Artur ZACNIEWSKI

# USING ZERNIKE MOMENTS IN THE PROCESS OF AUTOMATIC IDENTIFICATION

*In this article, the problem of automatic identification with Zernike moments is presented. Due to their orthogonal properties they store image information with minimal redundancy and are rotational invariant, what make them accurate tool in systems of automatic identification of images. Content Based Image Retrieval system is presented, and two phases (offline and online) of this system are described in detail. Practical experiments are conducted on two datasets – standard MPEG-7 dataset and customized sprite dataset, showing promising results.*

## INTRODUCTION

Content Based Image Retrieval (CBIR) is a technique that allows a user to extract an image based on a query, from a database containing a large amount of images. One of the most important issues in designing a CBIR system is to select the image features that best represents the image contents in a database. CBIR could be seen as a process of finding images similar in visual content to a given query from an image database. It is usually performed based on a comparison of low level features (image information), such as colour, texture and shape features, and their combinations, extracted from the images themselves. CBIR system extracts features that are used to retrieve relevant images from image database that best match with query image [2][4].

Shape plays an important role in describing image contents and for CBIR purpose, a shape representation should be robust, invariant, and easy to derive and match. In describing a meaningful shape representation, a common method is to use a moment descriptor [2]. The feature of image moments has been widely used in the areas of computer vision such as object identification techniques [7] or shape classification [6].

Moment functions capture global features and thus are suitable for shape recognition. Some moment functions exhibit natural invariance properties including invariance to translation, rotation or scaling. There are various examples of moments including geometric, complex, radial and orthogonal. Geometric moments are widely used in image processing, however these moments are not optimal with respect to information redundancy. Hu [3] was the first who introduced seven moment-based image invariants from geometric moments that are invariants to rotation [6].

In order to make the features invariant to translation and scaling, it needs to substitute the geometric moment with the normalized central moment. These moment values are invariant to translation, rotation and scaling. Unfortunately, the computation of higher order moment invariants, is a quite complicated process. To overcome the shortcomings associated with geometric moments, Teague [14] suggested the use of orthogonal moments.

Orthogonal moments can be categorized into discrete and continuous orthogonal moments. The most well-known orthogonal moment families include Zernike [5][14], Pseudo-Zernike [15], Fourier-Mellin [1], Legendre [10], Tchebichef [9] and Krawtchouk [17] that were applied in most image processing applications. Moments with a continuous orthogonal base set such as Legendre [10], Zernike Moments (ZM) and Pseudo-Zernike can be used to represent an image with minimum redundancy information.

## 1. MATERIALS AND METHODS

### 1.1. Zernike Moments

The benefits of the orthogonal moments are the unique description of an object with low information redundancy and their ability to perfectly reconstruct an image. ZM are able to store the image information with minimal information redundancy and have the property of being rotational invariant [2]. In [13] it was shown, that ZM are more accurate, flexible, and easier to reconstruct than Hu moments. The accuracy of Zernike moments was achieved by increasing the order of the moments.

The magnitude of ZM's are rotationally invariant, which is crucial for certain image processing applications, such as classifying shapes that are not aligned. The orthogonal properties of ZM's suits them better for shape recognition applications because unlike geometric moments their invariants can be calculated independently to arbitrary high orders without having to recalculate low order invariants. These orthogonal properties also allow one to evaluate up to what order to calculate these moments to get a good descriptor for a given database of images. As a feature, Zernike moments are constructed using a set of complex polynomials and are defined inside the unit circle and the radial polynomial vector. [16].

Zernike Polynomial (ZP) in polar coordinates is defined as in equation (1):

$$V_{nm}(\rho,\theta) = R_{nm}(\rho)exp(-jq\theta) \quad (1)$$

where:
$R_{nm}$ - radial polynomial with order $n$ and repetition $m$,
$n$: positive integer or zero; i.e. n = 0, 1, 2, ..., ∞,
$m$: positive integer subject to constraint $|m| <= n$, and $(n - |m|)/2 = 0$,
$\rho$: length of vector from origin to (x, y) pixel, i.e. $\rho = \sqrt{x^2 + y^2}$ (radius),
$\theta$ : angle between the vector $\rho$ and the $x$ axis in the counter clockwise direction.

$V_{nm}(\rho, \theta)$ forms a complete orthogonal set over the interior of the unit disc of $x^2 + y^2 \leq 1$. Complex Zernike moment is defined as in equation (2):

$$A_{nm} = \frac{n+1}{\pi} \sum_{\rho} \sum_{\theta} [V_{nm}]^* I(\rho, \theta), \text{ such that } \rho \leq 1. \quad (2)$$

where: $I(\rho, \theta)$ - image function [2][16].

The architecture of CBIR system, which uses ZM for image retrieval is shown in Fig. 1.

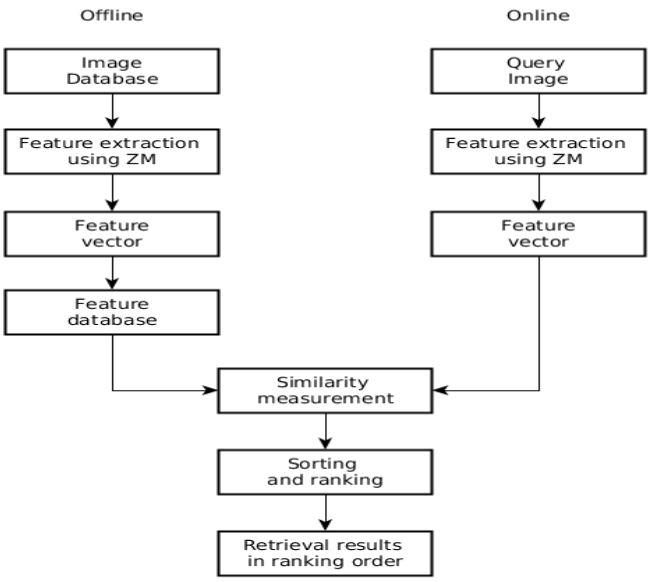


***Fig. 1.*** *Architecture of CBIR system using Zernike Moments [2]*

In offline phase features are extracted from image database and then are saved to feature database. In online phase one can select image query from image database, then both features are compared [2].

To improve the performance of retrieving the relevant images from the image database, e.g. Support Vector Machine (SVM) classifier can be used to classify the image database [4]. Similarity measurement can be performed using Euclidian Distance (ED) as shown in equation (4):

$$ED(Q, V^{(k)}) = \sqrt{\sum_{i=1}^{n} (q_i - v_i^k)} \quad (3)$$

where:

$Q = \{q_1, q_2, \ldots, q_n\}$ - feature vector corresponding to query image,

$V = \{v_1^{(k)}, v_2^{(k)}, \ldots, v_n^{(k)}\}$ - feature vector corresponding to image $k$ in the image database.

ZM's are orthogonal that they will be made easy to use for reconstructions. However, the fact that they are orthogonal are not their most important features. Any orthogonal polynomial could be used as a basis function from which easy reconstructions could be made. Rotational invariance is achieved by computing the magnitudes of the ZMs. The rotation of an image is easily expressed in polar coordinates since it is a simple change of angle [16].

### 1.2. MPEG-7

The binary images used are the MPEG-7 Core Experiment CE Shape-1 Part B dataset that was obtained from [8] and created by the Motion Picture Expert Group (MPEG) committee. It contains 1400 binary images grouped into 70 categories. There are 20 images within every category called for example tree-1.gif, tree-2.gif, …, tree-20.gif. Figure 2 presents six examples of trees from MPEG-7 dataset – tree-1 to tree-5 and tree-16 to tree-20. Every image has 256 x 256 pixels.


***Fig. 2.*** *First three and last three images of trees from MPEG-7 dataset [8]*

## 2. EXPERIMENTS

Experiments of ZM implementation that were carried out by using both color and binary image datasets are presented. The experiments were implemented under Ubuntu 16.04 operating system using OpenCV 3.2.0 library, on Intel i5 CPU 2.90 GHz with 8 GB RAM. The experiments were conducted using the ZM method in testing the capability in retrieving similar look images from the colored sprites database [11] and the binary MPEG-7 image database [8]. Below, the tracking approaches belonging to these two subcategories are presented.

When utilizing ZM for shape description, one should be careful with translation and scaling of the object in the image. Depending on where the object is translated in the image, ZM could be significantly different. Similarly, depending how object is scaled (how small or how big) in the image, ZM won't be identical. Fortunately, as it was mentioned earlier, magnitudes of the ZM are independent of the rotation of the object, which is desired property when working with shape descriptors [12].

### 2.1. Offline phase

In offline phase, to find outline of the image, the image should be converted to grey scale, all pixels should be inverted, and threshold should be applied. The largest contour is an outline of image, and will be used to compute the ZM. Results of these operations are presented on figure 3.

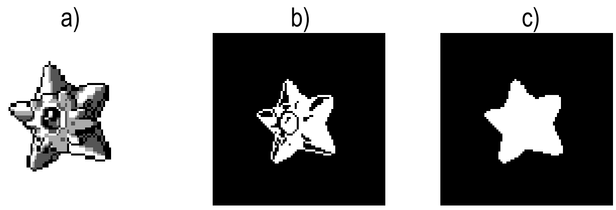


***Fig. 3.*** *Consecutive phases of preparing image for computing ZM:*
a) original image, b) thresholded image, c) outline

This particular image comes from colored images dataset [11], which consists of 151 images, and every image is a 56 x 56 pixels high quality sprite and it is single class in the sense of classification. For image from figure 2, ZM are presented on figure 3. These are the moments up to order 8 from $Z_{00}$ to $Z_{88}$ as it was shown in equations (1) and (2).

[0.31830989, 0.00422651, 0.25367941, 0.00236395, 0.00658118, 0.01656146, 0.04836281, 0.00810191, 0.07239312, 0.00798573, 0.00138033, 0.15577706, 0.0730983, 0.00098222, 0.04974908, 0.06654034, 0.00608102, 0.00333253, 0.12918171, 0.01854248, 0.00843612, 0.00845771, 0.00953795, 0.05887543, 0.02508559]

***Fig. 3.*** *Zernike Moments up to order 8 for 56 x 56 pixels image*

The result from figure 3 means that every image from this database is quantified with 25 floating point values (feature vector), and every value is a moment of particular order [2]. In offline phase, ZM for every image from image database are computed and stored in feature database, as it was shown on figure 1. What is important, when polar coordinates are used to calculate the ZM, only value of radius $\rho$ from equation (1) is necessary to declare, what makes computations very efficient [12].

### 2.2. Online phase

In online phase (real-life scenario), objects in images very often are rotated, translated and scaled. To avoid descriptors with different values based on the translation and scaling, segmentation should be performed first. In this process the foreground (object) should be distinguished from the background (part of the image that is not needed to be described). After segmentation bounding box is formed around the object to crop it out to obtain translation invariance. Then the cropped object is being resized to constant dimensions to obtain scale invariance [12]. After aforementioned normalization stage, ZM can be applied to images to characterize shape of the object.

On figure 4a, image with game console is presented. One of the sprites is displayed on the screen of device.

a) b) c)

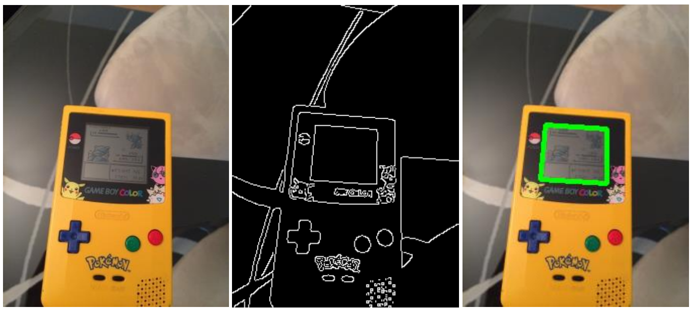

*Fig. 4. Detecting screen of the device: a) original image, b) image after bilateral filtering and edge detection, c) detected screen*

This object is scaled and translated and should be separated from the background. First, to find screen of the console bilateral filtering (to remove noise) and edge detection could be used. Result of this operations is shown on figure 4b .

After finding contours, it is necessary to find exactly this particular with screen of the device. It could be performed by analysis of the sight of device - screen has four sides and four vertices, and that kind of contour should be searched for [12]. Found contour is presented on figure 4c, and because it is skewed and also perspective is wrong, perspective transformation should be applied. Result of this operation is presented on figure 5a.

a) b)

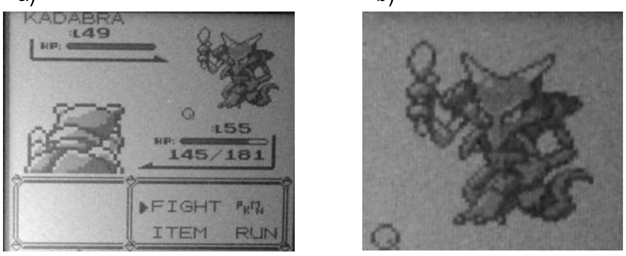


*Fig. 5. Transformations of the image from screen: a) after perspective transformation, b) after cropping*

Further thing to do is to crop sprite from the screen after perspective transformation. It is done empirically by trial and error, because it depends on particular situation (object location on screen). Result of cropping is presented on figure 5b.

ROI (Region Of Interested) was extracted and the feature vector (ZM) for query image should be computed. Query features are compared with pre-computed features from a database by measuring ED, as it was shown in equation (4). The smallest ED gives an information about searched object.

MPEG-7 dataset is ready to compute feature vectors for offline database, no pre-processing of images is necessary. The selected retrieval ranking results for MPEG-7 image dataset are shown in Table 1. Two images were chosen as a query images, and results are presented for two different values of radius: $\rho$ = 120 px and $\rho$ = 60 px. In the right column names and values of the five smallest ED are presented for both radiuses.

***Tab. 1.** Retrieval results in ranking order for MPEG-7 dataset*

| Query image | (ED, name) |
|---|---|
| <br>tree-7<br>$\rho$ = 120 | [(0.0, 'tree-7'),<br>(0.20954757380560468, 'car-02'),<br>(0.2129231421130642, 'car-03'),<br>(0.21349363645144995, 'lmfish-20'),<br>(0.21373099408506643, 'bottle-08')] |
| <br>tree-7<br>$\rho$ = 60 | [(0.0, 'tree-7'),<br>(0.08566661893993531, 'tree-18'),<br>(0.11683438629876214, 'lmfish-4'),<br>(0.11750287343419684, 'bottle-08'),<br>(0.12249908505010194, 'car-03')] |
| <br>car-01<br>$\rho$ = 120 | [(0.0, 'car-01'),<br>(0.04404438700011718, 'car-05'),<br>(0.06532094445649551, 'car-06'),<br>(0.08932868270131038, 'car-09'),<br>(0.10050682202982025, 'car-07')] |
| <br>car-01<br>$\rho$ = 120 | [(0.0, 'car-01'),<br>(0.030743296717113185, 'car-07'),<br>(0.04057633229879128, 'car-05'),<br>(0.052745655759383815, 'car-06'),<br>(0.05302131915863909, 'car-09')] |

For different values of radius $\rho$ retrieval results are different. Both in ranking of retrieved images and also for the same name and position in ranking values of ED are different. It means that value of radius should be chosen carefully, depending on the searched object in the image.

## SUMMARY

Implementing Zernike Moments as feature extractor in image retrieval could give significant result, because ZM are rotational invariant. They are defined over the unit disk, they are naturally unaffected by rotation, and it is proven that ZM can be computed fast. If object is translated or scaled in the image, some pre-processing is necessary.

The image retrieval by using ZM features shows that they can retrieve similar looking images for both binary and color images. The proposed future work is aimed at achieving better accuracy at retrieving images taken at different angles and in mixed lightning and weather conditions.

## BIBLIOGRAPHY


1. J. Altmann and H.J.P. Reitbock, "*A fast correlation method for scale and translation invariant pattern recognition*", IEEE Transactions on Pattern Analysis and Machine Intelligence, PAMI-6(1), pp. 46-57, 1984.
2. M.S. Hitam et al., "*Content-Based Image Retrieval Using Zernike Moments for Binary and Grayscale Images*", in "*Moments and Moment Invariants, Theory and Applications*", GCSR Volume 1, Science Gate Publishing P.C. 2014.
3. M.K. Hu. "*Visual pattern recognition by moment invariants*", IRE Transactions on Information Theory, Vol. 8(2), pp.179-187, 1962.
4. K. Jenni et al., „*Content Based Image Retrieval Using Colour Strings Comparison*", 2nd International Symposium on Big Data and Cloud Computing (ISBCC'15), Chennai, India, January 2015.
5. A. Khotanzad, Y.H. Hong, "*Invariant image recognition by Zernike moments*", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 12(5), pp. 489-497, 1990.
6. H. Mahi et al., "*Zernike Moments and SVM for Shape Classification in Very High Resolution Satellite Images*", The International Arab Journal of Information Technology, Vol. 11, No. 1, January 2014.
7. M. Mercimek et al., "*Real object recognition using moment invariants",* Sadhana, Indian Academy of Sciences, Vol. 30(6), pp. 765-775, 2005.
8. MPEG-7 dataset, , retrieved on th February 2017.
9. R. Mukundan et al., "*Image analysis by Tchebichef moments*", IEEE Transactions on Image Processing, Vol. 10(9), pp. 1357-1364, 2001.
10. R. Mukundan and K.R. Ramakrishnan, "*Moment Functions in Image Analysis: Theory and Appplications*", World Scientific, Singapore, 1998.
11. Pokemon database, http://pokemondb.net/pokedex/national#gen-1, retrieved on 7th February 2017.
12. PyImageSearch – blog about Computer Vision, retrieved on 7th February 2017.
13. R. Sabhara et al.,"*Comparative Study of Hu Moments and Zernike Moments in Object Recognition*",Smart Computing Review, vol. 3, no. 3, June 2013.
14. M.R. Teague. "*Image analysis via he general theory of moments*", J. Opt. Soc. Am., Vol. 70(8), pp.920-930, 1980.
15. C.H. Teh and R.T. Chin, "*On image analysis by the methods of moments*", IEEE Transactions on Pattern Analysis and Machine Intelligence, 10(4), 496-513, 1988.
16. Vorobyov, M., "*Shape Classification Using Zernike Moments*", Technical Report, iCamp-University of California Irvine, 2011.
17. P.T. Yap et al., "*Image analysis by Krawtchouk moments*", IEEE Transactions on Image Processing, Vol. 12(11), pp. 1367-1377, 2003.



**Użycie momentów Zernike w procesie automatycznej identyfikacji**

*W artykule przedstawiono problem automatycznej identyfikacji przy użyciu momentów Zernike. Z powodu ich ortogonalnych własności mogą przechowywać informacje o obrazie z minimalną nadmiarowością, a także są odporne na zmiany związane z obrotem obrazu. Przedstawiono system wyszukiwania obrazów bazujący na ich zawartości, a także omówiono szczegółowo dwie główne fazy działania takiego systemu. Praktyczne eksperymenty przeprowadzone na dwóch zbiorach obrazów – standardowym MPEG-7 oraz na zbiorze z niestandardowej bazy kolorowych postaci, pokazują zadowalające wyniki uzyskane dzięki momentom Zernike.*



Author:

**Artur Zacniewski, PhD, Eng.** – Polish Naval Academy in Gdynia, Poland, The Faculty of Navigation and Naval Weapons, Institute of Naval Weapons and Informatics, a.zacniewski@amw.gdynia.pl.