Assignment 3: Markov Random Field Models

Team- Vidhixa Joshi and Divya Lalwani

Code Organization

Our submission folder consists of following files:

1. segment.cpp
   - naïve_segment
   - mrf_segment

   The output generated are going to be named as naïve_segment_result_bg.png, naïve_segment_result_fg.png, mrf_segement_result_bg.png and mrf_segement_result_fg.png.

2. stereo.cpp
   - unary_stereo function
   - mrf_loopy function

   The output generated are going to be named as naïve_stereo.png and loopy_mrf_stereo.png

Tasks Performed

<u>Setup</u>

Files were successfully untared and run on tank linux machine.

<u>Segmentation</u>

This file consists of functions required for Object Segmentation Process. The functions are: calMeanStd (calculate means and variances/standard deviations w.r.t. foreground stroke), DCost (calculate unary cost for a given pixel and label), VCost (calculate pairwise cost between current pixel and it's neighboring pixels), naïve_segment (assign labels based on naïve_segmentation method), mrf_segment (assign labels based on mrf_segmentation method).

1. Naïve segmentation:
   Following procedure was followed for naïve segmentation:
   - Calculate Means and variances for a given image using its foreground stroke from seeds. It calculates required things and stores them in global variables. This function returns an array which has foreground and background pixels stored in it. If a pixel doesn't belong to foreground or background stroke in seeds image, we assign it a value of 100, in order to differentiate it.
   - Next, we calculate unary cost D(L(p), I(p)) for ever pixel using label 0 and 1. The dist function works as: If a pixel belongs to foreground in seeds, assign it minimum distance for label 1. If it belongs to background in seeds, assign it minimum distance for label 0. And if pixel doesn't belong to either of them, we use Gaussian probability density function and a constant Beta to assign label which gives minimum cost. For Gaussian probability density function, we have used the equation of negative log probability posted on forum. And we

haven't used any Z, because we have adjusted that factor in our Beta. On trial and error, Beta = 3930 worked best for cardinal image.

Here are the results for cardinal:

2. MRF Segmentation:
   - We have used the algorithm described in http://nghiaho.com/?page_id=1366 for MRF inference using loopy belief propagation.
   - For energy minimization, we are using the equation where in energy is summation unary and pairwise cost.
   - For loopy belief propagation, we pass messages between a pixel and its neighbors to establish a belief about a neighbor being assigned a label l.
   - We use a three dimensional array data structure to store messages. First 2 dimensions are pixel row and column, third dimension is for label (which could be 0 or 1). The fourth dimension is a hidden dimension for pixel's data, its right, left, top and bottom neighbor.

Steps followed are:

1.) Initialize all messages to 0.

2.) For each pixel, compute unary cost and pairwise cost between its neighbor using all possible combinations of labels and sum these costs. Now, add the messages from neighbors and data of the observed variable to these costs in order to compute label for hidden pixel. Pass these messages to neighbors in a fashion that if we pass a message to right neighbor of a pixel, we consider messages from observed variable, and left, top, bottom variables. Similarly, we pass messages to left, top and bottom nodes.

3.) We then compute minimum cost for assigning a label. And store this cost as our new message for all neighbors respectively.

4.) Then, for a given pixel, we sum up all the messages and compute the minimum one and accordingly assign a label to the pixel i.e. the belief.

5.) We repeat this procedure for few iterations until we get changes in the belief. For cardinal image, we didn't get any significant change after the 5^{th} iteration. So, we ran the entire procedure for 5 iterations.

But, we couldn't get expected results for MRF Segmentation. Also, somehow foreground strokes appear in the background image.

Stereo

1. In this step we were required to perform stereo over a left image and right image of same scene. From the data provided, we assume that there is horizontal translation in the camera. All we need to do is to compute the horizontal disparities by comparing the pixels of left and l=right image based on following formula:

$$D_2(d, I_L, I_R, (i,j)) = \sum_{u=-W}^{u=W} \sum_{v=-W}^{v=W} (I_L(i+u, j+v) - I_R(i+u, j+v+d))^2$$

- We have considered a window of size 8 for the technique window matching.
- The number of possible labels are assumed to be 30 i.e 'd' or unary cost.
  (The values work good with Aloe images, which is the test set for stereo part)

2. With the unary cost (D2) computed above, we start with MRF inference using loopy belief. The following steps were carried out for it:

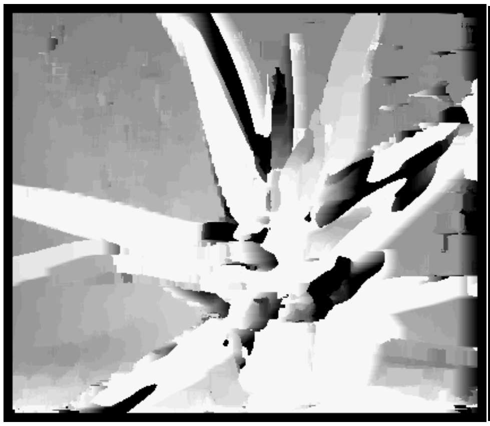
$$F(L, I) = \sum_{p \in I} D_2(L(p), I_L, I_R, p) + \alpha \sum_{p \in I} \sum_{q \in \mathcal{N}(p)} V_2(L(p), L(q))$$

- We used the quadratic function for calculating smoothness cost. We subtract the label costs of all neighbors ([i,j+1], [i,j-1], [i+1,j], [i-1,j]) with its own label [i,j] and square the difference. We then add all the square of difference and call it smoothening cost for that pixel. Although it is more resource consuming but, gave better results than other models. We call this vcost in out program (V2).

$$(L(p) - L(q))^2$$

- We pass messages as in a loop fashion for 10 number of iterations. The results obtained pretty much the same for all iterations thus tentatively, we give it a number 10.
- For each pixel, we compute a message in all directions for all the labels and select the best one.
- Here, we consider 30 labels and form messages according to its direction, vcost(V2) and pixels unary cost(D2)
- Message size kept increasing in every direction so we followed professor's suggestion made on oncourse forum. We took average of three messages (excluding the message from direction where we are intending on sending it). Then we subtract each of those messages from the average computed and add them up to the message packet.
- For all the label, we compare such messages and the one which shows the minimum cost is selected as best. Based on this best cost, associated label is selected.
- Each iteration keeps updating the disparity over previous based on new messages and final output that we obtained for a smoother stereo than naïve method.

- Results obtained from naïve and loopy belief respectively. Worked well with only this example, bowling. Remaining images, it showed flaws.
- Error computed was between 18 and 40 depending upon window size and number of labels.S



References

1. http://www.ski.org/Rehab/Coughlan_lab/General/TutorialsandReference/BPtutorial.pdf
2. http://nghiaho.com/?page_id=1366
3. Course Forum