Motion Estimation & Tracking Oct. 25, '04

References:

- motionTutorial.m
- ~jepson/pub/matlab/utvisToolbox/tutorials/motionTut
- Your lecture notes on Motion, Optical Flow and Tracking

Motion Estimation & Tracking

The Brightness Constancy constraint states that as pixels move around in an image sequence, their brightness remains the same. The underlying idea is that the brightness of a given pixel that is located at (x, y) at time t_0 travels to position $(x + dx, y + dy)$ at time $t + dt$, where $dx = v_x \cdot dt$, $dy = v_y \cdot dt$, and (v_x, v_y) are the components of the optical flow vector at (x, y), at time t_0.

The above constraint can be written in several ways, here are a few of them:

$$\mathbf{v}_x I_x(x, y, t) + \mathbf{v}_y I_y(x, y, t) + I_t(x, y, t) = 0 \tag{1}$$

$$I_1(x + v_x, y + v_y) \approx I_0(x, y) \tag{2}$$

$$\frac{\partial}{\partial t} I(\vec{x}(\vec{p}, t), t) = 0 \tag{3}$$

$$\nabla I(x, y, t) \cdot \vec{v}(x, y, t) + \frac{\partial}{\partial t} I(x, y, t) = 0 \tag{4}$$

Where $I(x, y, t)$ is the image at time t, $I_x(x, y, t)$, $I_y(x, y, t)$, and $I_t(x, y, t)$ are the image derivatives along x, y, and the temporal dimension. And v_x and v_y are the x and y components of the velocity vector.

The above equations encode a constraint for each pixel position (x, y). One important aspect of the above is that the equations are under-constrained (one equation and two unknowns per pixel). The result is that local velocity measurements are not possible, the motion is only constrained to lie along a line in $\vec{v}$ space perpendicular to the image gradient.

Notice also that the brightness constancy constraint does not hold if there are changes in lighting (including changes in shading due to the angle between the surface normal and the light source direction), or at occlusion boundaries. However, it offers a useful way of estimating the optical flow.

Motion Estimation and Tracking

The tutorial code lets you select a region in an image, and track it across an image sequence. As in the lecture notes, it assumes a near identity warp, with the additional simplification that only translation is considered, hence $\vec{W}(\vec{x};\vec{a}) = \vec{x} + \vec{a}$.

Given the above, the code estimates a Least Squares solution to the objective function

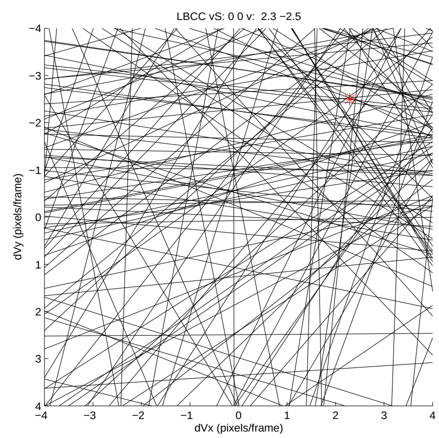
$$\tilde{E}(\vec{a}) = \sum_{\vec{x} \in R} \rho(\nabla I_R(\vec{x}) \cdot \vec{a} + [I_R(\vec{x}) - I_L(\vec{x})]) \qquad (5)$$

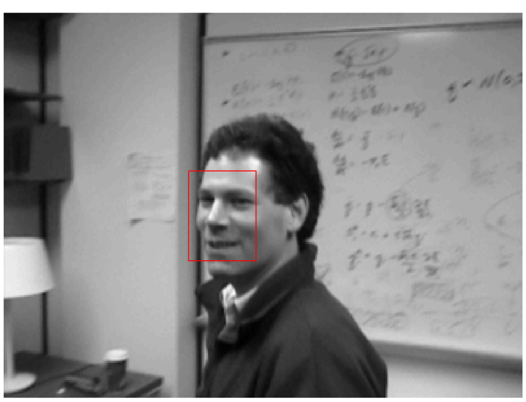
The code uses the least squares estimator $\rho(e) = \frac{1}{2}e^2$, and plots the brightness constancy constraints for each pair of frames.

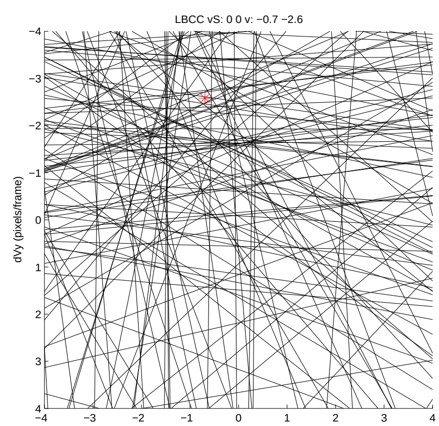
Let's see some sample results...

Motion Estimation and Tracking

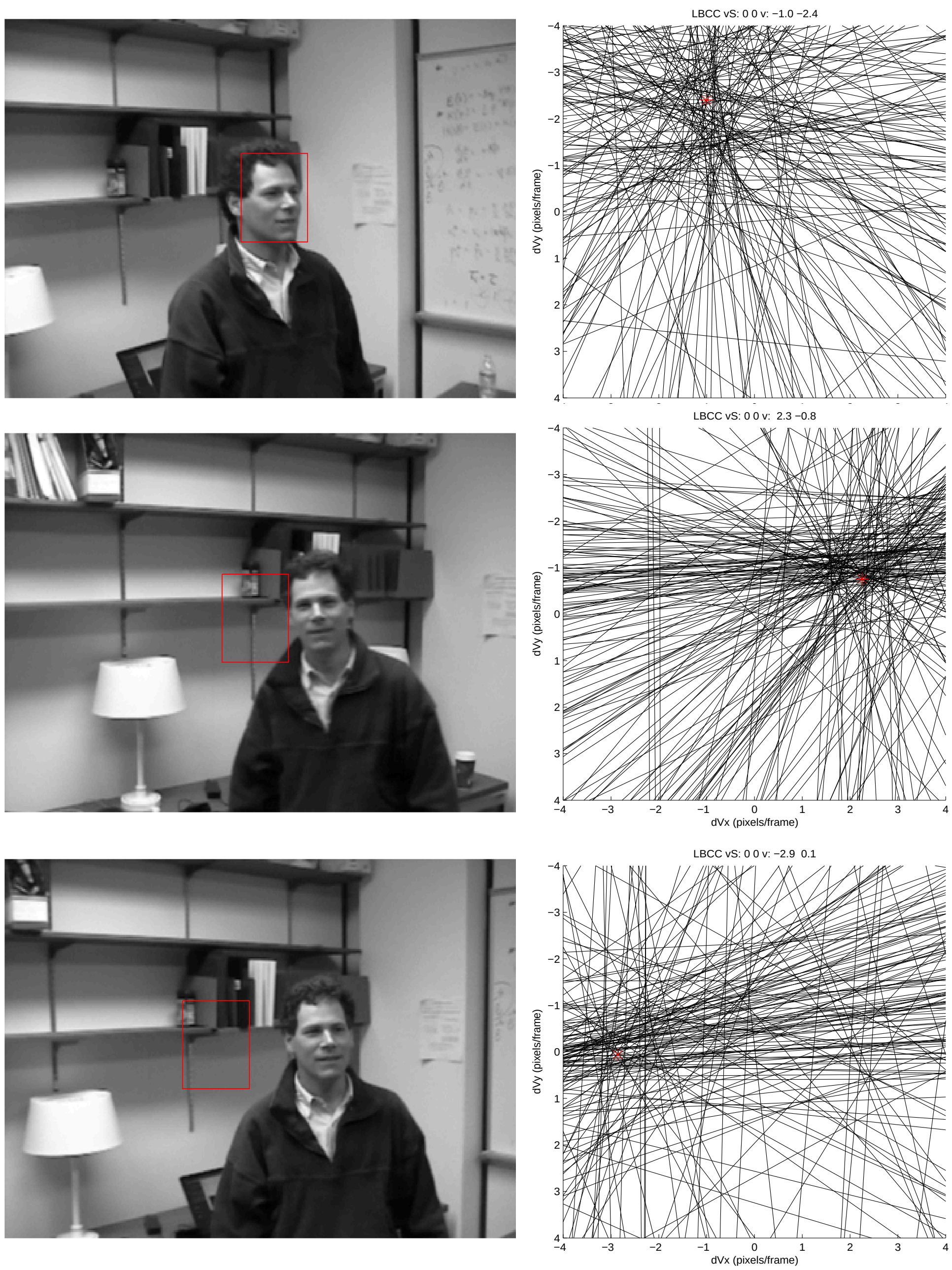
Tracking Results 1: Simple Least Squares estimate







Motion Estimation and Tracking



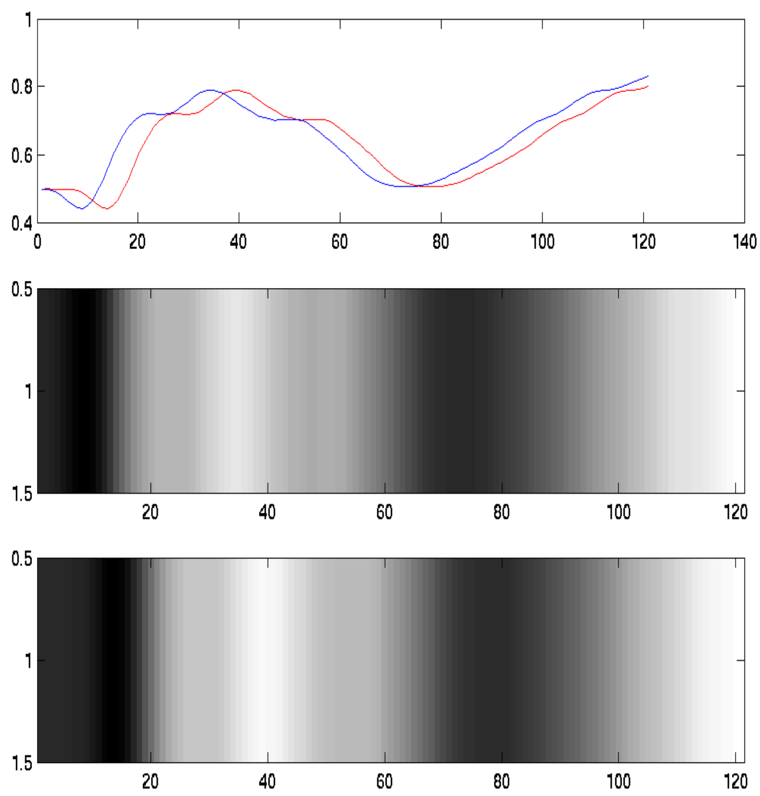
Motion Estimation and Tracking

Observations:

- The least squares estimate does remarkably well even though the displacements between frames can be large, and even though the appearance of the region varies significantly (think about what this means in terms of the brightness constancy constraint).
- However, accumulated errors in the estimates as well as violations of the assumption that there is a single motion within the region of interest cause the algorithm to eventually lose the patch being tracked.
- This is expected and quite reasonable given the type of motion and the changes in the appearance of the tracked region. Notice in the constraint plots that for several of these it is not clear at all what the dominant motion is.

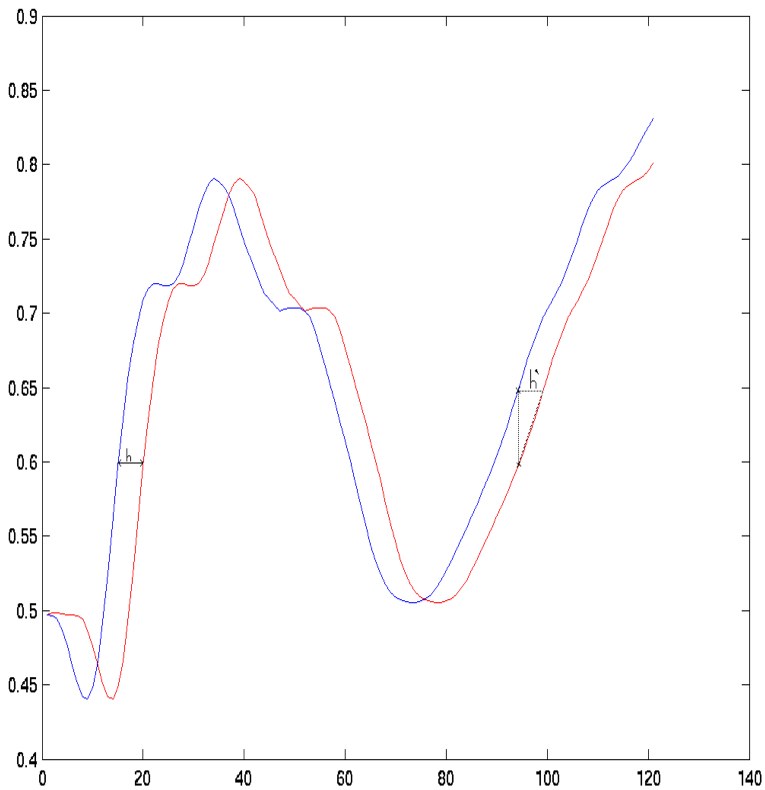
Motion Estimation and Tracking

What is the problem with large motions from frame to frame? consider the image below, it shows two frames of a single row of an image. From the intensity images it is easy to see that a simple shift to the left took place between the top and bottom frames. The graph at the top shows the intensity profiles for both frames.



Motion Estimation and Tracking

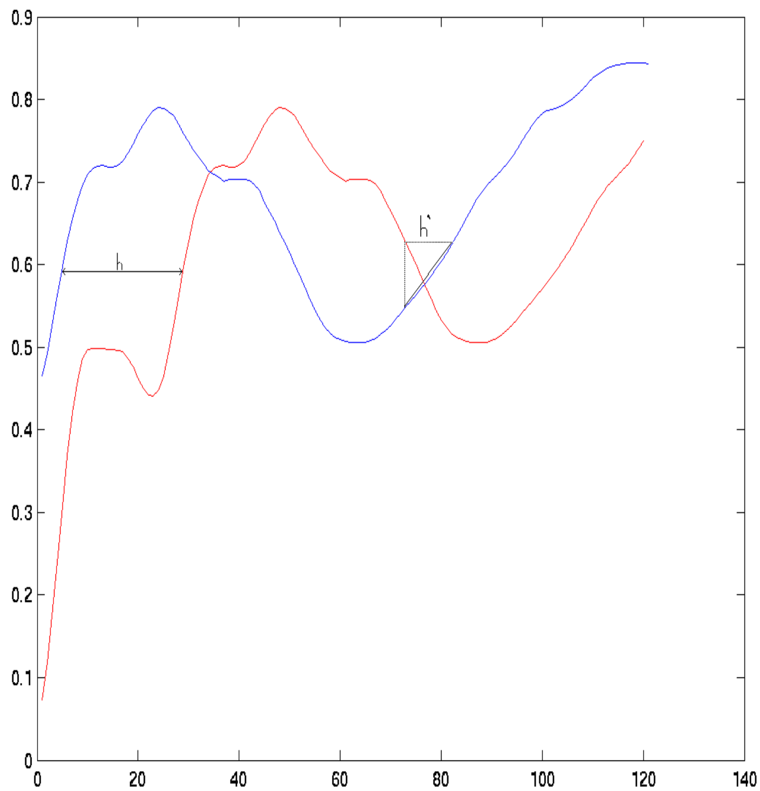
The graph below is an enlargement of the intensity profiles for both frames.



Notice that the estimate for the displacement (labeled h') that we would expect from our linear model is quite close to the real h. This works nicely due to the linear behavior of the intensity profile within the range of the displacement h.

Motion Estimation and Tracking

The graph below is an enlargement of the intensity profiles for two frames where the displacement is much larger.



Now it is clear that the estimate obtained from our linear model is inaccurate. The magnitude of the motion is such that our linear model no longer applies over the width of the displacement h.

One way to deal with this problem is to increase the blurring in each of the frames of the sequence, increasing the blur extends the range of displacements over which the linear approximation is valid, however, it has the downside of reducing the resolution that we can achieve in the motion estimates.

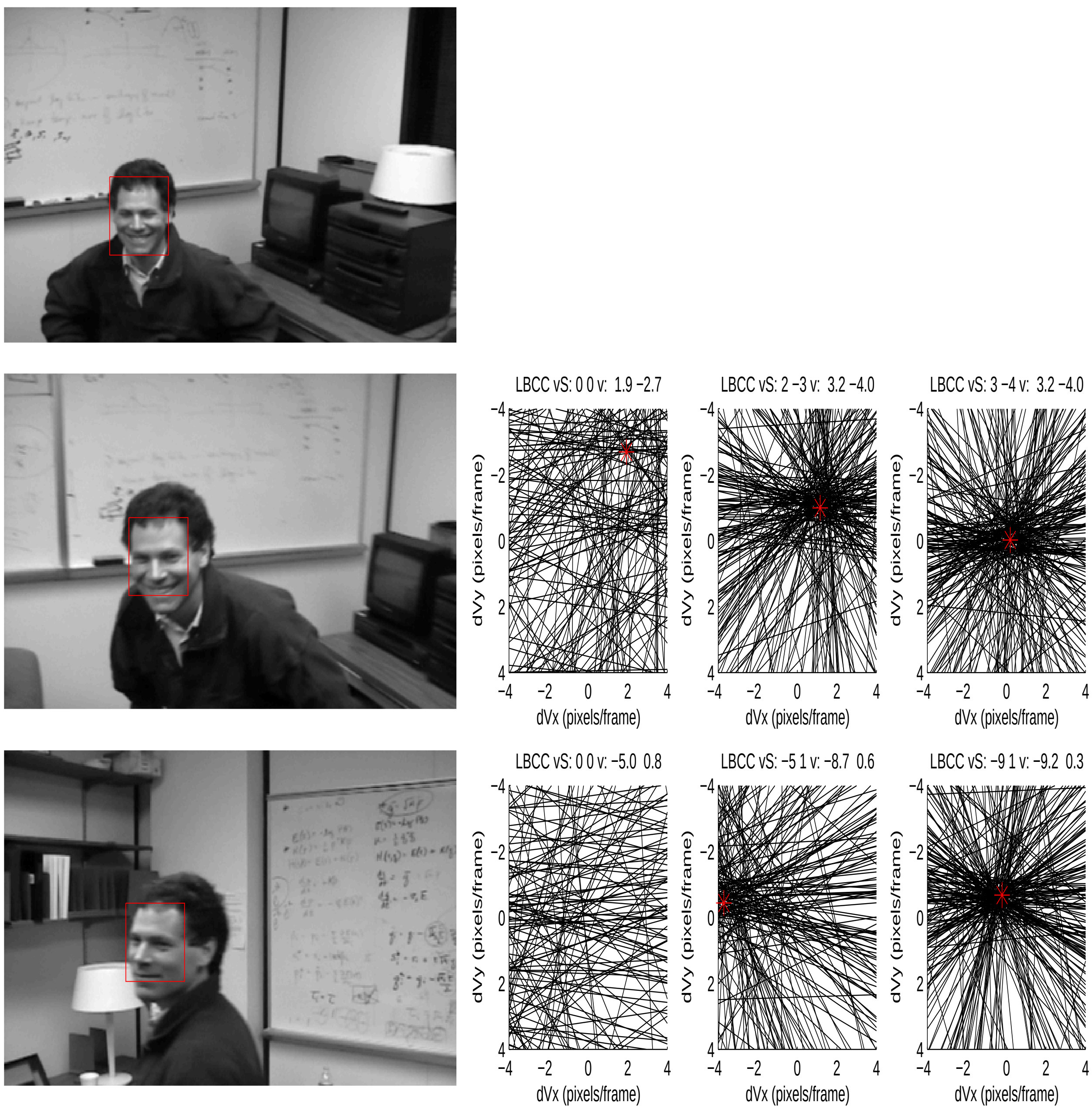
Motion Estimation and Tracking

We need something more than the original gradient constraints and the least squares estimator for tracking purposes. In the motionTutorial.m script two techniques are used to deal with the large displacements between frames:

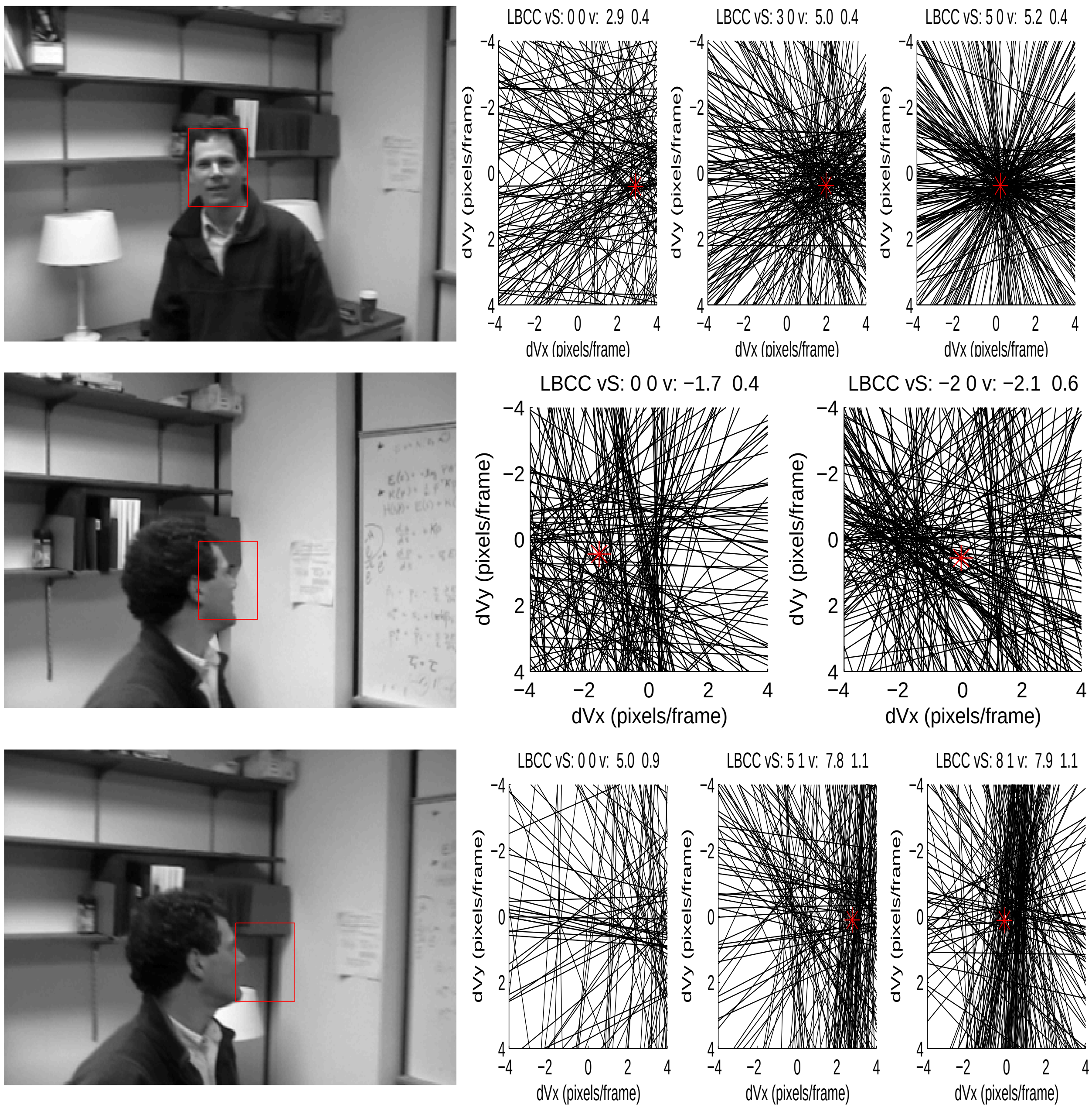
- **Iterative Rewarping:** compute an initial estimate of $\vec{v}(x, y, t)$, estimate the translation of the region of interest and warp the image accordingly, then re-compute the least squares estimate of $\vec{v}(x, y, t)$. Repeat this procedure until the estimate converges.
- **Use the estimate of** $\vec{v}(x, y, t-1)$ to compute an initial shift for the region of interest, then compute $\vec{v}(x, y, t)$ with or without iterative rewarping.

Motion Estimation and Tracking

Tracking Results 2: Iterative rewarping, no initial shift



Motion Estimation and Tracking



Motion Estimation and Tracking

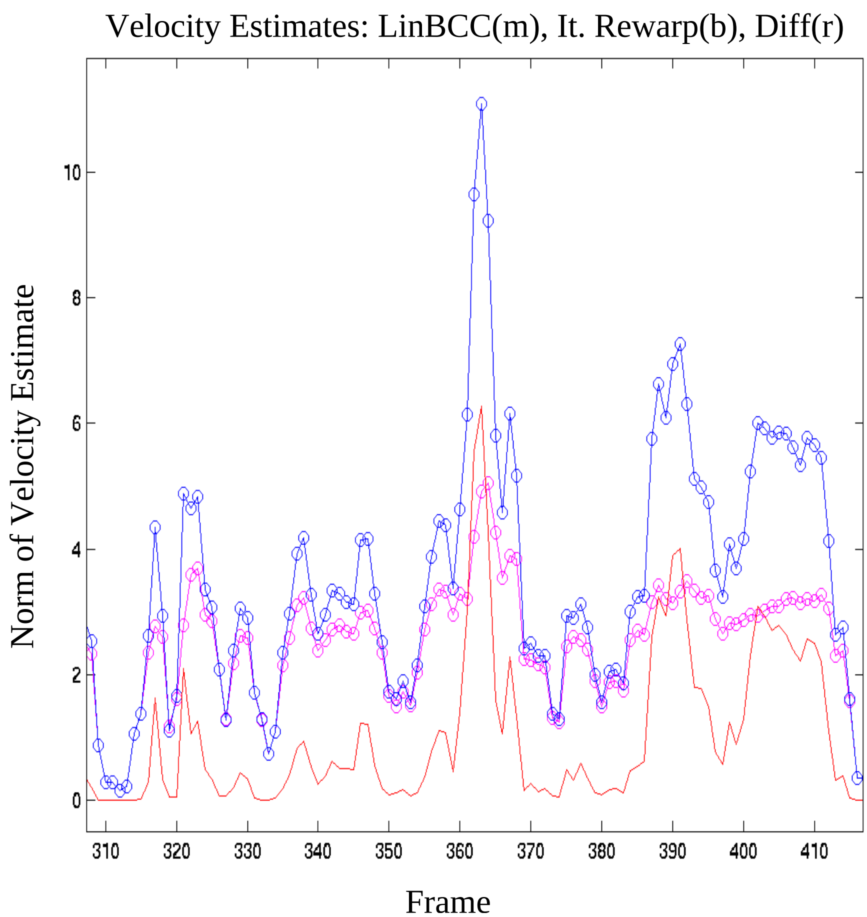
Observations:

The tracker stayed with the face much longer. Notice from the constraint plots that for large frame-to-frame displacements the initial plot does not contain a well defined solution, so the initial estimate is erroneous. Iterative rewarping refines the estimate, and with each successive iteration, the solution becomes better resolved and closer to the origin of $dV = (0, 0)$.

The sequence also illustrates that the Least Squares estimate can lead to unexpected solutions. At the point where the tracker loses the face, there are two different motions in the same patch, the face moving to the left, and the background moving to the right. The least Squares solution in this case is poor (a weighted average of the linear constraints from the two motions). In the end the tracker becomes locked onto the background.

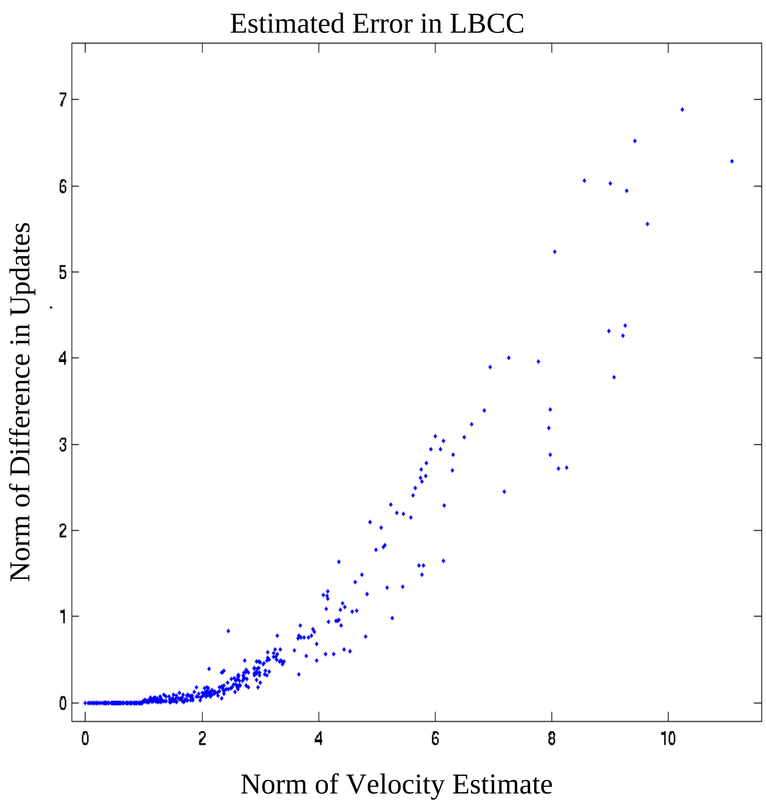
Motion Estimation and Tracking

The graph below shows the norm of the velocity estimates for the Linear BCC (magenta) and iterative rewarping (blue) procedures. The difference between them is also displayed (solid red line). Notice that the error is large where the velocity estimated by the iterative rewarping procedure is large, and smaller otherwise.



Motion Estimation and Tracking

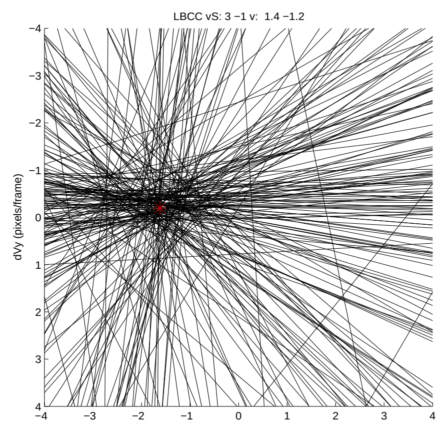
Another way to look at this is to plot the difference between the updates from the Linear BCC, and the iterative rewarp (red line in the previous plot) against the magnitude of the estimated velocity update.

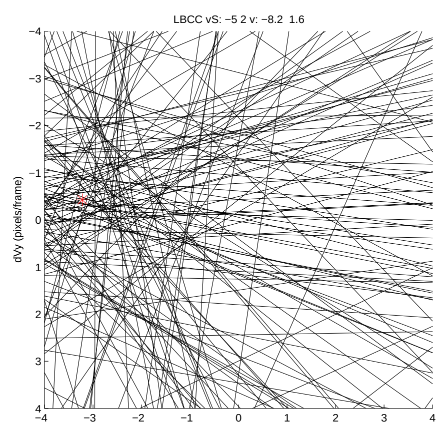


You can see that the error grows in a quadratical fashion beyond the point where the norm of the velocity update is equal to the sigma used to blur each frame (in the above sequences, $sigmaBlur = 2$).

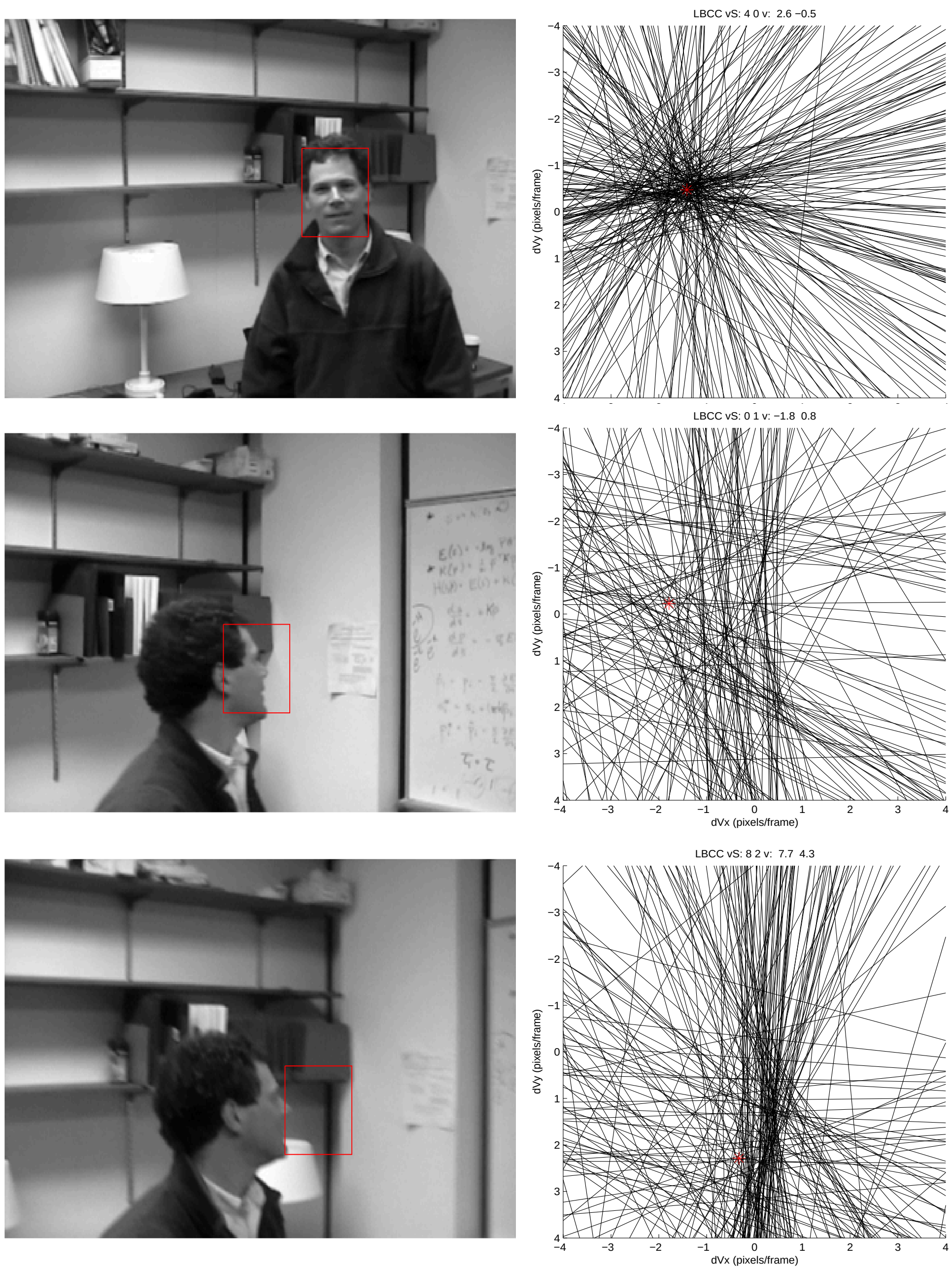
Motion Estimation and Tracking

Tracking Results 3: No iterative rewarping, non-zero initial shift





Motion Estimation and Tracking



Motion Estimation and Tracking

Observations:

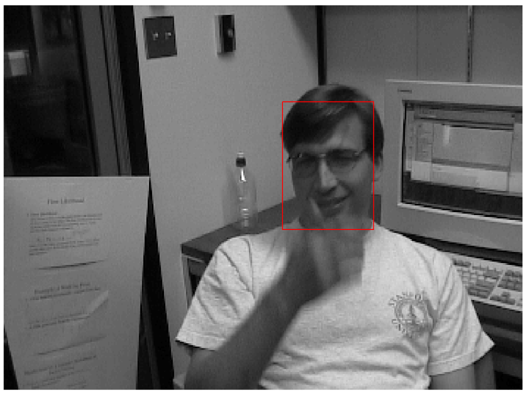
This sequence illustrates a different way of dealing with large frame-to-frame shifts, an initial motion estimate is determined from previous frames, and used to move the tracking window at the start of the estimation process to a place that will likely be close to the actual solution.

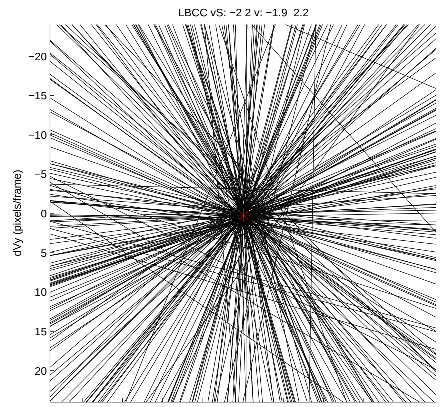
The initial shift is usually close enough to the true estimate, that one iteration of the Least Squares estimator is enough to recover the motion within the patch.

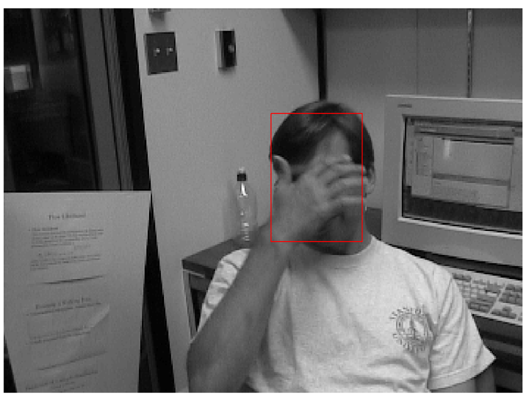
Under the above conditions, the tracker is also able to stay with the face much longer than the least squares version. At the end of the sequence, just like the iterative procedure, the tracker becomes attached to the background.

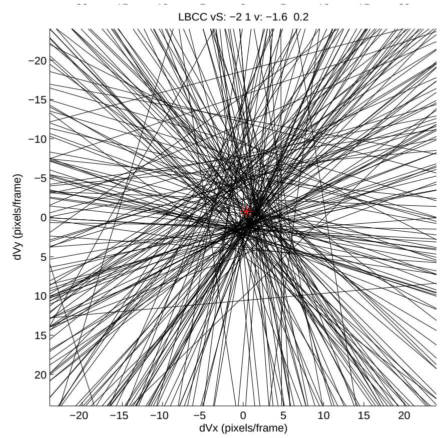
Motion Estimation and Tracking

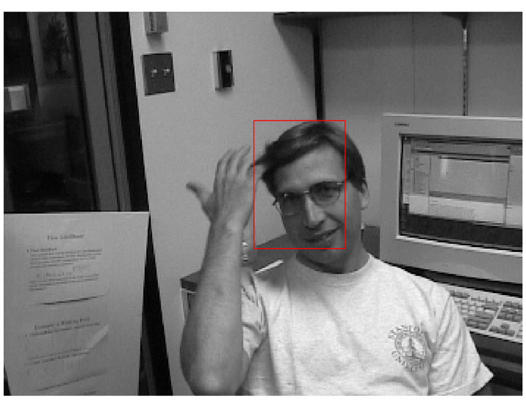
Tracking Results 4: No iterative rewarping, non-zero initial shift

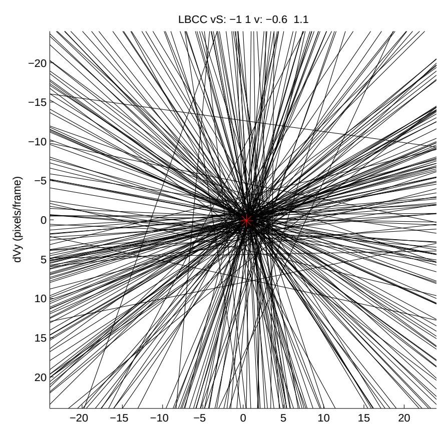




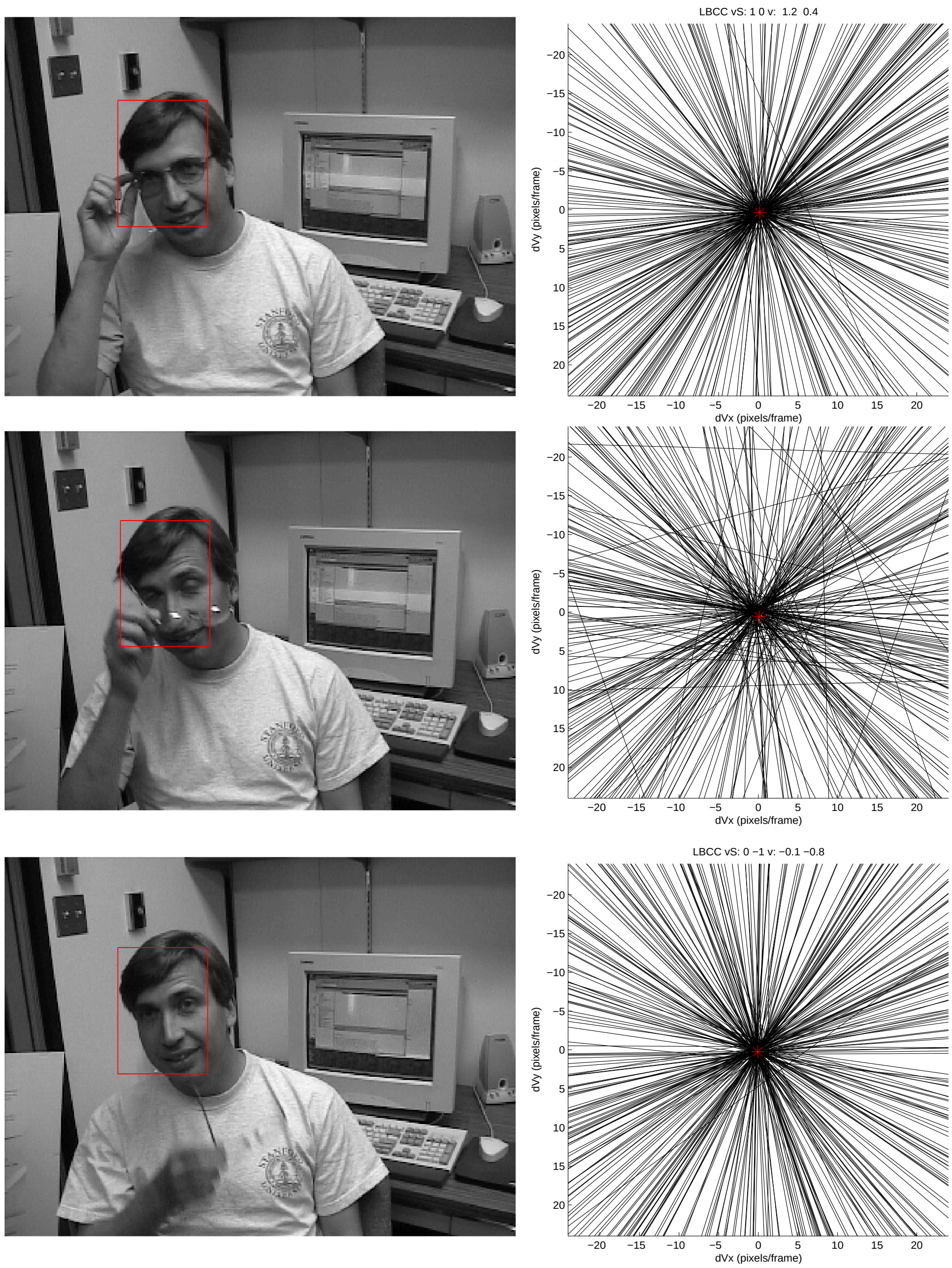








Motion Estimation and Tracking



Motion Estimation and Tracking

This sequence illustrates two cases of occlusion, each triplet of frames shows the region being tracked before, during, and after it is crossed by an occluding object. In both cases, the occluding objects cross the tracking window too fast for the tracker to pick up (notice that the constraint plots show a dominant motion close to $(0, 0)$ in $v - space$).

Motion Estimation and Tracking

Notes:

- motionTutorial.m uses the previous motion estimate to compute an initial shift for the tracking window. There are other ways in which we can obtain a prediction of the motion from one frame to the next, for example, we could use a Markov model to generate a prediction distribution $p(\vec{v}(x, y, t)|\vec{z}_{1:t-1})$, which is based on taking a Gaussian distribution over possible velocities at t, and passing it through appropriate model dynamics (see your notes on tracking!).
- Instead of calculating a single estimate for $\vec{v}(x, y, t)$, we could use a probabilistic optical flow technique to obtain a 2-D distribution that gives a posterior probability for $\vec{v}(x, y, t)$. (See Simoncelli, Adelson, and Heeger, *Probability Distributions of Optical Flow* , CVPR, 1991).