Q1

Algorithm Design

Load two successive frames, do a 2D Gaussian smooth on both of them, also smooth them in time by a 1D Gaussian. Then calculate the partial derivatives $\frac{\partial I}{\partial x}$ and $\frac{\partial I}{\partial y}$.

In the loop I make a window of size *ws,* and slide it on the image I1 first horizontally and then vertically. In each step, we need to calculate the least sum of square

$$\sum_{(x,y)\in Ngd(x_0,y_0)} \{I(x+h_x, y+h_y) - J(x,y)\}^2$$

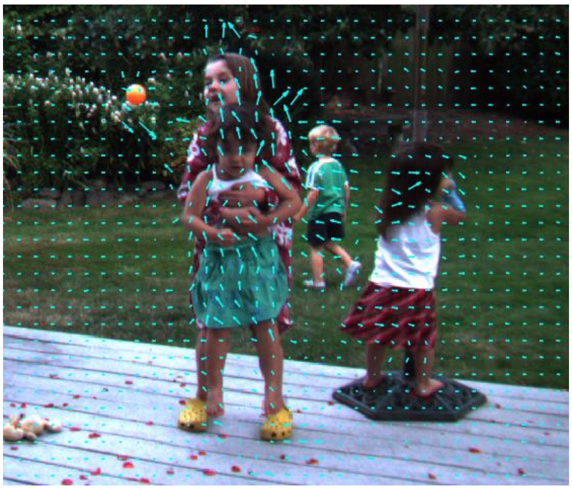
In this algorithm we use a Taylor approximation, then we just need to solve the equation

$$\begin{bmatrix} \sum(\frac{\partial I}{\partial x})^2 & \sum(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) \\ \sum(\frac{\partial I}{\partial x})(\frac{\partial I}{\partial y}) & \sum(\frac{\partial I}{\partial y})^2 \end{bmatrix} \begin{bmatrix} h_x \\ h_y \end{bmatrix} = - \begin{bmatrix} \sum(I(x,y) - J(x,y))\frac{\partial I}{\partial x} \\ \sum(I(x,y) - J(x,y))\frac{\partial I}{\partial y} \end{bmatrix}$$

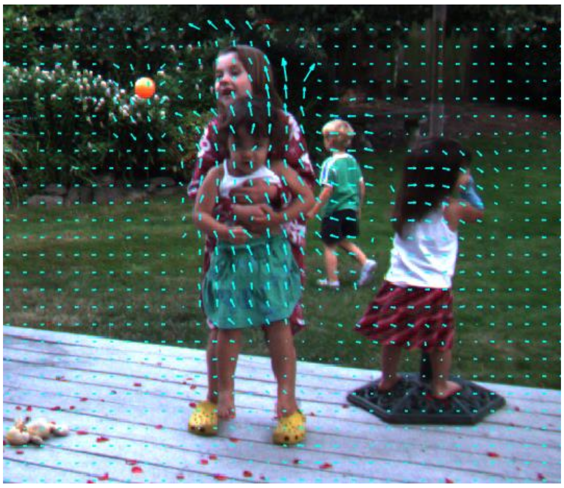
Then we get $(h_x, h_y) = (v_x, v_y)$, which is the motion filed of the center of the window.

Parameter Analysis

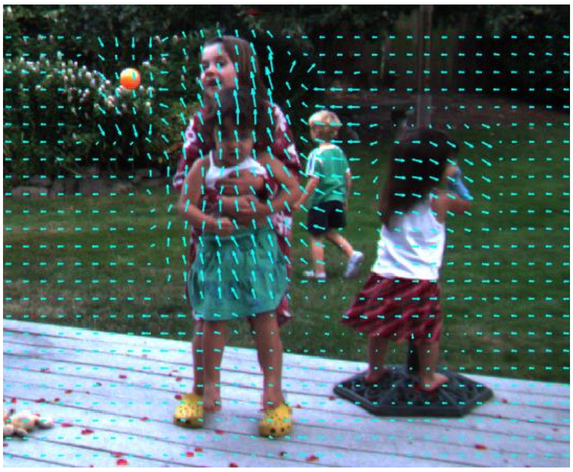
1. Smoothing factor in time (*sigma_t*):
   The 1D Gaussian smooth aims to make the pixel values differentiable in time. But since our frame sequences were short, changing *sigma_t* did not make apparent differences to my results. So I used *sigma_t* = 0.5 for all the experiments.
2. Window size (*ws*)
   This parameter defines the size of the neighborhood in the LK algorithm. The smaller the window is, the less it changes when it is shifted. So, if the moving object is big or moves fast between frames, it is hard for a small window to find the optical flow. On the other hand, the bigger window is better at capturing the motion. But the motion field will be easily affected by its neighbor area (i.e. the noise of the background) when the window is too large, which is actually not the major motion we want to detect.
   For example, look at the motion fields got from frame09-10 of Backyard of different window sizes, with sigma being unchanged. In this image we have four major motions. We can see compared with (c), (a) has messy arrows on the lifting girl, (a) and (b) do not capture the optical flow of the spinning girl very well, and (d) has too many arrows on the background which should be relatively still.



(a) ws = 25



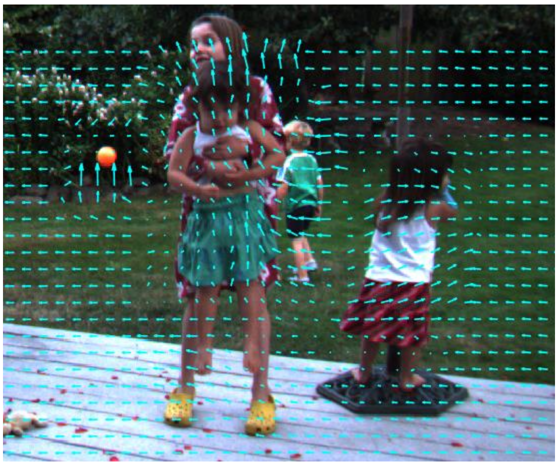
(b) ws = 45

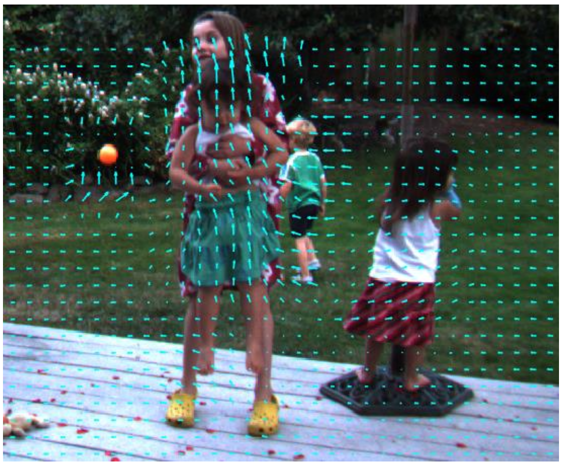


(c) ws = 65

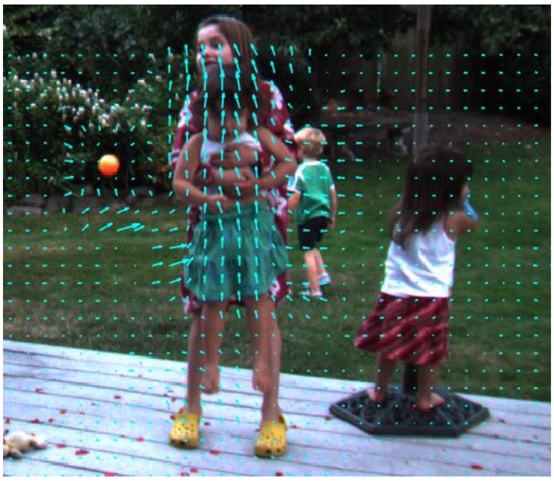
(d) ws = 105

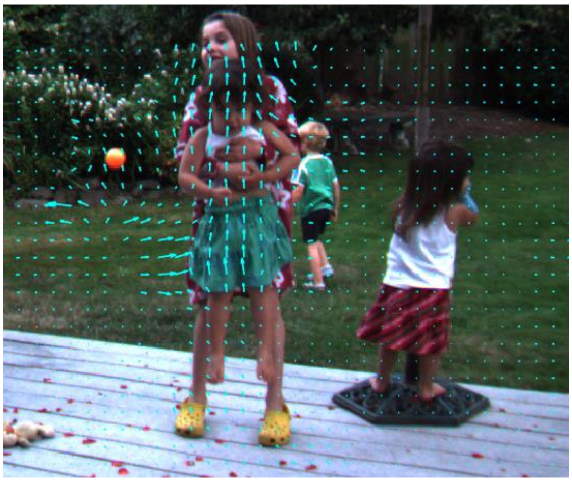
3. Smoothing factor of 2D Gaussian (sigma)
   Blurring the images with a Gaussian can make them smooth, then the intensity values are continuous and differentiable along x and y directions. Sigma decides how much the images are blurred, it is a trade off between response of noise and motion.
   For example, look at the motion fields got from frame12-13 of Backyard of different sigma, with window size being unchanged. Compared with (b), (a) is affected too much by the background noise, (c) and (d) do not capture the motion of the spinning girl well.





(a)sigma = 2 (b)sigma = 4





(c)sigma = 8 (d)sigma = 16

Algorithm Discussion

Lucas-Kanade algorithm is based on the assumption that shift h is small and varies slowly, that is the pre-condition of Taylor approximation. Therefore, this algorithm does not work well when the corresponding pixels are too far away, so that the second pixel is not inside the neighborhood of the first pixel where intensity varies linearly. In our experiments, our frame sequence is short so some objects move too fast between frames, like the small falling ball in ‘Backyard’.

Overall the algorithm can detect most optical flows accurately with proper parameters.