CPS296.1- Homework 5

Due on November 20, 2007

Questions may continue on the back. Please write clearly. What I cannot read, I will not grade.

The Lucas-Kanade tracker finds the displacement vector $\mathbf{d}$ that minimizes the Sum of Squared Differences (SSD) $\epsilon(\mathbf{d}, \mathbf{x}_0)$ between a window $\mathcal{W}(\mathbf{x}_0)$ centered at $\mathbf{x}_0$ in the first frame $I(\mathbf{x})$ and a window in the second frame displaced by $\mathbf{d}$, that is, in $J(\mathbf{x} + \mathbf{d})$:

$$\epsilon(\mathbf{d}, \mathbf{x}_0) = \sum_{\mathbf{x} \in \mathcal{W}(\mathbf{x}_0)} \left[J(\mathbf{x} + \mathbf{d}) - I(\mathbf{x})\right]^2 \, w(\mathbf{x}) \; .$$

Here, $w(\mathbf{x})$ is a Gaussian-shaped weighting function centered in the window.

An approximate solution to this problem leads to the 2×2 linear equation

$$A\mathbf{d} = \mathbf{b}$$

where

$$A = \sum_{\mathbf{x} \in \mathcal{W}(\mathbf{x}_0)} \nabla I(\mathbf{x}) (\nabla I(\mathbf{x}))^T \, w(\mathbf{x})$$

and

$$\mathbf{d} = - \sum_{\mathbf{x} \in \mathcal{W}(\mathbf{x}_0)} \left[J(\mathbf{x}) - I(\mathbf{x})\right] \nabla I(\mathbf{x}) \, w(\mathbf{x}) \; .$$

In these expressions, $\nabla I(\mathbf{x})$ is the gradient of the first frame at $\mathbf{x}$.

The tracker solves this equation iteratively. At every iteration, it shifts the second frame by $\mathbf{d}$, and accumulates $\mathbf{d}$ to find the overall displacement.

The Matlab file `badTracker.m` in the zip file on the homework web page does some of this work, but not all. It uses a window of size 7×7. Please read this code carefully, and recognize the expressions above in it. You also need the auxiliary files supplied in the zip file.

(a) Print a copy of `i_0001.pgm` and use some symbol to mark the following points in (row, column) format: $p_1 = (342, 31)$, $p_2 = (297, 129)$, $p_3 = (279, 200)$, $p_4 = (408, 400)$, $p_5 = (139, 457)$, $p_6 = (248, 61)$.

(b) One of these points is obviously not a good feature to track. Which, and why?

(c) Run the `badTracker` on this sequence, and on the six points given above (including the bad one). Mark the final position of each of the points that survive in the last frame, and print the picture.

(d) Look carefully at the tracking results. For each of the six points, state which is tracked well and which is not. Briefly describe the errors. A position error of magnitude comparable to the window size at the end of the sequence is a bad error.

(e) What is the main problem with `badTracker.m` ?

(f) Modify `badTracker.m` to address this problem. Explain briefly what you do, and hand in the code. If new problems arise with your modifications, address those as well. If your tracker somehow loses features, it should realize this, and set their coordinates to `NaN` ("Not a Number"). Please mark your modifications and additions clearly with comments. Make your tracker determine feature positions to within a fifth of a pixel from the best solution (unless it fails to track altogether).

(g) Show the results of your tracker superimposed on the last frame of the sequence, as you did above for `badTracker.m`.

(h) Can you tell which way the camera moves? For this, it may be useful to plot the tracks of the points. If you cannot tell from these tracks, track more points.

(i) Look at your tracking results for point p_6: understand what the scene looks like there, by looking at the full image. Then blow up the image and look at it in detail, comparing the positions of point p_6 in the first and last frame. What problem do you see in the tracking result for this point? This problem would cause errors in a reconstruction algorithm that uses this feature and assumes a static scene. Explain what happens, and why. This problem is not the tracker's fault.