

# Estimation of Speed and Acceleration Using Computer Vision in Python

Dr. D. Priour / Joseph Shives

## Abstract:

The goal is create a fairly simple program in Python to estimate the speed and acceleration of a moving object in a video. This was done using the default Python libraries as well as OpenCV. OpenCV is a free, open-source computer vision library, containing many pre-written commands for image processing. The general algorithm for this program is to read a video file frame-by-frame and analyze the frames with image processing tools to identify moving objects. Then, the speed of the object is estimated using the geometric centroid as the reference point. The speed of the object is calculated as the distance moved between measurements, in pixels, divided by the time interval between measurements, which is arbitrary. Acceleration is calculated as the change in speed divided by the length of the same time interval. A frequency of 50 per second was chosen as the frequency with which to take measurements.

## Prerequisite Information:

OpenCV is an open-source Python library containing many functions and methods related to image-processing. The subdiscipline of image detection and processing is called "computer vision." (This is what CV stands for in OpenCV.)

Average speed between two points is defined to be the distance between the points divided by the time taken to travel between them. One can calculate instantaneous speed by looking at the limiting value as the time between measurements becomes small. In calculus, this is called the "derivative," and the act of calculating it is called "differentiation." Acceleration is change in speed over time, hence, acceleration is the derivative of velocity with respect to time.

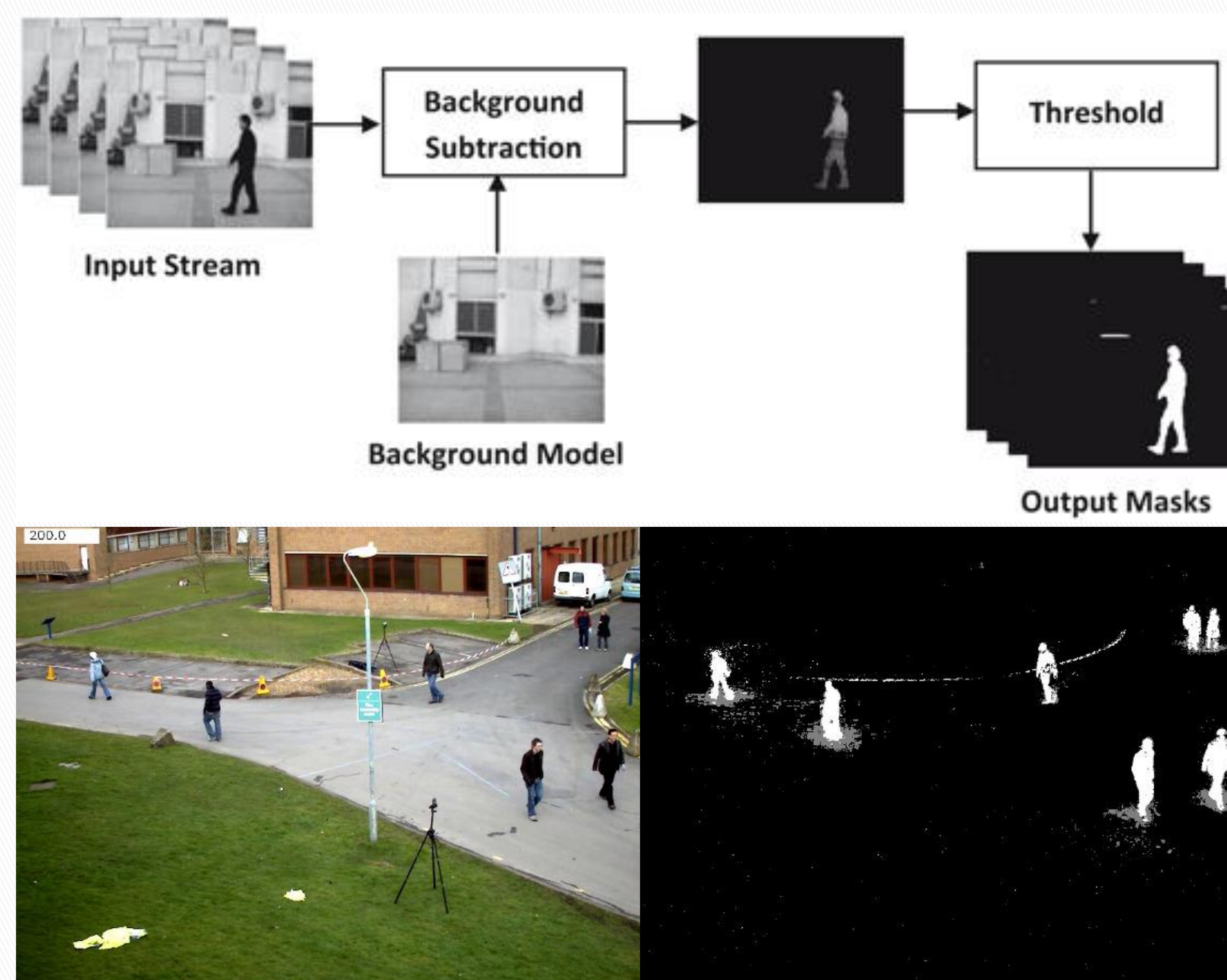
## Background Model and

### Subtraction:

Some frames are selected randomly, and the median values of each of the pixels is used as the "background" against which the video is compared. Each frame is read one at a time and the background is subtracted pixelwise from the frame, which highlights regions that are different from the background.

### Thresholding:

Difference values are compared to a predefined threshold; those below are set to 0 (black), and above to 255 (white). This makes calculation of the centroid easier, and helps to negate random noise.



## Mathematics:

$$\begin{aligned} \frac{d^2y}{dx^2} &= \lim_{h \rightarrow 0} \frac{\frac{dy}{dx}(x+h) - \frac{dy}{dx}(x)}{h} \\ &= \lim_{h \rightarrow 0} \frac{\frac{f(x+2h) - f(x+h)}{h} - \frac{f(x+h) - f(x)}{h}}{h} \\ &= \lim_{h \rightarrow 0} \frac{(f(x+2h) - f(x+h)) - (f(x+h) - f(x))}{h^2} \\ &= \lim_{h \rightarrow 0} \frac{f(x+2h) - 2f(x+h) + f(x)}{h^2} \end{aligned}$$

Conjecture:  $\frac{d^n y}{dx^n} = \lim_{h \rightarrow 0} \sum_{i=0}^n \frac{(-1)^i \binom{n}{i} f(x+ih)}{h^n}$  ?

$$\begin{aligned} \vec{x} &= \begin{pmatrix} x \\ y \end{pmatrix} \rightarrow d\vec{x} = \begin{pmatrix} x_i - x_{i-1} \\ y_i - y_{i-1} \end{pmatrix} \\ \rightarrow \|d\vec{x}\| &= \sqrt{(x_i - x_{i-1})^2 + (y_i - y_{i-1})^2} \rightarrow v = \left\| \frac{d\vec{x}}{dt} \right\| \approx \frac{\|d\vec{x}\|}{dt} \\ \rightarrow a &= \frac{dv}{dt} = \frac{d^2x}{dt^2} \approx \frac{d\|d\vec{x}\|}{(dt)^2} \end{aligned}$$

## Kinematic Calculations:

The geometric centroid can be derived from contiguous areas of white pixels. The centroid is used as the reference point. The distance the centroid moves between measurements is calculated, and the definition of the derivative is used to calculate speed and acceleration.

