Optical Flow and Tracking

Your goal in this assignment is to extend the optical flow estimation algorithm in the motion tutorial. The motion tutorial code uses a translational model of the optical flow within a region of interest (ROI). It implements an iterative estimator based on integer warps (preshifts) and a least squares (LS) gradient-based estimator to find the residual motion after each warp. The iteration continues until the magnitude of the residual horizontal and vertical velocities are less than approximately 0.5 pixels/frame.

1. [2 marks] As explained in the course notes, the normal equations for the gradient-based LS estimator of 2D image velocity can be expressed as

\[ Mv = b, \]

where \( M \) is the sum of outer products of spatial image gradients at points within the ROI, and \( b \) is the sum of the spatial gradients multiplied by their respective temporal differences. An SVD of \( M \) can be used to determine how well constrained the velocity is. Roughly speaking, the smallest singular value of \( M \) measures the total signal strength of the gradient constraints in that direction least constrained by the gradient constraints in the ROI. Generally, it depends on the size of the neighbourhood and the distribution of gradient magnitudes and orientations.

For frames 320 to 330 of the fleetface sequence, find two examples of the aperture problem in practice. That is, find two regions in which the smallest singular value is sufficiently small that the local image structure is approximately one dimensional, and the velocity estimates are very poor as a result. Demonstrate both cases with a simple demo matlab program; show the ROI, the singular values of \( M \), and a plot of the constraint lines in 2D velocity space. Also show the estimated velocities by superimposing the translated ROI over the images over 5 to 10 frames.

Using the same frames of the same image sequence, find another ROI that clearly overlaps two surfaces so that some of the gradient constraints come from one surface, while the remainder come from the other surface. By showing the estimated velocity of the ROI over 5-10 frames show that such ROIs also often produce poor motion estimates. Also show the constraint lines in velocity space for each frame. Is there anything in the structure of the constraint lines that might indicate that the estimated velocities might be poor.

3. [3 marks] Outliers are a problem in many estimation problems, and optical flow is no exception, especially with large ROIs. Many successful optical flow algorithms therefore use some form of robust estimation. One such robust estimator can be derived from the Geman-McLure error function

\[ \rho(\epsilon) = \frac{\epsilon^2}{\epsilon^2 + \sigma^2}, \]

where \( \epsilon \) is the residual error in a gradient constraint. The influence function corresponding to (2) is given by

\[ \phi(\epsilon) = \frac{2\epsilon\sigma^2}{(\epsilon^2 + \sigma^2)^2}. \]

With the Geman-McLure error function, derive an iteratively reweighted LS (IRLS) estimator for the translational image velocity in the ROI. That is, given an initial guess \( v_0 \), and a value for \( \sigma \), give equations for \( \epsilon \) and for the objective function we need to minimize, and derive equations for the weights, and for the weighted LS estimator.

4. [4 marks] Implement the robust flow estimator derived above for Question 3. Initially, assume that the initial guess for the optical flow is given by a LS estimator. For the LS estimator, in the first iteration, assume that the preshift between the two image ROIs is \((0, 0)\). Finally, as with the LS estimator in the motion tutorial, apply this robust estimator iteratively, using integer preshifts (warping) to reduce the velocity and increase the accuracy of the estimator with each iteration.
At each iteration of the IRLS estimator, the Geman-McLure error function requires an estimate of the standard deviation, \( \sigma \), of the residual noise of the inlier gradient constraints. One can assume that the residual errors for inlier gradient constraints are Normally distributed. Then, given the current velocity estimate \( v \), and the constraint errors \( e(x) = \nabla I(x) v + I_t(x) \), assume that the median absolute constraint errors \( |e(x)| \) is approximately equal to the median of the absolute values of fair samples from the Normal distribution. The median of the absolute value of samples from a Normal distribution is roughly 2/3 the standard deviation of the Normal. Therefore let the Geman-McLure scale parameter at each iteration be \( \sigma = 1.5 \times \text{median}(\{|e(x_j)|\}) \).

Demonstrate the estimator with a demo program. For each of two ROIs pre-selected by you, show the original image ROIs at one time and then at the next time. These can be displayed back and forth in the same window to illustrate the motion between them. Your demo program should show your results at each iteration of the estimation algorithm, including 1) the difference between the original image in the first frame and the warped image from the second frame warped toward the first, 2) a plot of the constraint lines in velocity space, 3) the estimated standard deviation \( \sigma \), and 4) the location of the ROI in the second frame.

When plotting constraint lines, plot the inliers and outliers in separate colors. To determine the putative inliers, we can use the weights. To do this, note first that the Geman-McLure influence function will initially increase (quadratically) as a function of error near the origin. Then as the error increases further, the influence function will eventually cease to increase and then begin to decrease. By decreasing it is giving less influence to constraint with particularly large errors (outliers). It is common to define outliers to be those constraints for which the influence is descending (i.e., has negative slope). As a function of \( \sigma \), find the point at which the influence function is maximal. From this, find the threshold on the weights (again, as a function of \( \sigma \)) that can be used to identify inliers and outliers. (Hand in these derivations.)

Finally, in some ROIs you may find that \( \sigma \) becomes very small. Can you find such an ROI? What happens to the set of inliers if \( \sigma \) becomes close to zero? In general, as discussed above, one expects that \( \sigma \) should approximate the standard deviation of the noise in the inlier constraints. Let’s assume 2 bits of noise in the difference between the two frames when they are perfectly registered. When not registered, the noise in the image difference will also depend on the displacement between the images and the magnitude of the image gradients. Taking these two sources of noise into account, with a displacement error of about half a pixel, and using the median gradient magnitude, one might place a lower bound on \( \sigma \) at something like \( 4 + 0.5 \times \text{median}(||\nabla I||) \). This should help to control the behaviour of the robust estimator.

5. [3 marks] Apply this algorithm to a sequence of 25 frames for an ROI of the markers choosing. You could for example ask the user to enter 4 numbers for the upper left and lower right corners of the ROI, you could also indicate in the matlab demo file where the markers ROI needs to be entered. At each time step show the ROI superimposed on the current image once the velocity is computed. Show the stabilized ROI. That is, warp each frame of the sequence (or just the image in the neighbourhood of the ROI) back to the initial frame of the sequence. When the velocity is well estimated then one would expect the resulting image sequence to be stabilized; i.e., after the warping of each frame there should be little or no apparent motion left.

Also for each frame of the sequence, plot the mean squared image error between the original ROI and the warped ROI. Discuss whether you can use this statistic as a diagnostic to determine whether or not after 25 frames you are still tracking the region of the image corresponding to the ROI in the first frame. That is, are you still tracking the right thing? If you can think of another statistic that may be a better diagnostic, try it out and report the results.

6. [3 marks] At each frame, given the translational motion already estimated by the robust estimator, find a robust affine warp from the ROI in the current frame back to the ROI at the first frame. The model for an affine warp is given in the course notes, as is the form of the LS estimator. For the robust estimator use the German-McLure error function and an IRLS solver, very much like that used to estimate the translation motion. For translational motion we used an iterative estimation algorithm that involved iteratively warping the second image and then re-estimating the velocity. For this problem, to find the affine warp, you do not need to apply the estimation iteratively with re-warping at each iteration.

With this implemented in addition to the tracking in Question 5, at each frame, plot the mean squared image error between the original ROI and the affinely warped ROI at the current frame. Discuss (and demonstrate) whether there are simple statistics of these affinely warped ROIs which might be useful as diagnostics to determine whether,
or not, after 25 frames you are still tracking the same thing.

**Bonus Problem. [1 mark]** Use an iterative algorithm with rewarping to find the affine warp in Question 6. For rewarping you can use the matlab interp2 function and bilinear or cubic interpolation. Does this improve the affine warp estimation?

**What to hand in**

Write your Matlab code for the different questions in the style of a Matlab "tutorial". Break out important pieces of code into functions to keep it manageable where suitable. Make sure your Matlab "tutorial" is well commented so the marker can run it and understand what you have done. Your comments are his evidence that you understand what you are doing.

The Matlab code should display important intermediate results where appropriate, using comments in the matlab script to indicate to the marker what is being displayed and why.

Also hand in written work and derivations asked for in the questions above. Also include any written work to support any other issues underlying your implementation. Hand-written and typeset formats are both acceptable.