Universidad Autónoma de Guadalajara



MS in Computer Science

Image Processing

Created by: Luis Raul Villanueva Lopez

Guadalajara, Jalisco, August 2013.

Moving objects detection with Python and OpenCV

Raul Villanueva, UAG 2013

# Abstract:

Identifying moving objects is a critical task for many computer vision applications; it provides a classification of the pixels into either foreground or background.

On this paper we review a simple way of detecting moving objects using Python and OpenCV [1] which is a very popular image processing framework. We are not proposing anything new here, since most of the algorithms for motion detection are well known [2] and this is more an attempt for the people with almost no background in image processing to understand what image processing is with a simple application in mind.

The algorithm shown on this paper present acceptable results in good lighting situations (day light). We did not test under low lighting conditions so it remains something to try further.

# Introduction:

Today, there is plenty of ways of processing images for pattern recognition. Within that area of knowledge there exists also different way to recognize the movement of an object within a sequence of images. All of that exists already [2]. However, due to its nature and mathematical fundations, it turns out that not all people is able to really understand math or complex algorithms. On the other side, there are no low cost or free professional tools – ie: Matlab, Mathematica, etc. – that allows anyone to implement a simple object tracking algorithm using popular free programming languages such as Python.

Then this paper attempts to introduce basic image processing using Python as the primary language, and many popular libraries for Math and Image processing such as NumPy, OpenCV or SimpleCV among others.

It is not the intent of this paper to introduce any sort of new algorithms for image processing, or to discuss pit and falls of existing ones.

This paper will focus on identifying moving objects with the simple technique of background removal. Motion detection is a useful and effective function of surveilance camera system. In general, motion detection is implemented by the image segmentation method in industrial applications [3] [4] [5]

It will be shown how to perform such algorithm easily using python and OpenCV.

# Development Environment:

For this laboratory, the following hardware and software was utilized:

* A Standard webcam with resolution of 640x480 pixels was used to capture video for image processing. Although any kind of webcam will suffix for this experiment.
* Python(x, y) [3] Python distribution was used to simplify the install of NumPy and [1] libraries. As of this writing, version 2.7.5.0 was available.

# Methodology:

Being new to image processing is kind of difficult. There is a certain basic knowledge to have in order to start using image processing tools in order to do very basic image processing.

Image processing is at least divided in the following steps:

Image acquisition

Image Processing

Output Conditioning

## Image Acquisition

This is the processing of capturing either video or still images using an image sensor. This could be either a webcam, a photo camera, an infrared camera and so on. The final output of any image capture sensor should be in a standard image format such as JPEG, GIF for still images or MPEG or AVI formats for video images. In our case, we are using video since we are attempting to identify motion.

Within this process, there may exists an additional step in charge of image contiditioning in order to accent or diminish certain features in the image, but in this case we are not doing so, since it is out of our scope. Instead, we are using ideal lighting conditions to facilitate the process.

## Image Processing

This is by far the complex one of the whole process. This process may include filtering, decomposition, color transformation, morphological operations, etc. Here is where this article focus more on. Here is a simple diagram of what we will do:



Code for the above diagram is hosted at <http://git.io/CWQMbQ>.

# Results:

In order to test the algorithm, a video was captured from a second floor of a building pointing to an esplanade with plenty of day light so that it would be kind of ideal lighting for our purposes. Here is the original image:



The image above shows 3 people apparently moving downwards. By averaging a configurable set of previous images we can calculate what objects are moving. The result of this is shown in the next image:



From above, we can conclude that not only the 3 people are moving, but there is also another object moving in the right side of the screen. Which turns out to be very helpful as the algorithm is detecting moving objects that otherwise would not be detectable using our eyes (or not that easy).

# Conclusions:

Image processing can be difficult to understand from a Mathematical standpoint, but now a days there are plenty of frameworks that allow image processing quite easily. OpenCV seems to be very robust for image processing and for our example it exceeded our expectations. Python made the things easier as it is very good for prototyping image processing that would then end up in faster programming languages such C or C++.

Moving objects detection was possible using a simple algorithm and it is a nice example for applications such as surveillance and monitoring of public places.

# References

|  |  |
| --- | --- |
| [1] | "OpenCV," [Online]. Available: http://opencv.org/. |
| [2] | K. M. E.-S. a. S. H. A. Shireen Y. Elhabian, "Moving Object Detection in Spatial Domain using Background Removal - State-of-the-Art," Information Technology Department, Faculty of Computers and Information, Cairo University, Cairo, 2008. |
| [3] | R. E. W. Rafael C. Gonzalez, Digital Image Processing, Addison-Wesley Publishing Company, Inc, 1992. |
| [4] | P. A. Michal Irani, "A Unified Approach to Moving Object Detection in 2D and 3D scenes," in *Proceedings of the 13th International Conference on Pattern Recognition, vol. 1, p.712-717*, 1996. |
| [5] | L. K. Chris Cho-pin Li, "A New Approach to the Detection of Moving Objects," *Information Sciences,* vol. 103, pp. 115-134, 1997. |
| [6] | "Python(x,y)," Python(x,y), [Online]. Available: https://code.google.com/p/pythonxy/. |
| [7] | "NumPy," [Online]. Available: http://www.numpy.org. |