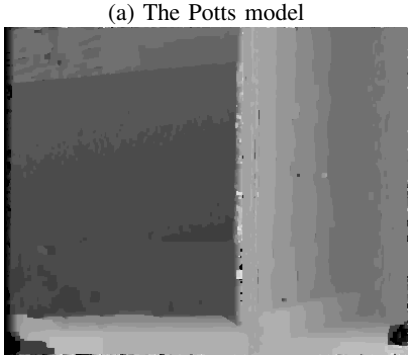
For implementation of this project we used the following libraries:

- Opencv[2]

[1] http://vision.middlebury.edu/stereo/data/

[2] http://opencv.org

(a) The Potts model



(b) Linear model

Fig. 3: Disparity image for the "Wood1" dataset using α-expansion with Potts and linear smoothness models.

- Min-cut library[3]

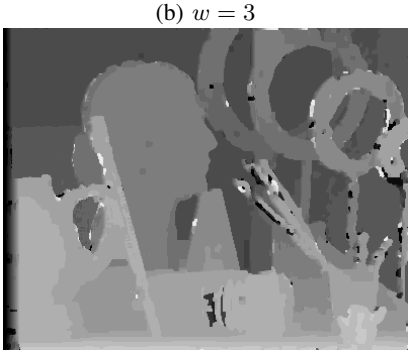
The source code of this project is also publicly available at: $https://github.com/asadat/graphcut\_stereo$.

REFERENCES

[1] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 23, no. 11, pp. 1222–1239, 2001.

(a) $w = 1$

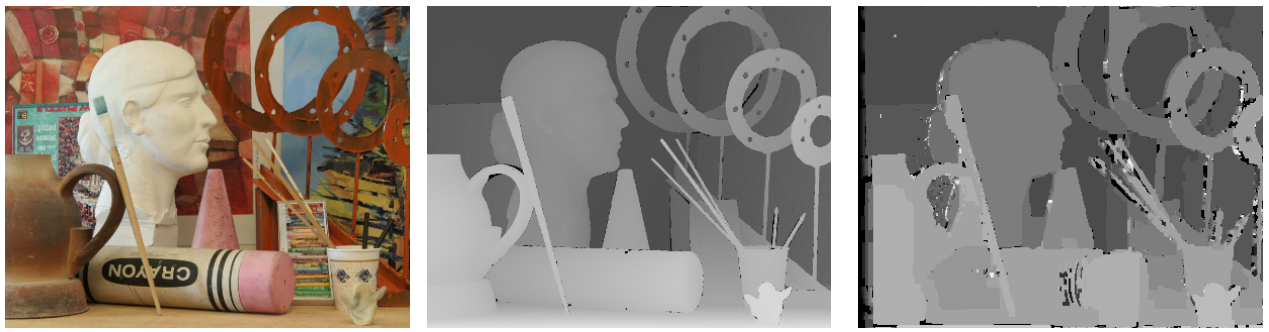
(b) $w = 3$



(c) $w = 5$

Fig. 4: Disparity image for the "Art" dataset using α-expansion with 3 different window sizes for data term aggregation.

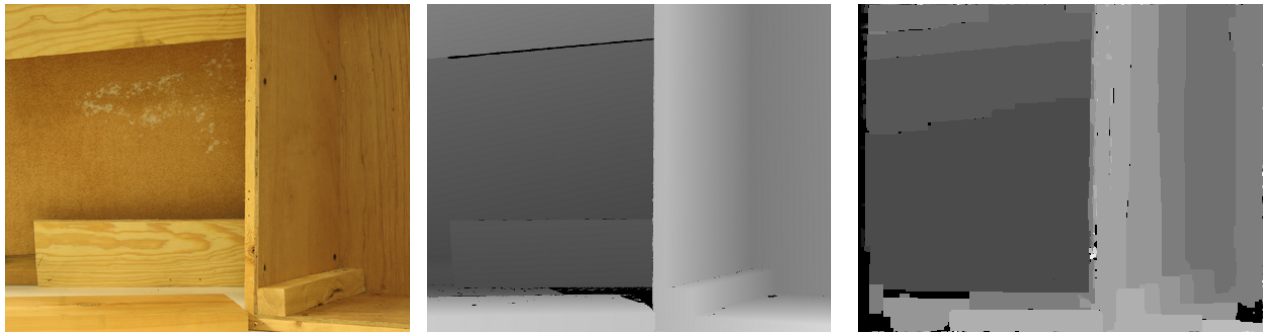
[3]http://vision.csd.uwo.ca/code/



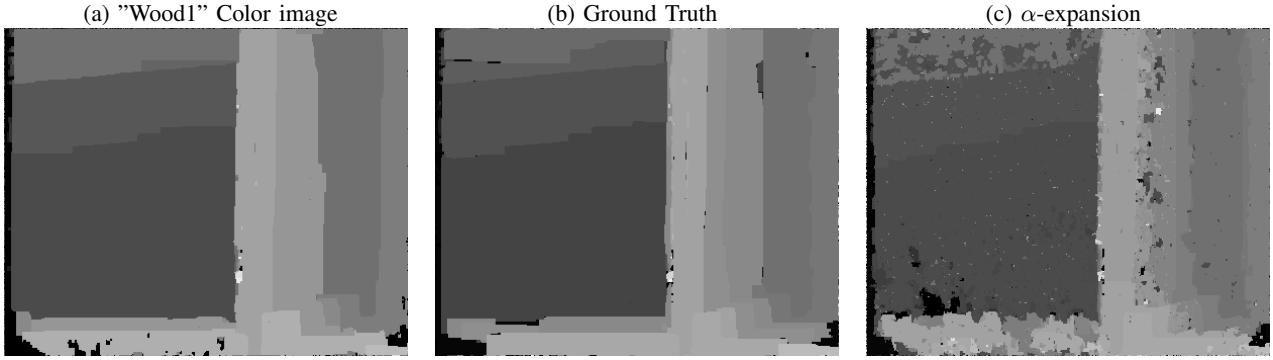
(a) "Art" Color image (b) Ground Truth (c) α-expansion

(d) $\alpha\beta$-swap (e) α-expansion - subpixel accuracy (f) $\alpha\beta$-swap - subpixel accuracy

Fig. 5: Disparity image for the "Art" dataset using the both $\alpha\beta$-swap and α-expansion with pixel and subpixel accuracies.



(a) "Wood1" Color image (b) Ground Truth (c) α-expansion



(d) $\alpha\beta$-swap (e) α-expansion - subpixel accuracy (f) $\alpha\beta$-swap - subpixel accuracy

Fig. 6: Disparity image for the "Wood1" dataset using the both $\alpha\beta$-swap and α-expansion with pixel and subpixel accuracies.