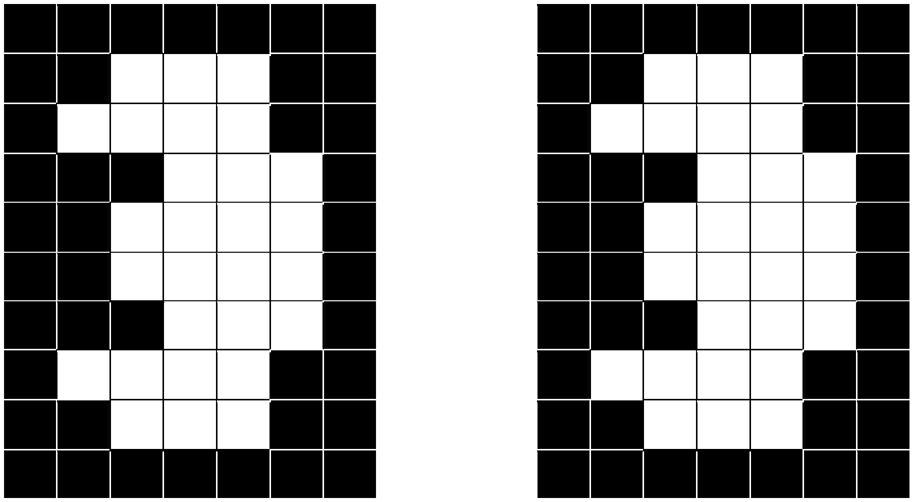
Homework 4: Due 10th November at 9:50 am (before class)

Same weight as Homework 1

Section 1: 50 points

1. For the given shape, mark the P_8 perimeter in the figure on the left, and P_4 perimeter in the figure on the right.



   Compute the C_1 circularity of this shape. Also show the sequence of the order in which pixels will be visited by a recursive connected components labeling algorithm. Assume that the algorithm scans the image from left to right, top to bottom, and that the eight neighbors are checked in clockwise direction from North; i.e. in the order N, NE, E, SE, S, SW, W, NW.

2. Derive the expression for a 2D Laplacian of Gaussian (LoG) filter (using definition of Laplacian operator) and show that the following 3x3 mask approximates it for small σ (Hint: try σ close to 0.5):

0	-1	0
-1	4	-1
0	-1	0

3. Rotation Invariance of Gradient Magnitude: Show that the gradient magnitude is rotation invariant. Let the gradient at point (*x, y*) be given by (f_x , f_y). Assume that the object is rotated around *Z*-axis, and the point (*x, y*) moves to (*x*', *y*'). Show that $\sqrt{f_x^2 + f_y^2} = \sqrt{f_{x'}^2 + f_{y'}^2}$

4. Prove that the least-squares solution for global flow (affine) is given by the solution of the following equation:

$$\left[\sum_{\forall(x,y)\in I} \mathbf{X}^T (\mathbf{f_x})(\mathbf{f_x})^T \mathbf{X}\right]\mathbf{a} = -\sum_{\forall(x,y)\in I} f_t \mathbf{X}^T \mathbf{f_x}$$

5. Show that normal flow, *d*, is given by:

$$d = \frac{f_t}{\sqrt{f_x^2 + f_y^2}}$$

   The following matrices, **a** and **b**, show a bright triangle translating by (*u*=0, *v*=1). For these matrices, compute *d* at locations (6,5) and (3,8), using the masks given in the handout (assume origin of mask at bottom right corner). How would you interpret these values?

a =

3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	7	3	3	3	3	3	3	3	3
3	9	7	5	3	3	3	3	3	3
3	9	9	7	5	3	3	3	3	3
3	9	9	9	7	5	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3

b =

3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	7	3	3	3	3	3	3	3
3	3	9	7	5	3	3	3	3	3
3	3	9	9	7	5	3	3	3	3
3	3	9	9	9	7	5	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3
3	3	3	3	3	3	3	3	3	3

Given the matrices, **a** and **b** above, compute the brightness constancy equation at location (6,5), and show that with (u = 0, v = 1), it is equal to zero. What other possible values of (u,v) are also valid solutions of BCE? Draw a few of them on the image and comment.

Section 2: 50 points

6. Implement Lucas-Kanade Algorithm (without pyramids) in MATLAB or in C. Apply the function that you have written on yosemite1r.pgm and yosemite2r.pgm files given with the homework document.

   Tips for MATLAB implementation:

   Implementation in MATLAB is really quite simple because of the ease of using matrices. The following tips will help you in implementation:

   a. Use the `help` command to look up help of any function e.g. type `>> help` for to see how loops work in MATLAB.
   b. Your function should take two binary images as input. The code of reading binary images (readbinpgm) is available in the homework folder. The output should be two matrices u and v containing the (u,v) values at each pixel.
   c. Use the same derivative masks as given in the handout. You can use the `conv2` function for a quick implementation in a single line.
   d. The `inv` command is used for matrix inversion. Transpose of a matrix `A` is given by `A'`. This makes taking pseudo-inverse really easy.

   Viewing the results

   Once you have gotten the output matrices u and v, use the following command in MATLAB to view the results: `>> quiver(flipud(u), flipud(-v), 0);`

   You can also view the u and v matrices as images in C or in MATLAB (e.g. `>> imagesc(u)`) and see if they make sense). In fact, viewing the magnitude and direction of motion vectors may be more useful. Comment on the results, and see if the results make sense at all locations in the images.

   Submit a printout of your results and code.

7. This problem may be attempted in C or in MATLAB.

   The LUMS sequence consists of 256 frames, out of which roughly the first 100 frames contain only the background. The rest of the frames have a sequence of two persons walking in it. Use the initial frames to build a color background model based on the assumption that the color at each pixel is can be modeled as an Gaussian distribution independent of other pixels. Use this model to perform background subtraction to detect the persons as foreground regions in the rest of the sequence. Select an appropriate threshold such that the detected persons appear as complete as possible, and output your results as binary images. Comment on your results, and *suggest ways of improving the output*. Submit the output sequence in the submission folder, and show selected frames in your homework solution.