Assignment 2: CS 763, Computer Vision

Due: 17th February before 11:55 pm

Remember the honor code while submitting this (and every other) assignment. All members of the group should work on all parts of the assignment. We will adopt a zero-tolerance policy against any violation.

Submission instructions: You should ideally type out all the answers in Word (with the equation editor) or using Latex. In either case, prepare a pdf file. Put the pdf file and the code for the programming parts all in one zip file. The pdf should contain the names and ID numbers of all students in the group within the header. The pdf file should also contain instructions for running your code. Name the zip file as follows: A2-IdNumberOfFirstStudent-IdNumberOfSecondStudent-IdNumberOfThirdStudent.zip. (If you are doing the assignment alone, the name of the zip file is A1-IdNumber.zip). Upload the file on moodle BEFORE 11:55 pm on 17th February. Late assignments will be assessed a penalty of 25% per day late. Note that only one student per group should upload their work on moodle. Please preserve a copy of all your work until the end of the semester.

1. Prove that collinearity of points is preserved under perspective projection. [4 points]

   Ans. Let P_1, P_2 be two points on a line and p_1, p_2 be their respective images through perspective projection. we need to prove that linearity is preserved.It is sufficient to show that slope of line formed by p_1, p_2 is independent of any coordinate of P_1, P_2 which will imply that slope will remain same for another pair p_2, p_3 formed by P_2, P_3 where P_3 is also a point on given $3D$ line.
   Let line formed by P_1, P_2 be given by

   $$\frac{X}{k} + l = \frac{Y}{m} + n = Z.$$

   where k, l, m, n are constants for a given line.
   Slope of line formed by p_1, p_2 is $\frac{x_2 - x_1}{y_2 - y_1}$.

   $$\frac{x_2 - x_1}{y_2 - y_1} = \frac{\frac{fX_2}{Z_2} - \frac{fX_1}{Z_1}}{\frac{fY_2}{Z_2} - \frac{fY_1}{Z_1}}$$

   $$\implies \frac{x_2 - x_1}{y_2 - y_1} = \frac{X_2 Z_1 - X_1 Z_2}{Y_2 Z_1 - Y_1 Z_2}$$

   $$\implies \frac{x_2 - x_1}{y_2 - y_1} = \frac{kl}{mn}$$

   which is independent of coordinates of P_1, P_2 implying that slope of p_2, p_3 will be same and p_1, p_2, p_3 are collinear if P_1, P_2, P_3 are collinear.
   Thus proven that linearity is preserved under perspective projection.

2. *We have defined the concept of the Shannon entropy in class. Given a discrete random variable X having probability mass function $P(X) = (p_1, p_2, ..., p_N)$, prove that $H(X) \geq 0$ where $H(X)$ is the Shannon entropy of X. Recall that $H(X) = -\sum_{i=1}^{N} p_i \log p_i$ and that $\sum_{i=1}^{N} p_i = 1$ and $\forall i, 0 \leq p_i \leq 1$. Also prove that a uniform distribution (ie $\forall i,\ p_i = \frac{1}{N}$) maximizes the Shannon entropy. To this end, find a stationary point of $J(X) =$*

$H(X) - \lambda(\sum_{i=1}^{N} p_i - 1)$ *where* λ *is a Lagrange multiplier to impose the hard constraint that the probabilities all sum up to 1. [2+4 = 6 points]*

Ans. Note that $0 \leq p_i \leq \forall i$ and therefore $logp_i < 0$. Thus

$$H(X) = -\sum_{i=1}^{N} p_i \log p_i = \sum_{i=1}^{N} p_i(-\log p_i) \geq 0$$

To find the point of maximum of $H(X) = -\sum_{i=1}^{N} p_i \log p_i$ given constraint $\sum_{i=1}^{N} p_i = 1$, we need to find the stationary point of Lagrangian $J(X) = H(X) - \lambda(\sum_{i=1}^{N} p_i - 1)$

Thus $\forall i = 1 \dots N$,

$$\frac{\partial J}{\partial p_i} = 0 \tag{1}$$

$$-\log p_i - 1 + \lambda = 0 \tag{2}$$

$$p_i = 2^{\lambda - 1} = \nu \tag{3}$$

for some constant ν. Putting this back in the constraint,

$$\sum_{i=1}^{N} p_i = 1$$

we get $p_i = \nu = 1/N \ \forall i = 1 \dots N$. Thus maximum entropy for a discrete random variable is achieved at unifrom distribution.

3. This is a straightforward exercise to make sure you understand the basic update equations in the Horn-Shunck algorithm for optical flow. As seen in class, we seek to minimize the quantity $J(\{(u_{ij}, v_{ij})\})$ w.r.t. the optical flow vectors $(u_{i,j}, v_{i,j})$ at all pixels (i, j), where $J(\{(u_{i,j}, v_{i,j})\}) = \sum_{i=1}^{N}\sum_{j=1}^{N}(I_{x;i,j}u_{i,j} + I_{y;i,j}v_{i,j} + I_{t;i,j})^2 + \lambda((u_{i,j+1} - u_{i,j})^2 + (u_{i+1,j} - u_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2 + (v_{i+1,j} - v_{i,j})^2)$. Setting the partial derivatives w.r.t. $u_{k,l}$ and $v_{k,l}$ to 0, prove that

$$u_{k,l} = \bar{u}_{k,l} - \frac{I_{x;k,l}(I_{x;k,l}\bar{u}_{k,l} + I_{y;k,l}\bar{v}_{k,l} + I_{t;k,l})}{I_{x;k,l}^2 + I_{y;k,l}^2 + 4\lambda} \tag{4}$$

$$v_{k,l} = \bar{v}_{k,l} - \frac{I_{y;k,l}(I_{x;k,l}\bar{u}_{k,l} + I_{y;k,l}\bar{v}_{k,l} + I_{t;k,l})}{I_{x;k,l}^2 + I_{y;k,l}^2 + 4\lambda} \tag{5}$$

where $\bar{u}_{k,l}$ and $\bar{v}_{k,l}$ are as defined in the lecture slides. [4 points]

Ans. $J(\{(u_{i,j}, v_{i,j})\}) = \sum_{i=1}^{N}\sum_{j=1}^{N}(I_{x;i,j}u_{i,j} + I_{y;i,j}v_{i,j} + I_{t;i,j})^2 + \lambda((u_{i,j+1} - u_{i,j})^2 + (u_{i+1,j} - u_{i,j})^2 + (v_{i,j+1} - v_{i,j})^2 + (v_{i+1,j} - v_{i,j})^2)$

setting partial derivative to 0 w.r.t. $u_{k,l}$ implies

$$\frac{\partial J}{\partial u_{k,l}} = 0$$

Here terms containing $u_{k,l}$ are necessary which are $i = k, j = l$ and $i = k - 1, j = l$.

$$\implies (I_{x;k,l}^2 + 4\lambda)u_{k,l} + I_{x;k,l}I_{y;k,l}v_{k,l} = 4\lambda\bar{u}_{k,l} - I_{x;k,l}I_{t;k,l}$$

where $\bar{u}_{k,l} = (u_{k+1,l}+u_{k,l}+u_{k,l}+u_{k,l-1})/4$.
setting partial derivative to 0 w.r.t. $v_{k,l}$ implies

$$\frac{\partial J}{\partial v_{k,l}} = 0$$

Same as above terms containing $v_{k,l}$ are necessary which are $i = k, j = l$ and $i = k-1, j = l$.

$$\implies (I^2_{y;k,l} + 4\lambda)v_{k,l} + I_{x;k,l}I_{y;k,l}u_{k,l} = 4\lambda\bar{v}_{k,l} - I_{y;k,l}I_{t;k,l}$$

where $\bar{v}_{k,l} = (v_{k+1,l}+v_{k,l}+v_{k,l}+v_{k,l-1})/4$.
solving these two equations for $u_{k,l}$ and $v_{k,l}$ gives

$$u_{k,l} = \bar{u}_{k,l} - \frac{I_{x;k,l}(I_{x;k,l}\bar{u}_{k,l} + I_{y;k,l}\bar{v}_{k,l} + I_{t;k,l})}{I^2_{x;k,l} + I^2_{y;k,l} + 4\lambda}$$

$$v_{k,l} = \bar{v}_{k,l} - \frac{I_{y;k,l}(I_{x;k,l}\bar{u}_{k,l} + I_{y;k,l}\bar{v}_{k,l} + I_{t;k,l})}{I^2_{x;k,l} + I^2_{y;k,l} + 4\lambda}$$

(4)

You know that both the Horn-Shunck as well as Lucas-Kanade methods bank on the brightness constancy assumption. Given a pair of images, let us suppose that this assumption holds good for most physically corresponding pixels, but not for some p% of the pixels. Briefly explain how you will modify the Horn-Shunck method and Lucas-Kanade method to deal with this. [4 points]

In this assignment, you will build up on the previous assignment to estimate the homography between two images. This time, you should use the RanSaC algorithm to estimate the homography in order to make the estimate resistant to the presence of incorrect point correspondences. The code for RanSaC for various problems including estimation of homographies is available at http://www.csse.uwa.edu.au/ pk/research/matlabfns/. You should work with the images in the folder http:
www.cse.iitb.ac.in/ ajitvr/CS763$_S$*pring*2015/*HW*2/*Homographyandalsoonanyonepairofpicturesofanapproximatelypla*
overlappingareastomakethismoreinteresting.)*Also*, *ineachcase*, *youshouldwarponeoftheimagessothatitalignswiththeo*

Here we will register a flash and a no-flash image pair using the joint entropy criterion we studied in class. Download the flash and no-flash images from http:
www.cse.iitb.ac.in/ ajitvr/CS763$_S$*pring*2015/*HW*2/*ImageReg.Convertbothimagestocolorgray*−*scale.Theflashimagean*
flashimagehavedifferentimageintensitiesatmanyplaces, *andtheno*−*flashimageisdistinctlynoisier.Rotatethegivenno*−
flashimagecounter−*clockwiseby*28.5*degrees*, *translateitby*−6*pixelsintheXdirection*, *andaddGaussiannoiseofstandardd*
255*scale*).*Notethattherotationmustbeappliedaboutthecenteroftheimage.Setnegative*−*valuedpixelsto*0*andpixelswithval*
forcesearchtofindtheangleθ and translation t_x to optimally align the modified no-flash image with the flash image, so as to minimize the joint entropy. The range for θ is between -60 and +60 in steps of 1 degree, and the range for t_x should be between -12 and +12 in steps of 1. Compute the joint entropy using a bin-size of 10 for both intensities. Plot the joint entropy as a function of θ and t_x using the surf and imshow commands of MATLAB. Also, determine a scenario where the images are obviously misaligned but the joint entropy is (falsely and undesirably) lower than the 'true' minimum. Again, display the joint entropy as mentioned before. Include all plots in your report. [7 points]