Summer Research Project
Video Based Mouse Seizure Detection

Juarez Sampaio

Rice University

July 1, 2015

Content

- A few concepts in image processing and computer vision:
  - Convolution
  - Image pyramid
- Using classification to tracking
- line wise machine learning approach to tracking
- Computing optical flow

Convolution: Definition

- Convolution is technique broadly use in image processing
- It consists of replacing the intensity value of every pixel by the result of an operation that involves a window around that pixel
- The operation is a sort of dot product between a kernel and the window.
- Corresponding elements of the kernel and then window are multiplied and the added.
- In other words:

$$I(i,j) = dot(A, I[i-k : i+k, j-l : j+l])), \forall i,j \qquad (1)$$

Convolution: Example 1

$$* \frac{1}{25}\begin{bmatrix} 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 \end{bmatrix} =$$

Figure : Example of convolution to blur the image

Convolution: Example 2

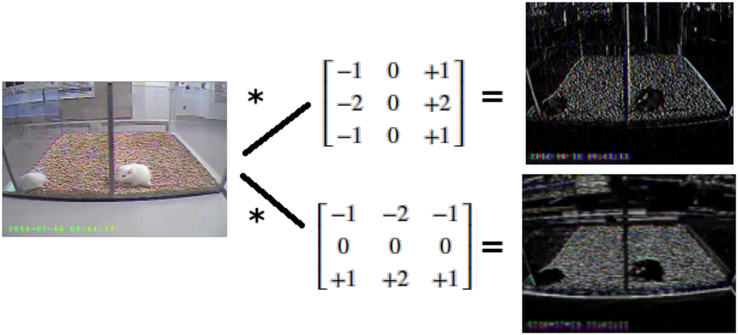

Figure : Example of convolution to compute derivatives. The upper filter computes x derivative while the bottom one computer y derivative. Both can be combined to find the gradient.

Image Pyramid: Introduction

- In many image processing tasks we want to work with different sizes of the same image.
- The concept is as follows: the base of the pyramid contains the highest resolution image available and as we go up the pyramid the resolution decreases.
- The literature actually sees this process as an inverted pyramid. Thus, the high resolution image stands at the top and to decrease resolution we actually go down the pyramid.

Image Pyramid: Introduction

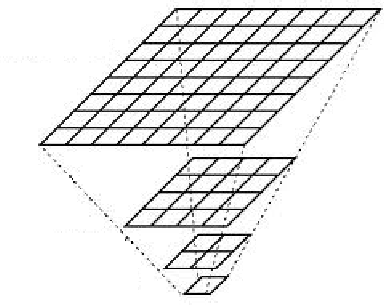

Figure : Inverted pyramid. [1]

[1] source: open cv documentation
http://docs.opencv.org/doc/tutorials/imgproc/pyramids/pyramids.html

Image Pyramid: Down sampling

Down sampling works as follows

- first we blur the image by convolving the image with a low pass kernel
- then we remove n-1 out of n rows and columns.
- The size of the blurring window depends on the degree n. Larger n implies larger blurring operation.
- As in signal processing, the low pass operation is required to avoid alias.

Image Pyramid: Down sampling

Figure : Down the pyramid

Image Pyramid: Up sampling

Up sampling works as follows

- we add n-1 zeros between every element of the image
- we convolve the resulting image with the same low pass kernel
- This results into a interpolation of the original values
- The previous description allowed us to down sample by integer ratios D and up sample by U. Combining these two operations we can obtain any rational U/D.
- Keep in mind that it is important to up sample first! Down sampling always implies loss of data!

Image Pyramid: Up sampling

Figure : Up the pyramid

Image Pyramid: Demo

Run pyramid demo

Classification for Tracking: Generalizing the convolution

- The convolution replaces the value of a pixel by the result of a dot product between a kernel and a window around the pixel.
- What if instead of this dot product, the value of the pixel were replaced by the result of a function applied to a window around the pixel?
- This is precisely the idea I applied last week.
- The function used is a classifier of some sort.

Classification for Tracking: Problem

- The problem with this was that the libraries we're working with are not familiarized with this extension of the convolution concept and provide no easy way of realizing this classification.
- The convolution had to be hand implemented and the process became slow.
- Also, given the dimensionality of the problem, the accuracy rate of 94% was not enough to allow this approach to reliably track the mouse

Line wise machine learning approach to tracking: idea

- The previous approach tried to use machine learning to answer the question 'is there a mouse in this square'?
- Now, we'll use machine learning to answer 'is there a mouse in this line'?
- Another classifier, trained in the same manner, will answer 'is there a mouse in this column'?
- Both classifiers will be solving a way smaller version of the problem. The row classifier will learn a function on space of dimensionality 320 and the column one on a space with d=240.
- As before, we can use image pyramids to reduce those dimensions to half and quarter of the full resolution!

Line wise machine learning approach to tracking: Details

- The function to be learn has to detect something of the sort 'is there a sequence of consecutive pixels of similar intensity?'
- The value of a pixel alone has no meaning, it is its connections that matter.
- Thus, there is a hidden feature to be learned here. This requires a classifier that is able to learn those hidden states.
- My bet is that neural networks are the proper tool to this task.
- If this approach works, the hidden nodes will learn the hidden properties and the output layer will combine them to produce the classification.

Collecting samples

Run collect demo

Optical Flow experiments

- Optical flow is computed on the tracking window produced my the current best tracker
- Because we've reduced the problem to a 30x30 window, we can run dense optical flow.
- Currently I am only plotting the highest value of magnitude seen in the flow.
- The idea is that we can detect seizures by looking at the data
- Maybe we could divide this data into time slices of a given length and then compute its frequency components.

Run collect demo