Assignment3

Jingyuan Wang

Question(a)

First, a naïve Lucas-Kanade image registration is implemented, using the following formula:

$$\begin{bmatrix} \sum W(\frac{\partial I^n}{\partial x})^2 & \sum W(\frac{\partial I^n}{\partial x})(\frac{\partial I^n}{\partial y}) \\ \sum W(\frac{\partial I^n}{\partial x})(\frac{\partial I^n}{\partial y}) & \sum W(\frac{\partial I^n}{\partial y})^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = - \begin{bmatrix} \sum (I^n(x,y) - I^{n+1}(x,y))\frac{\partial I^n}{\partial x} \\ \sum (I^n(x,y) - I^{n+1}(x,y))\frac{\partial I^n}{\partial y} \end{bmatrix}$$

After adjustment to parameters, here are two of the sequences that has rather good results. We can see from the backyard image, that several motions were captured with the right direction, including the yellow ball dropping, the girl on right moving, two kids in the middle are ready to jump as well as the boy walking to the left although the velocity is rather small.

Backyard 10 and 11   window size = 5

In the selected basketball scenes, two main movement of the basketball and the man's hand were also indicated. Also we can see the dots in the middle suggesting the movement of the shadow.

Basketball 13 and 14   window size = 15

In the meantime, there are also limitations of the naïve function, in the following backyard image, we can find out that several wrong directions and meaningless red dots on the right higher part. The performance is different due to the different parameters we set, especially summing window size. Larger window size will normally capture faster movement while small window size is more sensitive to a rather slow movement.

Backyard 12and 13   window size = 5

Also it is noteworthy that in my codes, the window size, m, refers to the side length of the summing window, which needs it to be an odd number.

Question(b)

Here are two successive sequences of the basketball. Compared to the naïve function, these pictures refined a lot on motion capturing. The basketball and hand movement are more outstanding, and other small motions like shadows on the wall are moving right, man's head are slightly changed, etc.

The iterative method is better because the naive function suppose the image I(x) is linear in the neighbourhood of X_0. However, the image is not. Although we applied a Gaussian smooth to the images, I(x) still have second and higher order terms of the Taylor series, which makes the solution of the one-shot function is gets an estimate of h instead of the exact value. So we try to find the best estimate h_{k+1}, using k iterations.

Basketball 12&13, 13&14   window size = 19

If showing all the pictures in order, it is easy to discover that most of the red arrows in the middle, are constantly moving to the right indicating the location and velocity of ball and shadows.

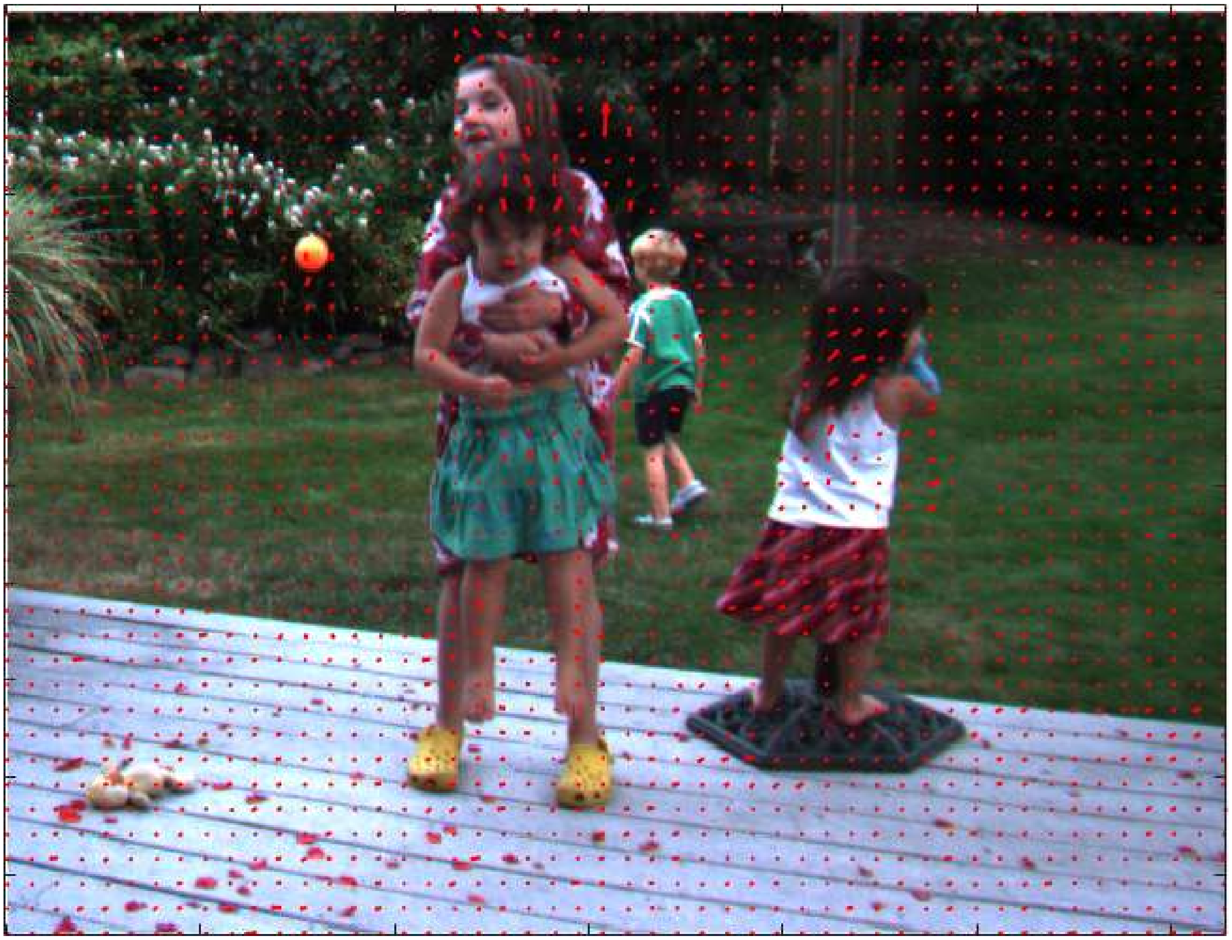
Question(c)

As discussed above, the size of integration window influence a lot on the motions captured. If the window is large, the algorithm will capture large displacement, i.e. object moving fast. On the contrary, for a slight change in the sequences, we can hardly capture it with a large window. Also, a fast movement can hardly be captured by a small window, since all the information are divided into different ones.

When comparing the image below with the second image in part b, it can be seen that more movements were found out in this method, like the movement of the shadow were more distinct as well as slight change in the two man’s clothes.

Basketball 13 and 14   window size = 19

Schefflera 9 and 10   window size = 15



Backyard 10 and 11   window size = 15

Also this method performs well in this sequence. Since most pixels are slightly rotated, we can sense that in the direction of the curved arrows.

Question(d)

In previous Lucas-Kanade function, we were finding pixels that satisfy, with two images I and J:

$$I(x + h_x, y + h_y) = J(x, y)$$

But to guarantee it recover all affine transformations, we need to add other parameters. Using a 2*2 matrix **D** to represent different transformations.

$$\begin{bmatrix} D_{11} & D_{12} \\ D_{21} & D_{22} \end{bmatrix}$$

For a uniform scaling transformation, D should be [s,0;0,s], for a rotation, D should be [cos(θ),-sin(θ);sin(θ),cos(θ)], etc.

So now we have:

$$I(\mathbf{x_0} + (\mathbf{I} + \mathbf{D})(\mathbf{x} - \mathbf{x_0})) = I(\mathbf{x} + \mathbf{D}(\mathbf{x} - \mathbf{x_0}) + \mathbf{h}) = J(\mathbf{x})$$

We use this formula and estimate along with vector **h**, to make the following property as small as possible.

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

Take a Taylor series expansion to it and we get:

$$I(\mathbf{x} + \mathbf{D}\Delta\mathbf{x} + \mathbf{h}) \approx I(\mathbf{x}) + \frac{\partial I}{\partial x}(D_{11}\Delta x + D_{12}\Delta y + h_x) + \frac{\partial I}{\partial y}(D_{21}\Delta x + D_{22}\Delta y + h_y)$$

Letting

$$\mathcal{I} = (\frac{\partial I}{\partial x}\Delta x, \frac{\partial I}{\partial x}\Delta y, \frac{\partial I}{\partial y}\Delta x, \frac{\partial I}{\partial y}\Delta y, \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y})$$

And we need to solve the minimum of the previous formula, take partial derivatives of it, get:

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

Then use iteration to solve for the six parameters of **d**:

$$\mathbf{d} = (D_{11}, D_{12}, D_{21}, D_{22}, h_x, h_y)$$