

# CS 6320 Computer Vision

## Homework 3 (Due Date – March 11<sup>th</sup>)

1. Implement a belief propagation program that uses a factor graph and cost tables as input, and produces the states/labels for all the variables. The BP program can be summarized in 5 steps as shown in the lecture (Slide 9 of [http://www.eng.utah.edu/~cs6320/cv\\_files/BeliefPropagation.pdf](http://www.eng.utah.edu/~cs6320/cv_files/BeliefPropagation.pdf)). First, we initialize all messages from variable to factor nodes with zero's or random values. The algorithm iterates through steps 2 to 5 based on two terminating conditions. The first terminating condition checks if the labels of all the variables change with subsequent iterations or not. If there is no change, the algorithm is terminated. The second terminating condition checks if the number of iterations has exceeded a threshold or not. Tips: During the implementation of BP, please assume that all variables take the same number of labels. Test the code on the factor graph shown in Fig 1.

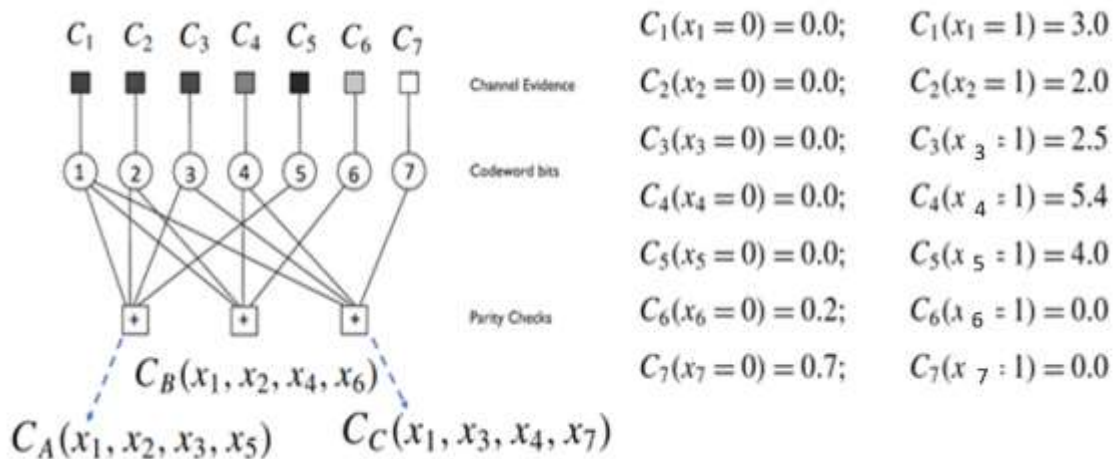


Figure 1: The above factor graph shows a toy problem with seven variable nodes (circles) and ten factor nodes. [Image courtesy: Jonathan Yedidia]

We have 7 Boolean variables  $\{x_1, x_2, \dots, x_7\}$ . The goal is to find the final states/labels for the Boolean variables based on the 10 cost tables  $\{C_1, C_2, \dots, C_7, C_A, C_B, C_C\}$ . As shown in the figure, the first 7 cost tables for the first 7 factor nodes are given by  $2 \times 1$  matrix as shown above. For the last 3 parity constraints, we have multi-dimensional tables of size  $2 \times 2 \times 2 \times 2$ . For example, the cost table for  $C_A(x_1, x_2, x_3, x_5)$  is given below:

$$C_A(x_1, x_2, x_3, x_5) = 0 \text{ if } (x_1 + x_2 + x_3 + x_5) == \text{even},$$

$$C_A(x_1, x_2, x_3, x_5) = 1000 \text{ (or infinity) if } (x_1 + x_2 + x_3 + x_5) == \text{odd}$$

All the parity cost functions associated with  $C_A, C_B$  and  $C_C$  should be the same, while they still take different input variables. We have a cost for every configuration of 4 Boolean variables, and thus the multi-dimensional cost table will have  $2 \times 2 \times 2 \times 2$  values. Set the maximum number of iterations to be 10.

[50 Points]

- Write a stereo matching program to compute the disparity map using the BP algorithm. The basic idea of stereo matching is given here. You are given two images that we normally refer to as “left” and “right” images. We can think of each image to have several horizontal scanlines. The images are said to be rectified if every pixel in the left image has a matching pixel in the right image at the same scanline. In a typical stereo pair, a pixel  $p(x,y)$  in the left image usually has a matching pixel  $q(x',y)$  in the right image. Note that the ‘y’ coordinate is the same on both the left and right images. Every pixel in the left image gets shifted by a small amount ‘d’ on the right image in the following manner:  $x' = x - d$ . Here ‘d’ is supposed to be the disparity for the pixel  $p(x,y)$  in the left image. Note that a pixel that is further away from the camera will have a very small disparity value. A pixel that is very close to the camera will have a high disparity value. In other words, the depth “D” of a pixel is inversely proportional to the disparity ‘d’. We consider the image as a 4-connected image graph as shown below:

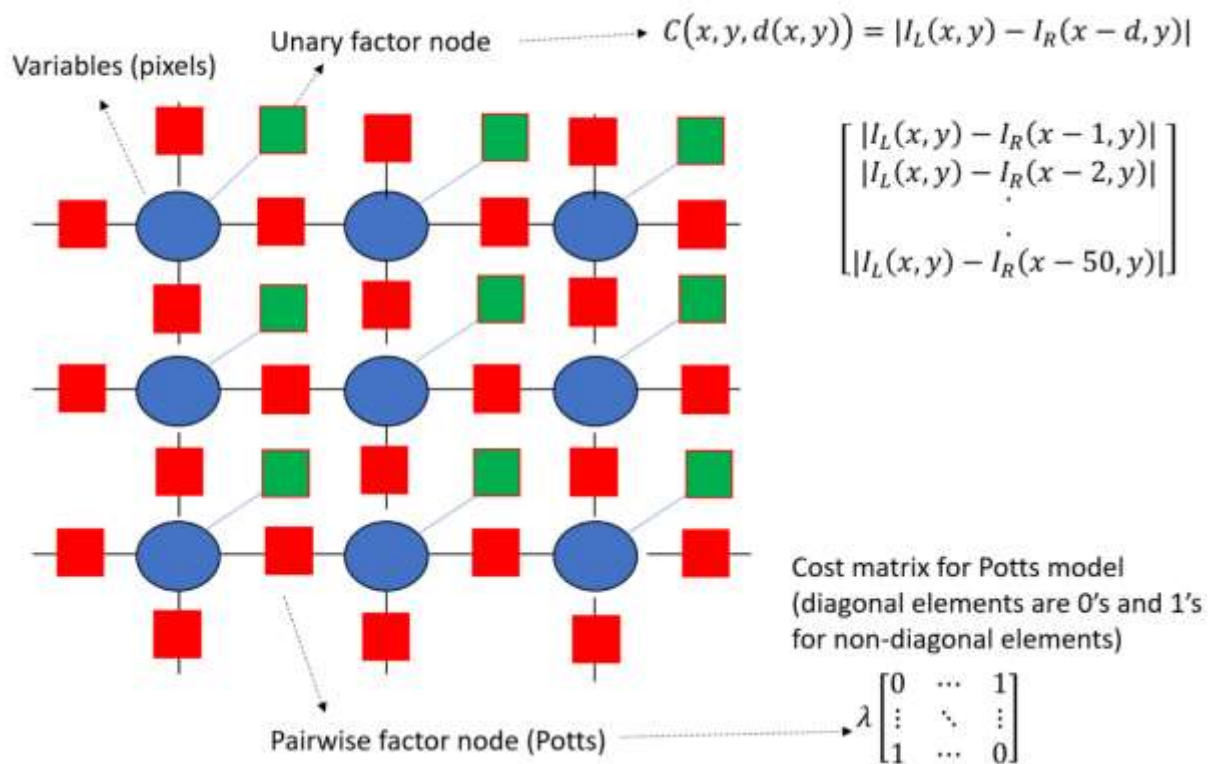


Figure 2: We show the factor graph for a stereo matching problem. The variables are shown in blue circles, and the factor nodes are shown in squares. The green squares denote the unary potentials or factor nodes that depend on only one variable. The red squares denote the pairwise potentials that depend on two variables. Consider an image of dimensions  $M \times N$ , where  $M$  is the height and  $N$  is the width of the image. We have  $MN$  unary factor nodes and  $\{(N-1)M + (M-1)N\}$  pairwise factor nodes.

We will use BP to minimize the following energy function to find the optimum disparity labels/states:

$$E(d) = E_{data}(d) + \lambda E_{smooth}(d)$$

The first term is the data cost, and it is given by:

$$E_{data}(d) = \sum_{x,y} C(x, y, d(x, y))$$

The cost  $C(x,y,d)$  in matching a  $p(x,y)$  with another pixel  $q(x-d,y)$  can be a simple pixel-wise intensity difference as shown below:

$$C(x, y, d(x, y)) = |I_L(x, y) - I_R(x - d, y)|$$

Let us consider the left image to be the reference image. Each pixel in the left image can be seen as a variable that can take different disparity values (say 1-50). As shown in Figure 2, the data term consists of  $MN$  ( $M$  is the height of the image and  $N$  is the width of the image) unary factor nodes where each factor node is attached to every pixel in the image. Each of the cost tables for unary factor node will only consist of 50 values (assuming that each pixel can take only 50 disparity states).

We will use Potts model for the smoothness term as shown below:

$$E_{smooth}(d) = \sum_{(i,j) \in E} f(d(x_i, y_i), d(x_j, y_j))$$

$$diff = |d(x_i, y_i) - d(x_j, y_j)|$$

$$\text{Potts: } f = 1 \quad \text{if } diff > 0 \quad \text{else } f = 0$$

In Figure 2, we show red squares that denote the pairwise or smoothness factor nodes. We consider a 4-connected graph. We will have  $\{(M-1)N + (N-1)M\}$  factor nodes for pairwise terms. Each of the cost tables for pairwise factor node will consist of  $[50 \times 50]$  elements (assuming that you have 50 labels for disparity values). Note that these tables have the same entries for every pairwise factor node. In fact, for the Potts model, these cost tables are simple matrices that have 0's for diagonal entries and 1's for non-diagonal entries. These simple matrices are multiplied by  $\lambda$ . Please do not end up storing different tables for every pairwise factor node as this may lead to unnecessary memory issues.

Use different values for  $\lambda = \{1, 10, 50, 100\}$ . The termination condition should be either 20 iterations or when the labels stop changing in subsequent iterations. Show the solutions and the value of the final cost function. Show the final result in a disparity image. There are three image pairs provided with this assignment. If you show the results on the first two image pairs (out of the 3 pairs of images

given with the assignment), it is sufficient. If you encounter overflow problems try to adjusting the contents of the messages by subtracting the minimum element. [Hint: Please pay attention to memory management and please don't try to code a general BP program, as it may be challenging.]

[50 Points]