

# Intro to AI: Stereo Disparity Estimation

PB20061372 Yunqin Zhu Jan. 15, 2023

## 1 Introduction

The task of this project is to write a MATLAB program that computes pixel disparities between a stereo pair of images, where the ./input is a rectified pair of images, and the output is a matrix with its values indicating the disparity of the stereo correspondence for each pixel in the left image.

There are two algorithms I implemented: (1) Dynamic Programming (DP), which optimizes the energy separately for each horizontal scanline; (2) Graph Cut (GC), which optimizes the energy on the full 2D grid. Both of them take into account the global smoothness of the disparity map and outperform the naive window-based matching method.

The code framework I followed in the project is from this writeup at Princeton University:  
[www.cs.princeton.edu/courses/archive/fall113/cos429/assignment3/assignment3](http://www.cs.princeton.edu/courses/archive/fall113/cos429/assignment3/assignment3)

## 2 Theoretical Background

Basically, the stereo matching problem can be formulated as to find the disparity map  $\mathbf{d}$  that minimizes the following energy function:

$$E(\mathbf{d}) = E_d(\mathbf{d}) + E_s(\mathbf{d}) = \sum_p D(p, d_p) + \sum_{p,q \in \mathcal{N}} R(p, d_p, q, d_q)$$

where the so-called data term  $D(p, d_p)$  is the pixel-wise dissimilarity cost at pixel  $p$  with disparity  $d_p$ , and the smoothness term  $R(p, d_p, q, d_q)$  is the regularization cost between pixels  $p$  and  $q$  with disparities  $d_p$  and  $d_q$  respectively, for all pairs of neighbouring pixels  $\mathcal{N}$ .

In this project, I formulate the data term as the Sum of Absolute Differences (SAD) with a clamp to reduce the effect of outliers. That is,

$$D(p, d_p) = \min \left\{ \frac{1}{|\mathcal{W}(p)|} \sum_{q \in \mathcal{W}(p)} |\mathbf{I}_q - \mathbf{I}_{q+d_p}|, \tau \right\}$$

where  $\mathcal{W}(p)$  is the window centered at pixel  $p$ ,  $\mathbf{I}_q$  is the intensity of pixel  $q$  in the left image,  $\mathbf{I}_{q+d_p}$  is the intensity of the corresponding pixel in the right image. Similarly, I formulate the smoothness term as the absolute difference between the disparities of two neighbouring pixels and clamp its range. That is,

$$R(p, d_p, q, d_q) = \min \{ |d_p - d_q|, \tau' \}$$

The energy function, on a per-scanline basis, can be minimized in polynomial time using Dynamic Programming, which leads to our first implementation. To be specific, for each row  $y$  of the disparity map, we compute the following aggregated cost function:

$$C_y^x(d) = D((1, y), d) + \min_{d'} \{ C_y^{x-1}(d') + R((x-1, y), d', (x, y), d) \}, \quad x = 2, 3, \dots, W$$

where  $W$  is the width of the image. By choosing the minimum  $C_y^W(d)$  and backtracking, we can then find the optimal disparity  $d_{(x,y)}^*$  for each pixel  $(x, y)$  in the row. In such cases, since no regularization is performed across scanlines, the resulting disparity map can demonstrate streaking artifacts.

Actually, for the smoothness term we use, it is possible to find a solution on the full 2D grid in polynomial time using Graph Cut optimization, which leads to our second implementation. For simplicity, our code will be based on GCMex, the MATLAB wrapper of Veksler, Boykov, Zabih and Kolmogorov's Graph Cut algorithm.

## 3 Implementation Details

See the ./code folder in the attached zip file. The maximum disparity is set to 60. The window size is set to 7x7. The clamp thresholds  $\tau$  and  $\tau'$  for the data and smoothness terms is set to 10 and 1.7 respectively. Furthermore, to trade off between the data and smoothness terms, I multiply the data term by a factor  $\lambda = 0.04$ . After running the code, the disparity maps will be saved in the ./disparity folder. The code also computes the Root Mean Square Error (RMSE) on each test image to evaluate the performance of the algorithms.

## 4 Results and Analysis

Table 1: Test images and output disparities of two stereo matching algorithms

Dataset	Input left	Input right	Ground truth	Disparity (DP)	Disparity (GC)
2001 Venus					
2001 Sawtooth					
2001 Tsukuba					
2003 Teddy					
2006 Baby2					
2006 Cloth1					
2003 Cones					

As is shown in Table 1, both algorithms can produce reasonable disparity maps. The DP algorithm is more sensitive to noise and produces more streaking artifacts. The GC algorithm, on the other hand, is more robust to noise and produces smoother disparity maps.

Table 2: RMSE on the test dataset

	2001 Venus	2001 Sawtooth	2001 Tsukuba	2003 Teddy	2006 Baby2	2006 Cloth1	2003 Cones
DP	1.0864	1.7392	3.1689	6.9476	4.3338	0.58045	5.7699
GC	0.88611	1.6715	3.2	6.4777	4.782	0.88025	6.1692

However, as is shown in Table 2, the GC algorithm does not necessarily produce results with lower RMSE, and the performance varies from one image to another. Part of the reason is that we use the same window radius for all test cases, which is not optimal for either GC or DP algorithms. Besides, in our MATLAB implementations, the computational cost of the GC algorithm is much higher (about 3 times) than that of the DP algorithm. Instead of asserting that one algorithm is better than the other, we should carefully choose the algorithm that best suits the application.