Assignment 3 Description

Yan Miao
260711311

Question 1:
Illustrate and discuss results for frames in the Basketball and Backyard sequence.

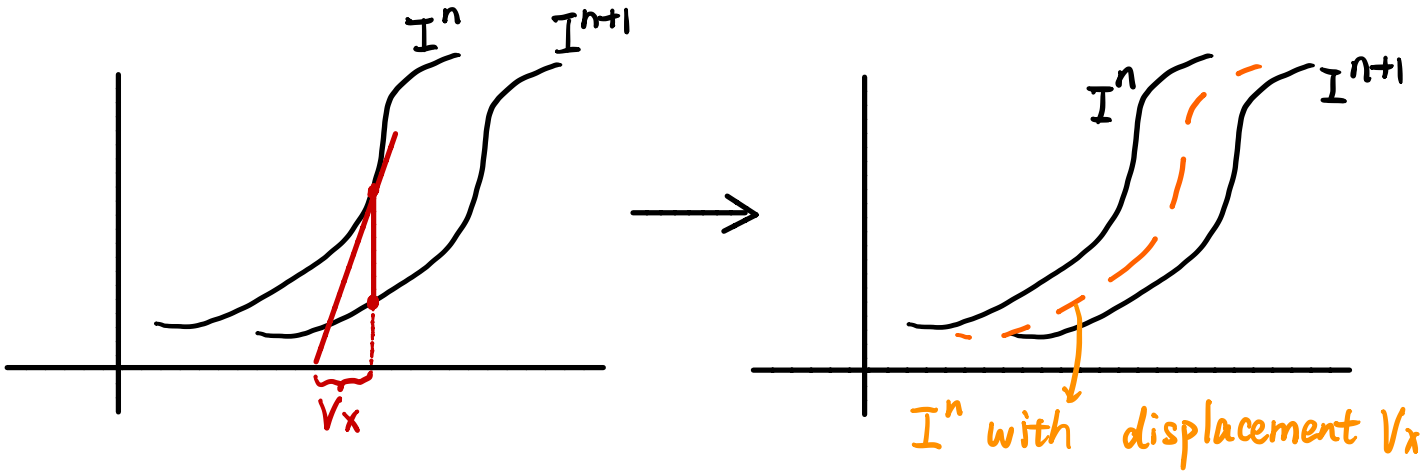
Illustration:

Basketball sequence with window size = 31

Backyard sequence with window size = 49

Discussion:
Performance of the algorithm:
For the Basketball sequence, we can see there are velocity vectors over the two people and the ball in the image and the velocity vectors are pointing at the directions which the objects are moving towards at the instant. When we focus on the ball and the hands of the man on the right hand part of the image (where the motion is relatively large), we can find velocity vectors over them. But most of the velocity vectors are in small magnitude and some of them are even pointing at wrong directions.
Similar things happen to the Backyard sequence. Velocity vectors are over the four kids in the image. Most of them are pointing at the directions the kids' moving towards and having small magnitude while the remaining ones are pointing at weird directions. Notice for the orange ball in the image, it is undetected in some of the frames and has velocity vector with wrong directions over it.

Limitations of the one-shot algorithm:
1. The algorithm assumes that the image function I(x) is linear in the neighborhood of any point (x0,y0) in the image by using only the first derivative term in Taylor Expansion. However, I(x) is not linear in most of the cases. This can cause problem.



As we can see from the figures above, we approximate the displacement (i.e. the velocity vector) using the tangent line. After only one displacement, I(n) still cannot perfectly move to the location where I(n+1) lies. This is the reason why we need to apply the iterative method: when the iteration goes on, we update the velocity vectors and the difference of the slopes of I(n) (after displacement) and I(n+1) at the same location will decrease, so ideally the value we used to update the velocity vectors will decrease as well.

2. The algorithm can only recover small motions. As we can see from the previous illustration of the Backyard image sequence, the velocity vectors in the neighborhood of the orange ball are often fail to describe the motion of the ball, which is probably because of the large displacement of the ball in two consecutive images.

Question 2 (iterative method)

Discussion about the improvement of the results
For the Basketball sequence, we can see that the velocity vectors in the neighborhood of the ball and the hands of the man on the right are better than that output by the naive algorithm, meaning their magnitude get increase in general and their direction get refined. Still, we focus on these two parts mainly because of the relatively large motion they have.

For the Backyard sequence, the velocity vectors on the kids clearly get refined. Take the little girl in the red skirt for example, the output velocity vectors from the naive algorithm don't show much about motion field whereas the output from the iterative algorithm present a better version of motion field, where you can see the velocity vectors lining up in a more ordered way.

Basketball sequence with iterative method using window size = 31

Naive algorithm

Iterative method

Backyard sequence with iterative method using window size = 49

Naive algorithm

Iterative method

Question 3 (coarse-to-fine approach)

1. Results comparison

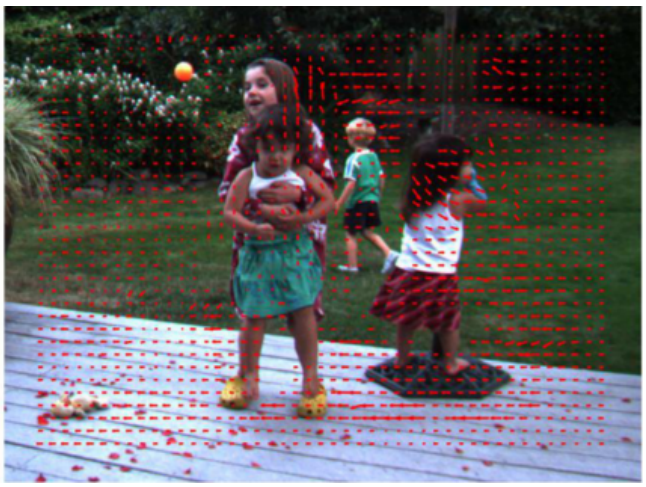
Basketball sequence with coarse-to-fine method using window size = 31

Iterative method

Coarse-to-fine

Backyard sequence with coarse-to-fine method using window size = 49

Iterative method



Coarse-to-fine

As we can see from the above comparisons, two sets of images are clearly refined by using a coarse-to-fine approach. Some large motions are captured, and both the magnitude and direction of the velocity vectors can better describe the motion field than using the previous two methods.

2. Discussion about the tradeoffs

If we perform fixed scale approach (like the iterative method in part b)) on the sequence of images, we will get the output velocity vectors in a relatively short amount of time, while some of the objects with large size or having large displacement may not have reasonable velocity vectors on them.

On the contrary, if we perform coarse-to-fine approach on the sequence of images, although we may need to wait for a while to get the output, the velocity vectors describe relatively precise motion field of the objects in the image.

Question 4

Modified iterative method for recovering general affine motion field (local translation, rotation and shearing)

We will introduce the deformation matrix D here to deal with the general case.
The relationship between the original image and image after transformation can be represented as follows:

$$I(\mathbf{x_0}+ \mathbf{D}(\mathbf{x} - \mathbf{x_0}) + \mathbf{h}) = J(\mathbf{x}) \qquad D = \begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

where x0 is the central pixel of the image patch, x is a pixel in the neighborhood of x0, D is the deformation matrix and h is a local translation just like [Vx, Vy].

When we want to recover local translation, we simply set D = I (identity matrix) and perform the iterative method.
As for recovering rotation, we let

D = $\begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$, which represents counter clock-wise rotation by theta.

For recovering shearing, we let

D = $\begin{bmatrix} 1 & \lambda_x \\ \lambda_y & 1 \end{bmatrix}$

Now that we have our deformation matrices in different cases, we need to find a way to solve for the parameters in the matrices. Following the iterative method, we use least-square approach the minimize the expression:

$$\sum_{\mathbf{x} \in Ngd(\mathbf{x_0})} (I(\mathbf{x} + \mathbf{D}(\mathbf{x} - \mathbf{x_0}) + \mathbf{h}) - J(\mathbf{x}))^2$$

We let $\mathbf{d} = (D_{11}, D_{12}, D_{21}, D_{22}, h_x, h_y)$ and $\mathcal{I} = (\frac{\partial I}{\partial x}\Delta x, \frac{\partial I}{\partial x}\Delta y, \frac{\partial I}{\partial y}\Delta x, \frac{\partial I}{\partial y}\Delta y, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$

Then we can transform the minimization problem to solving the following linear system:

$$(\sum_{(x,y) \in Ngd(x_0,y_0)} \mathcal{I}\mathcal{I}^T)\ \mathbf{d} = \sum_{(x,y) \in Ngd(x_0,y_0)} (J(x,y) - I(x,y))\ \mathcal{I}$$

We then solve for the vector d iteratively until the update values for each parameter are smaller than some small thresholds. This is where we have a convergence, which is the same idea as what we have done in part b).