Assignment 4: Optical Flow

Practical Problem

Levin Yu (dy903)

1.1 Normal Flow

1) Choose toy2.png and toy3.png, smoothing them with gaussian_filter(), sigma 2.0.

2) For every pixel in toy2.png, calculate it's:

time derivative:

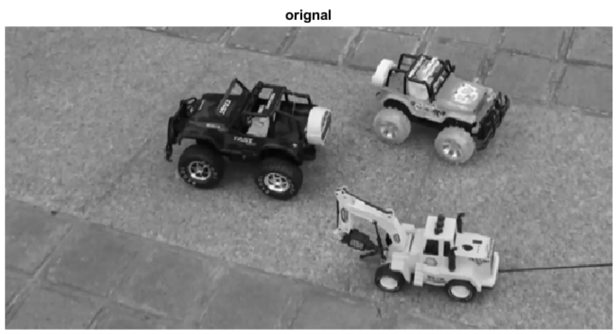
```
deriv_I_t(i, j) = filtered_toy3(i,j) - filtered_toy2(i,j);
```

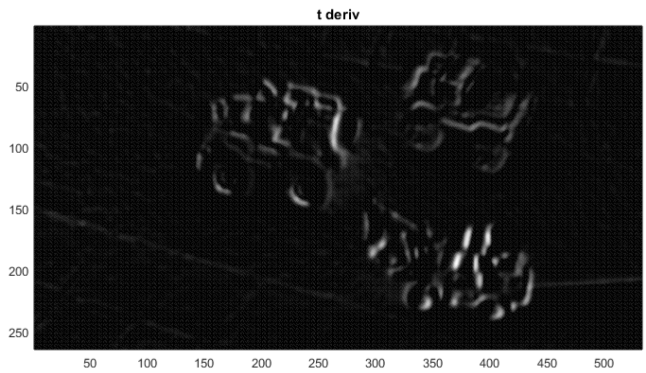
x derivative:

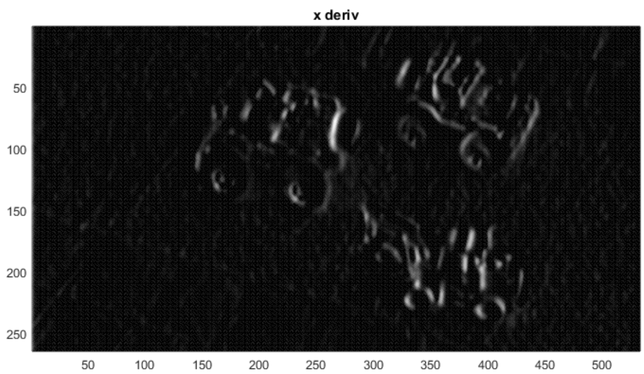
```
deriv_I_x(i, j) = filtered_toy2(i,j+1) - filtered_toy2(i,j);
```

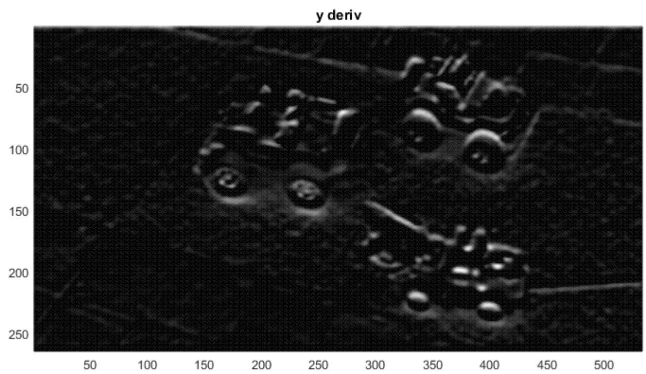
y derivative:

```
deriv_I_y(i, j) = filtered_toy2(i+1,j) - filtered_toy2(i,j);
```







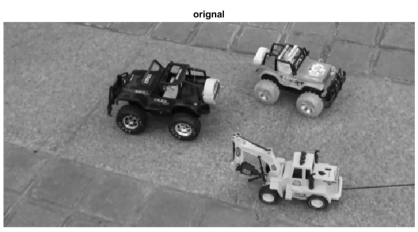


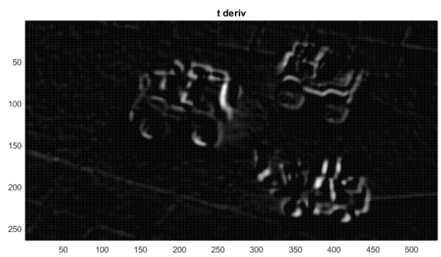
3) Calculate the normal flow at each pixel:

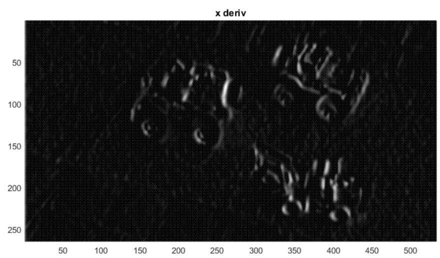
$$\boldsymbol{u} = -\frac{I_t \nabla I}{|\nabla I|^2}$$

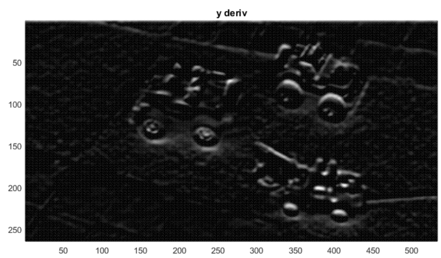
Result:

4) Apply the code to toy4.png and toy5.png









Normal Flow detected in most of pixels, but many of them are noises. The flows of objects we want to detect were submerged in the flows of back ground and noises. Maybe more noises are generated due to the aperture problem. My next step is try some algorithm and apply some threshold, trying to detect true flow with less noises.

1.2 Flow calculated over pixel neighborhood

1) Choose toy2.png and toy3.png, smoothing them with gaussian_filter(), sigma 2.0.

2) For every pixel in toy2.png, calculate its derivatives for x, y, and t.

3) For every pixel P (x, y), set a linear equation system with its adjacent 3 points: P(x+1, y), P(x, y+1), P(x+1, y+1).

$$\begin{bmatrix} I_x & I_y \\ I_{x+1} & I_y \\ I_x & I_{y+1} \\ I_{x+1} & I_{y+1} \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} I_{t(x,y)} \\ I_{t(x+1,y)} \\ I_{t(x,y+1)} \\ I_{t(x+1,y+1)} \end{bmatrix}$$

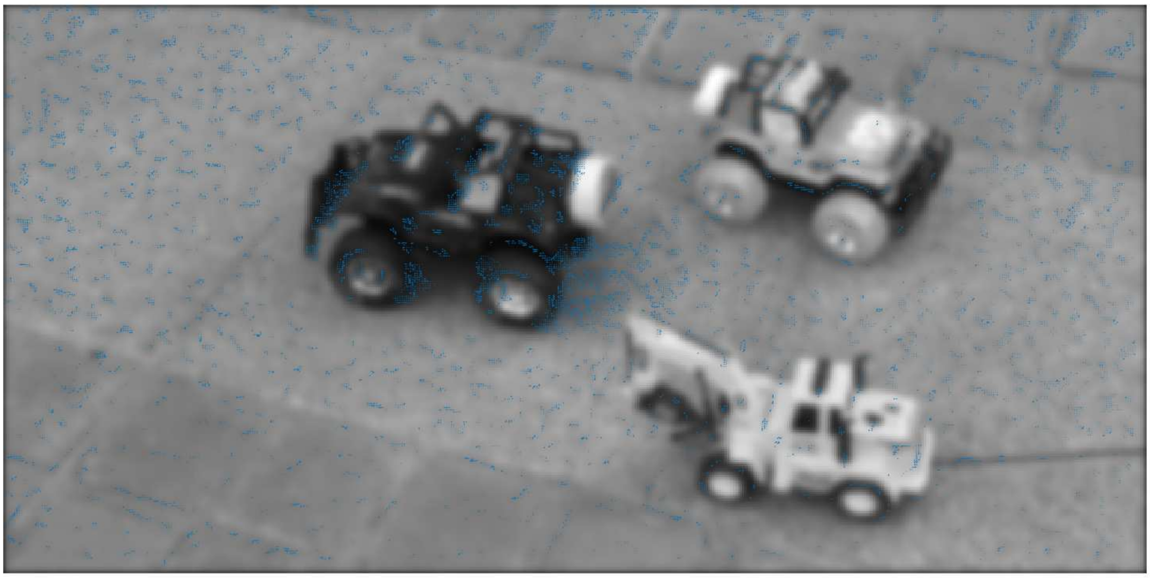
Solving each linear equation system with least square algorithm gives best solutions of flows in each pixel. Then set a threshold and remove all the point with a normal of flow vector greater than that threshold. (in my code, the threshold is 8)

Results:

Raw image:

Sigma 1.0:

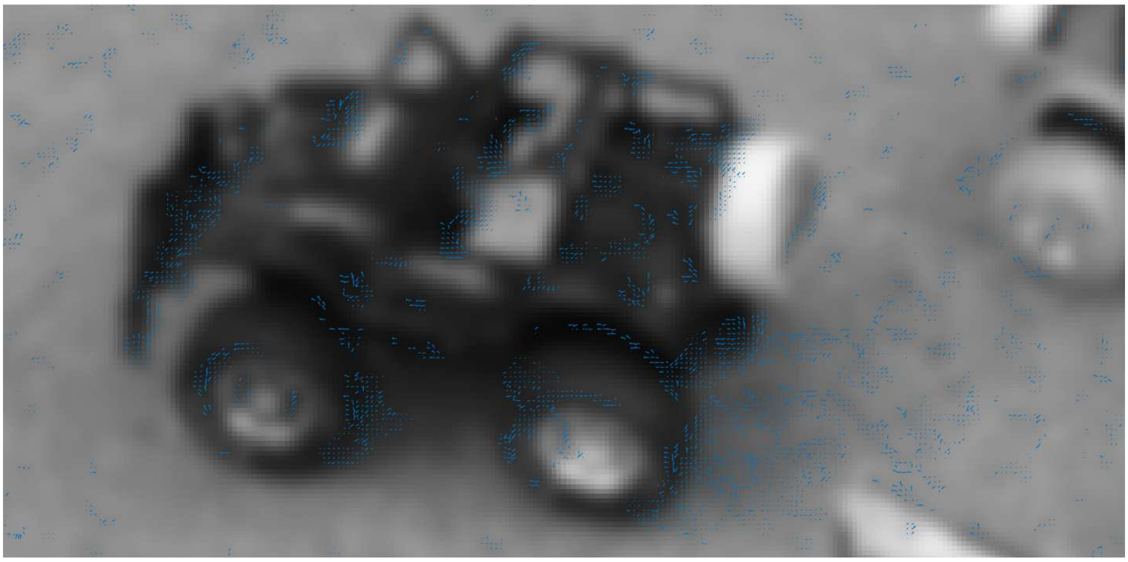
Sigma 2.0:



Flow calculated over pixel neighborhood shows better correctness of which direction are the objects real moving, compared with the normal flow. The application of my manual filter generated less noise and made it easier to focus on objects that were really moving.

With the increase of sigma, flows detected in the background were decreasing. Which made it easier to distinguish toy cars movement.

Here is some detail of flow vector with sigma 2.0:



The correctness of these vector is quite good and its more flow generated on the edge of the object, which is reasonable because these are the pixels which are changing. Also, we can see with the smoothing and my manual threshold, the noises in background is much less than normal flow.