# NYU CS-GY 6643, Computer Vision
# Spring 2017, Prof. Guido Gerig

## Assignment 4: Optical Flow

| | |
|---|---|
| Out: | Sun 04-16-2017 |
| Due: | Mon 05-01-2017 midnight (theoretical and practical parts, submission to NYUclasses) |
| TAs: | Rupanta Rwiteej Dutta |
| | Guangqi Chen |
| Office hours | TA: Tue 3-5pm Rupanta Dutta, Thu 2-4pm Guangqi Chen, MetroTech Center, 10th floor, |
| | Instructor Mo 3pm-5pm, 2 MetroTech Center, 10.094 |

Required Readings: Material on motion and optic flow (Horn: Tutorial)
**Slides to these chapters provided on course web-page, see also handwritten notes and tutorial by Shireen Y. Elhabian, all on the web.**

## II. Practical Problem (26pts)

### Grading

- Calculate and display spatial and temporal gradient image on smoothed images (4pts).
- Calculate normal flow with reasonable results (6pts).
- Calculate flow using 2x2 neighborhood, apply to raw and smoothed images with reasonable results (10pts).
- Report (coverage of all experiments, stle, clarify, organization, critical discussion) (6pts).

## 1 Optical Flow

Optical flow is a technique to determine motion in a series of image data. We are given a sequence of images showing a real life situation of a driver (soure R. Klette, Auckland). Our goal is to implement an optical flow method that can measure the motion of objects.
Implement an optical flow method to determine the motion between consecutive images. The methodology is based on the basic assumption of brightness constancy $\nabla E.v + \frac{\partial E}{\partial t} = 0$. Remember further that we cannot calculate a solution for the velocity vector at every pixel but need to include a pixel neighborhood. Note that this is a local estimate, where we can primarily only measure the normal flow (i.e. flow parallel to the image gradient and thus perpendicular to boundaries.).

### 1.1 Normal Flow

1. Choose two consecutive images from the video sequence.
2. Apply smoothing to the images (remember that optical flow assumes smooth object boundaries, i.e. boundaries with larger smoothness than the spatial shift). See note below regarding image smoothing.

toy2.png toy3.png toy4.png
toy5.png toy6.png toy7.png

Figure 1: Video sequence (courtesy of Anastasios Roussos, Queen Mary University of London, UK).

3. Calculate the temporal gradient image $\frac{\partial E}{\partial t}$ via the difference of the blurred versions of the two consecutive frames, i.e. simply by subtracting the two frames as I(x,t+1)-I(x,t).
4. Estimate the spatial derivatives $E_x = \frac{\partial E}{\partial x}$ and $E_y = \frac{\partial E}{\partial y}$ by calculating pixel differences I(x+1,y,t)-I(x,y,t) for $E_x$ and I(x,y+1,t)-I(x,y,t) for $E_y$.
5. Display the original image and the spatial and time gradients (see Fig. 2 for an example).
6. Calculate the normal flow at each pixel.
7. Display the resulting flow vectors as a 2D image. Overlay these vectors and the gray level image.
8. Eventually apply your program to another consecutive image pair of the 6 image video sequence and compare the results.
9. Look at the normal flow vectors of the toy cars and discuss your solution.

### 1.2 Flow calculated over pixel neighborhood

1. Apply image smoothing and differentiation as described above.
2. Choose 2x2 pixel neighborhoods for a local estimate of velocities using the following solution strategy (you may also use your own version of solving an equation system):

$$\nabla E.v + \frac{\partial E}{\partial t} = 0$$

now you have 4 measurements from 2x2 pixels to solve for v:

$$\begin{aligned} Av + b &= 0 \\ A^T A v &= -A^T b \\ v &= -(A^T A)^{-1} A^T b \\ C &= A^T A \end{aligned}$$

The columns of A are the x and y components of the gradient $\nabla E$ and b is a column vector of the t gradient component of E, $E_t$. The inverse can be calculated with Matlab's pinv, e.g., or the equation system can be solved by SVD.

This calculation must be performed at each (x,y) in the image with the columns of A and b extracted within a neighborhood of size 2x2 (or NxN if you want to extend the smoothness range). Please note that this calculation may give NaN at some pixel locations, so that you need to check for NaN and output a zero flow vector.

3. Apply this flow calculation to a) the raw images, b) images filtered by a Gaussian smoothing of sigma=1, and c) images filtered by a Gaussian smoothing of sigma=2. A Matlab Gaussian filter module will be provided with this assignment.

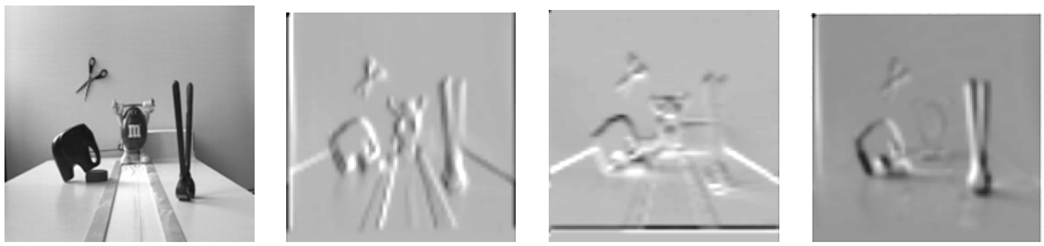

Figure 2: Example of derivatives used to solve optical flow. From left to right: Original image, derivative in x, derivative in y, derivative over time frames.

**Image smoothing:** We will provide a Matlab image filtering module based on a 2-D Gaussian filter.

**Image derivatives:** This is for non-imaging students: *After smoothing*, the simplest way to get the components of the image gradient is a [-1,1] filter that is applied horizontally and vertically, respectively. This is done by pixel differences I(x+1,y,t)-I(x,y,t) for $E_x$ and I(x,y+1,t)-I(x,y,t) for $E_y$ calculated on the smoothed images. Notice that you cannot filter the whole image but create a nonfiltered border of 1 pixel width. Please also note that the derivative operation creates a signed image, i.e. results that contain negative and positive values. You therefore need to choose the type “double” for all the differentiation and optical flow calculations. For display of images, we usually convert double back to integer and move the value range into a positive range with 0 in the middle of this range.

## 2 Bonus 1: Regularization: Horn and Schunck or Lucas and Kanade Method

This is a bonus question for students who would be willing to do extra work and want to earn extra credits.
Given the limitations of the local estimates above, you might implement the Horn and Schunck or Lucas and Kanade algorithms which provide a smoothing term to regularize the flow field (see details in slides Optimal-Flow-I and, the respective Horn-Schunck papers on the course web-site and slides and the tutorial by Shireen Y. Elhabian, all on the web.).

- Implement the Horn-Schunck or Lucas-Kanade algorithm.
- Compare the flow fields with the ones obtained with the simplified approach.

# 3 Bonus 2: Apply optical flow to your own video sequence pair of images

Be creative to capture a short videosequence of a moving object, extract a few images (if a video has a very high frequency you may not select consecutive images but a time sampling where images show a clearly visible object motion). Apply image filtering for smoothing and then calculate flow.

## Report

Write a report including your approach and results. You don't need to replicate the full theory as shown in the slides, but summarize the equations you use. Follow the instructions and calculate results. Important are discussions of your results, e.g. explanation what you see in the normal flow versus 2x2 calculation of the flow, and differences between the two. Compare the results obtained from the unfiltered versus Gaussian filtered images and discuss. Besides displaying the resulting images, you may also show a small zoomed patch of interest so that flow vectors can be better seen.