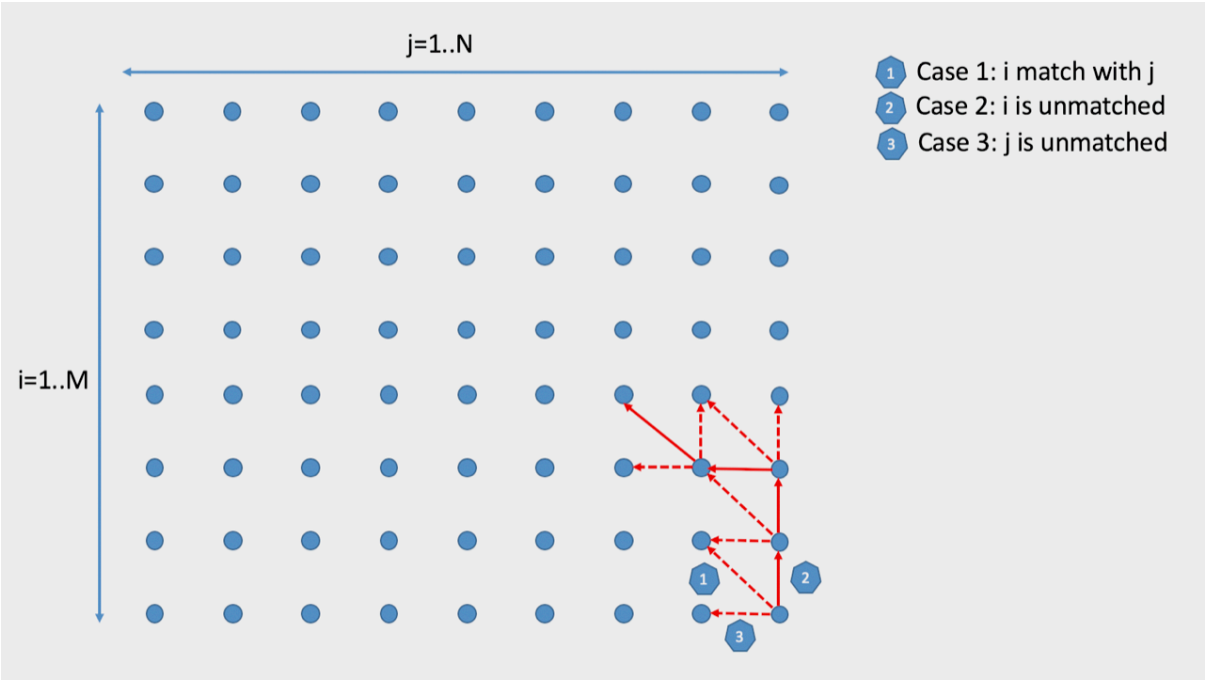
COMP0005_Algorithms Coursework 2

1. When our eyes view a scene, the left and right eyes see slightly different things. The left and right images are known as a stereo pair. By matching corresponding points in the two images, we are able to infer depth. The attached paper http://www.bmva.org/bmvc/1992/bmvc-92-035.pdf proposes using a dynamic programming algorithm to perform stereo matching. Implement the algorithm. Please note first the following points:
   - We will only focus our study on grayscale images, i.e. only one scalar value by pixel (pixel intensity). Therefore, vectors and matrices collapse to scalar values in that specific case (specifically, the covariance matrix becomes the variance).
   - The algorithm given on page 341 will allow you to do the forward pass and thus build the cost matrix. The parameters to tune the model are given in the « Experimental Results » section. Be careful, you will need to store which of the three costs (see paper) have been chosen for each i and j to reconstruct the optimal match:

   1] C(i-1,j-1)+c(z1i,z2j), i.e. pixels i and j do match;
   2] C(i-1,j)+occlusion, i.e. pixel i is unmatched;
   3] C(i,j-1)+occlusion, i.e. pixel j is unmatched.

You will need to implement the backward pass to infer depth. This part is not documented in the paper but follow the following procedure:



Starting from i=M and j=N (and until i=0 and j=0), you will move along one of the three arrows starting from the current point:
- going up: if pixel i is unmatched;
- going left: if pixel j is unmatched;
- going upper left: if pixels i and j match.

The distance between pixel i and j (if they match) is linked to the depth and is called disparity. As an output of the algorithm, you should display the disparity map (showing the disparity for each pixel).

To study the algorithm, create a synthetic random dot stereogram:
(i) create a 512x512 image **A** of random black and white pixels (0 and 255 as pixel values),
(ii) create a second 256x256 image **B** of random black and white pixels (0 and 255 as pixel values),
(iii) now create a left image **L** by placing the 256x256 image **B** into the 512x512 image **A** such that the top-left corner of the 256x256 image **B** starts at (124,128),
(iv) now create the right image **R** by placing the top-left corner of the 256x256 image **B** at (132,128).
You should apply the stereo matching algorithm on the pair of images (**L**,**R)**.
Provide results for the additional images provided here as well.

Please discuss the results and answer the following questions:
- Why do matching errors occur for the binary random dot stereograms?
- Investigate how the algorithm performs on other images as the occlusion cost is varied.
- For string matching, what is the equivalent of occlusion?