

🏠 (/) / Tutorials (/tutorials/) / Computer Vision (/category/computer-vision/) / Image Moments (/tutorials/image-moments/)



Like 3.5K people like this. [Sign Up](#) to see what your friends like.

Image Moments

An Image moment is a number calculated using a certain formula. Understand what that formula means might be hard at first. In fact, I got a lot of questions about moments from the tracking tutorial (/tutorials/tracking-colored-objects-opencv/) I did long back. So, here it is - an explanation of what moments area!

The math of moments

In pure math, the n^{th} order moment about the point c is defined as:

$$\mu_n = \int_{-\infty}^{+\infty} (x - c)^n f(x) dx$$

This definition holds for a function that has just one independent variable. We're interested in images - they have two dimensions. So we need two independent variables. So the formula becomes:

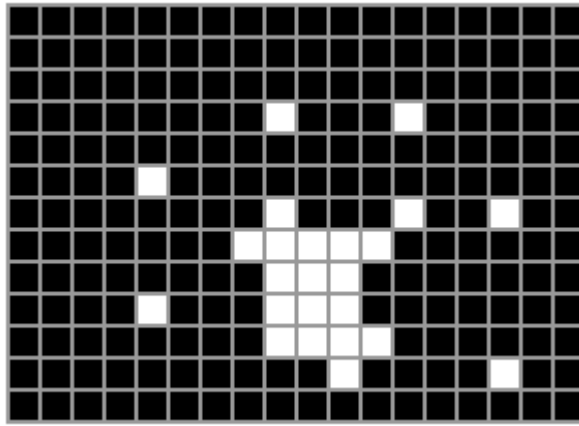
$$\mu_{m,n} = \int \int (x - c_x)^m (y - c_y)^n f(x, y) dy dx$$

Here, the $f(x, y)$ is the actual image and is assumed to be continuous. For our purposes, we need a discrete way (think pixels) to describe moments:

$$\mu_{m,n} = \sum_{x=0}^{\infty} \sum_{y=0}^{\infty} (x - c_x)^m (y - c_y)^n f(x, y)$$

The integrals has been replaced by summations. The order of the moment is $m + n$. Usually, we calculate the moments about $(0, 0)$. So you can simply ignore the constants c_x and c_y .

Now with the math part out of the way, let's have a look at what you can calculate with this thing.



A binary image with white and black pixels

Calculating area

To calculate the area of a binary image, you need to calculate its zeroth moment:

$$\mu_{0,0} = \sum_{x=0}^w \sum_{y=0}^h x^0 y^0 f(x, y)$$

The x^0 and y^0 don't have any effect and can be removed.

$$\mu_{0,0} = \sum_{x=0}^w \sum_{y=0}^h f(x, y)$$

Now, in a binary image, a pixel is either 0 or 1. So for every white pixel, a '1' is added to the moment - effectively calculating the area of the binary image! Another thing to note is that there is only one zeroth order moment.

Centroid

To calculate the centroid of a binary image you need to calculate two coordinates -

$$centroid = \left(\frac{\mu_{1,0}}{\mu_{0,0}}, \frac{\mu_{0,1}}{\mu_{0,0}} \right)$$

How did I get that? Here's a quick explanation. Consider the first moment:

$$sum_x = \sum \sum x f(x, y)$$

The two summations are like a *for* loop. The x coordinate of all white pixels (where $f(x, y) = 1$) is added up.

Similarly, we can calculate the sum of y coordinates of all white pixels:

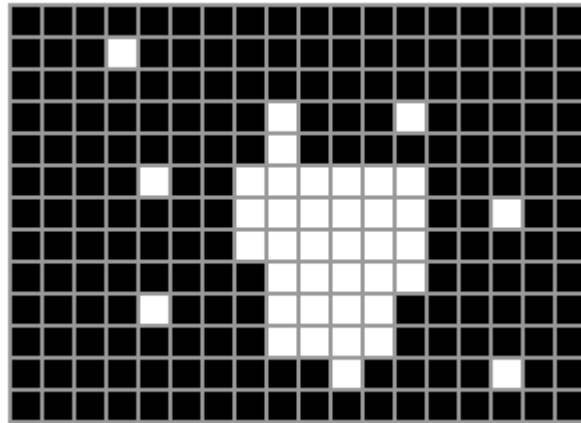
$$sum_y = \sum \sum y f(x, y)$$

Now we have the sum of several pixels' x and y coordinates. To get the average, you need to divide each by the number of pixels. The number of pixels is the area of the image - the zeroth moment. So you get:

$$\mu_{1,0} = \frac{sum_x}{\mu_{0,0}} \text{ and } \mu_{0,1} = \frac{sum_y}{\mu_{0,0}}$$

One interesting thing about this technique is that it is not very sensitive to noise. The centroid might move a little bit but not much.

Also, from the math it's clear this technique holds only for single blobs. If you have two white blobs in your image, the centroid will be somewhere in between. You'll have to extract each blob separately to get their centroids.



Calculating the centroid of this blob

Central moments

In fact, this kind of division is very common - dividing a moment by the zeroth order moment. It's so common that it has a name of its own - central moments.

So to calculate the centroid, you need to calculate the first order central moments.

Higher order moments

Going onto higher order moments, things get complicated really fast. You have three 2nd order moments, four 3rd order moments, etc. You can combine several of these moments so that they are translation invariant, scale invariant and even rotation invariant.

While reading about moments, I found an entire book dedicated to pattern recognition with moments (http://www.amazon.com/gp/product/0470699876/ref=as_li_ss_tl?ie=UTF8&tag=aish04-20&linkCode=as2&camp=217145&creative=399373&creativeASIN=0470699876). In fact, there are terms called *skewness* and *kurtosis*. These refer to third and fourth order moments. They measure how skewed an image is and whether an image is tall and thin or short and fat. Clearly, there's a LOT that can be learned about these mathematical tools.

[Like](#) 3.5K people like this. [Sign Up](#) to see what your friends like.