Research Homework 5

1st Ericka Céspedes Moya
Tecnológico de Costa Rica
San José, Costa Rica
ericka.cespedes@gmail.com

2nd Esteban Alonso González Matamoros
Tecnológico de Costa Rica
San José, Costa Rica
esteb.gonza29@gmail.com

***Abstract*—The following document contains the first and second part of the Research Homework 5 for the course of Introduction to Pattern Recognition. This homework required the students to read and analyse an article on the application of Hu's Invariant Moments, and to do the respective test sequence.**

***Index Terms*—image processing, pattern, recognition, Python, OpenCV, Hu, moments**

I. Analysis

This article discusses a descriptor focused on the shape contained in an image that works by calculating *moments*. A moment is a quantity, a figure that captures basic information about the shape of an object in an image, such as its area, the coordinates of its centroid, its orientation, among others. It is a statistical expectation of a random variable; that is, a value that the variable is expected to take in the future.

Ming-Kuei Hu postulated seven values (moments) that can be used to synthesize the shape of an object in an image. In the article *Visual Pattern Recognition by Moment Invariants*, he states that these values are not affected by changes in an image's rotation, translation, scaling, and reflection. The first moment of a random variable is the mean while the second moment is the variance. There are more moments, but their use is less frequent.

The first six moments have been proved to be invariant to translation, scale, and rotation, and reflection while the seventh moment's sign changes for image reflection. The fact that these values are not affected by changes in an image is not as robust, since it is highly dependent on the initial calculation of the centroid of the figure. If this preliminary calculation fails, successive moments will not be accurate.

To calculate the seven moments of Hu, one can work with the binary, segmented image of the object of interest, or with its contour. The first option is preferable the vast majority of the time [1].

The article on the application of Hu's Invariant Moments chosen for this homework was *Analysis of Hu's Moment Invariants on Image Scaling and Rotation*.

A. Formulae

If a single channel binary image I is considered. The pixel intensity at location (x,y) is given by I(x,y) that can take a value of 0 or 1.

1) Simplest Moment: The simplest kind of moment is the following:

$$M = \sum_x \sum_y I(x, y) \tag{1}$$

The sum of all pixel intensities is calculated. All pixel intensities are weighted only based on their intensity, but not based on their location in the image. For a binary image, the number of white pixels is calculated.

2) Raw Moments: More complex moments are referred to as raw moments to distinguish them from central moments.

$$M_{ij} = \sum_x \sum_y x^i y^i I(x, y) \tag{2}$$

where i and j are integers (e.g. 0, 1, 2, ...). The raw moments depend on the intensity of pixels and their location in the image, so intuitively these moments are capturing some notion of shape.

3) Centroid: The centroid of a binary blob is its center of mass. The centroid ($\bar{x}$, $\bar{y}$) is calculated using the following formula:

$$\bar{x} = \frac{m_{10}}{m_{00}}, \bar{y} = \frac{m_{01}}{m_{00}} \tag{3}$$

4) Central Moments: Central moments are very similar to the raw image moments, except that we subtract off the centroid from the x and y in the moment formula. These moments are translation invariant. No matter where the blob is in the image, if the shape is the same, the moments will be the same.

$$M_{ij} = \sum_x \sum_y (x - \bar{x})^i (y - \bar{y})^i I(x, y) \tag{4}$$

On the other hand, there are moments invariant to scale: normalized central moments.

$$\eta_{ij} = \frac{\mu_{i,j}}{\mu_{00}^{(i+j)/2+1}} \tag{5}$$

5) Hu's Seven Moment Invariants: Central moments are translation invariant, but that is not enough for shape matching. To calculate moments that are invariant to translation, scale, and rotation, one must calculate Hu Moments. [2]

$$\phi_1 = \eta_{20} + \eta_{02}$$
$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$
$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \mu_{03})^2$$
$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \mu_{03})^2$$
$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$
$$\phi_6 = (\eta_{20} - \eta_{02})[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$
$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2] - (\eta_{30} - 3\eta_{12})(\eta_{21} + \eta_{03})[(3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2]$$

Fig. 1. Hu's Seven Moment Invariants

II. Results

For this section, we decided to try some of the tests done in [2]. The image used for this tests was 2.

Fig. 2. Image used in tests

A. Only scaling test

The first test was done using an image and only scaling it from a resolution of 60x60 to 1600x1600, first we obtained the image in gray scale, then we did threshing to the image, we scaled it to the resolution that was used at that moment, then, we extracted the different Hu Moments of all the different images and used the Log Transform function on them based on [3], using: $H_i = -sign(h_i)log \mid h_i \mid$ to make them easier to represent and compare, then we proceeded to compare them based on only scaling of an image. The idea was to compare the Hu Moments of all the images and see if there were any representative of constant changes in them. This test was only to try the moments and hu moments functions and how well they behave.

For the coding of this test we used the Opencv functions resize, moments and HuMoments, getRotationMatrix2D, warpAffine, the Numpy functions mean and array, and the Math function copysign and log10.

B. Scaling and rotation

The second test done, was one were we had and image in gray scale, did threshold to it, and then proceeded to do 10 scales, we started with 60x60 and went up to 330x330 in augments of 30 pixels each, then, for each of these resolutions we had 360 images, which were done based with a rotation from 1°to 360°in augments of 1°. [2] Then we obtained the Hu Moments for every each one of this images and get the fluctuation for each hu moment at a given resolution.

For the coding of this test we used the Opencv functions resize, moments and HuMoments.

The function for obtaining the fluctuation in a group of data is:

$Fluctuation = \frac{max(x) - min(x)}{|mean(x)|} * 100$ [2]

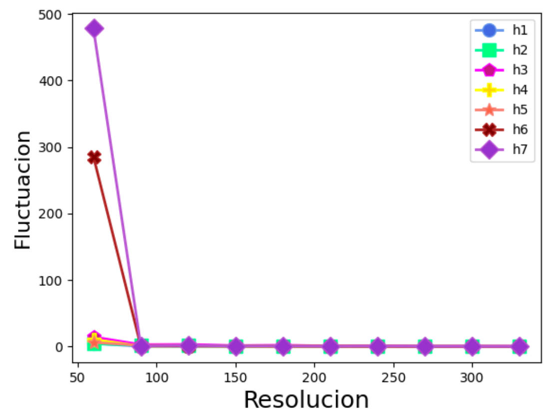

Fig. 3. Fluctuation of Hu Moments According to Image Resolution

III. Conclusions

Hu moments are used regularly to characterize the shape of an object. All 7 Hu Moments are invariant under translations (whether moved x or y axis), scale and rotation. If one shape is the mirror image of the other, the seventh Hu Moment flips in sign. Therefore, Hu Moments can be used to match shapes. If the distance is small, the shapes are close in appearance and if the distance is large, the shapes are farther apart in appearance.

The main advantages are its low dimensionality (only 7 values), the speed with which the vectors are computed and the fact that it is not necessary to scale the images to compare them.

The OpenCV functions moments and HuMoments are perfectly useful for getting the Hu Moments of an image. The best way to better understand, format and represent the Hu Moments obtained in OpenCV's HuMoments is using the Log Format because it converts the enormous numbers from decimal to maximum thousands.

Hu Moments are totally invariant in escalation, using both amplification and diminution, this is based on the results of the first test done in this investigation, the highest difference between a 7th Hu Moment of an image and the same image re-scaled is ≈ 0.001.

The tests and material in the Z. Huang and J. Leng paper [2] is true, based on the second test and it's results seen in Fig 3, we can observe that the fluctuation does became more variable on the lower the image is and it becomes more constant as it's resolution gets bigger, also we can conclude that most of these high variance in fluctuations stop at the resolution of 150x150. Also, same as said in [2] the Hu Moment with the highest fluctuation is the number 7 (Seen as h7 in the Figure 3), then we can observe that the lower the Hu Moments get, the lower their fluctuation gets.

A.

REFERENCES

[1] M.-K. Hu, "Visual pattern recognition by moment invariants," *IRE Transactions on Information Theory*, pp. 179–187, Feb 1962.

[2] Z. Huang and J. Leng, "Analysis of hu's moment invariants on image scaling and rotation," *2010 2nd International Conference on Computer Engineering and Technology*, vol. 7, May 2010. [Online]. Available: https://www.researchgate.net/publication/224146066

[3] S. Mallick and K. Bapat, "Shape matching using hu moments (c++/python)," *LearnOpenCV*, Dec 2018. [Online]. Available: https://learnopencv.com/shape-matching-using-hu-moments-c-python/