IMAGE INDEXING USING THE GENERAL THEORY OF MOMENTS

IMAGE INDEXING USING THE GENERAL THEORY OF MOMENTS

M. Romantan*, B. Vigouroux. B. Orza*, A. Vlaicu***
*Technical University of Cluj-Napoca, Romania, **LINA Angers, France

Abstract

The performance evolution of the computers enabled us to efficiently build and exploit software platforms dedicated to research on color image indexing and classification. This paper presents a method for unsupervised image indexing using as main classification feature the color distribution of true color images. The goal of the paper is to evaluate and experiment the applicability of the theory of moments for image indexing purposes, as well as to experiment the influence of various color space encodings. The application is coded in MATLAB and is designed to complement the *ISAAC* software Framework developed at the LISA laboratory in Angers. The application implements a Content Based Image Retrieval system that provides several useful features:

- **Operation modes**
  - ***Batch* – for the purpose of establishing the retrieval efficiency.**
  - ***Interactive* – through a User Interface.**
- **Possibility to work with distinct color spaces – RGB, HSV, YCcbCr and CIELa*b***
- **User-definable distance (metric) function.**
- **User-definable query result list length.**
- **Add and delete operations on the image database.**

Two image collections have been used to test the application: MOVI, containing specially designed series of images for the purpose of computing the effectiveness of the image indexing and classification algorithms and GOODSHOT, containing "natural", real life images. used to determine the perceptual similarity of images. The statistical results are obtained by running the application in *Batch* (non–visual) mode on the MOVI database and then centralizing the output results.

INTRODUCTION

CBIR refers to a collection of techniques for retrieving images on the basis of automatically derived features such as color, texture and shape.

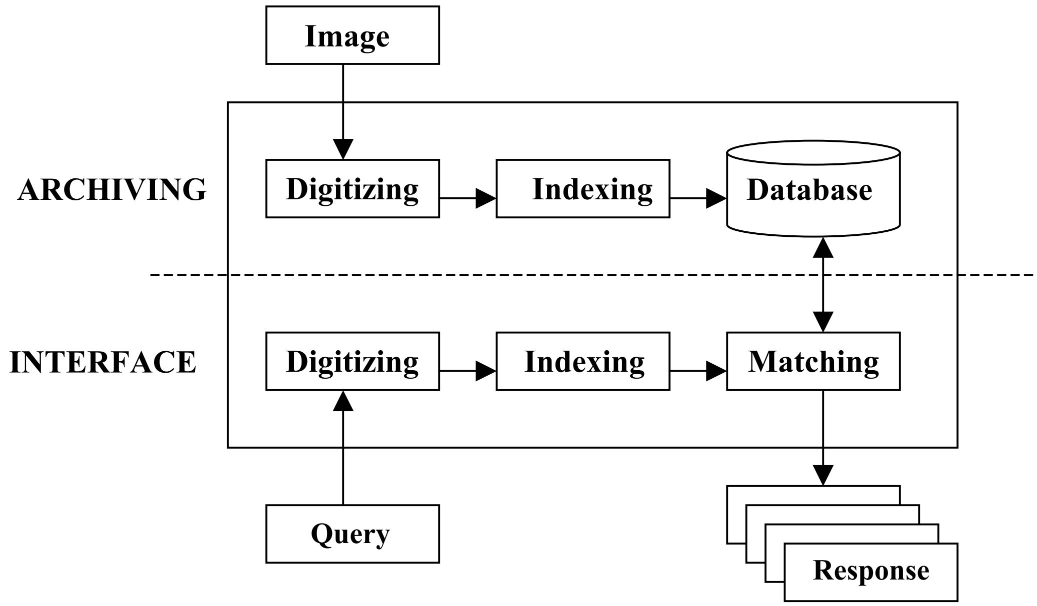
Users needing to retrieve images from a collection on a per example basis come from a variety of domains, including

- Medicine
- Architecture
- Publishing, …

Current indexing practice for images relies largely on text descriptors or classification codes, supported in some cases by text retrieval packages designed or adapted specially to handle images.

CBIR systems operate on a different principle from keyword indexing. Primitive features characterizing image content, such as color, texture, and shape, are computed for both stored and query images, and are used to identify a finite length list of the stored images most closely matching the query.

The figure 1, presented below, illustrates in a very simplified way, the architecture of a CBIR system.



MULTI-DIMENSIONAL HISTOGRAMS

When the indexing of images is based on the colors constituting that particular image, knowing the color histogram is essential. When it comes to digital images, a histogram represents the distribution of pixel intensities in an image.

The P-dimensional histogram of an image of MxN resolution with P components coded each one on Q bits consists of a P-dimensional table comprising $2^{P \cdot Q}$ cells. Each cell of this table must be able to contain a number to the maximum equal to $M \cdot N$, therefore coded on $\log_2(M \cdot N)$ bits. The histogram occupies $2^{P \cdot Q - 3} \cdot \log_2(M \cdot N)$ bytes of memory then. This represents a huge impact on memory footprint therefore a shrinking of the histogram must be enforced.

Let C be the number of cells actually occupied in the histogram $C \leq M \cdot N$. In practice, C is always largely smaller than the total number $2^{P \cdot Q}$ of available cells. Clément and Vigouroux(2001) proposed an algorithm for the computation of a P-dimensional compact histogram. What this means is that the generated histogram will not have empty cells, as well as the values being ordered by lexicographical order on the P components. Built around an optimized version of the quick sort algorithm, this algorithm provides a complexity of $Q(M \cdot N \cdot \log_2 M \cdot N)$ order. The memory space occupied by the compact histogram of an image is thus reduced by a factor:

$$\frac{2^{P \cdot Q}}{C \cdot \left(\frac{P \cdot Q}{\log_2 (M \cdot N)} + 1 \right)}$$

In the case of a color image (P=3) having a 512x512 resolution and with each color component encoded on 256 levels (8 bits-truecolor image), the classic histogram would occupy 36M bytes. If at worst of the cases each pixel of the image would contain a different color

(C=512x512), the compact histogram would only take 1.31 M bytes of storage space. Consequently, the compact histogram allows decreasing the algorithmic cost of it's handling.

GEOMETRICAL MOMENTS

The use of moment invariants was first used in 1962 by Hu in two-dimensional character recognition. The concept of moment invariants is based on the theory of invariant algebra that deals with the properties of a certain class of algebraic expressions that remain invariant under general linear transformations.

Geometric moments have been extensively employed in pattern recognition as image descriptors. In this paper we focus our attention on central geometric moments, though the method we propose could easily be extended to other kinds of moments, Zernike,Legendre,a.s.o

In this paper we make use of the (p+q) order Euclidean moments of a region defined by

$$m_{p,q} = \iint_{(x,y)\in R} x^p y^q L(x,y)\,dx\,dy \qquad (1)$$

the central moments defined by

$$\mu_{p,q} = \iint_{(x,y)\in R} (x - x_G)^p \cdot (y - y_G)^q \cdot L(x,y)\,dx\,dy \qquad (2)$$

and of the centralized normalized moments of order (p+q) defined by

$$\eta_{p,q} = \frac{\mu_{p,q}}{\mu_{0,0}^{\gamma}}, \text{ where } \gamma = 1 + \frac{p+q}{2} \qquad (3)$$

Besides the moment equations used above, 3D and Zernike moments were also employed, as well as invariant bi-dimensional moments. The mathematical model used to compute and derive these equations is beyond the scope of this paper.

EFFICIENCY COMPUTATION

We need to find a performance evaluation function to accurately describe the efficiency of the method. Thus, the efficiency of image indexing can be defined as [4]:

$$\eta_L = \frac{N_s}{N_T} \text{ ,where}$$

Ns is the number of similar images returned in the result list and **Nt** is the **gorund-truth,** that is the total number of similar images in the database. Here we need someone or something to tell us which are the similar images. In our case we will consider **Ns** as being the number of images in the result list which originate from the same folder as the query image, i.e the index of the first image not in the folder of the query image, and **Nt** as the number of images that the query image folder contains.

PRACTICAL APPLICATION

The application is encoded entirely using the MATLAB programming language. Since the version of this project was research-oriented, the choice for MATLAB and its powerful image manipulation and processing toolbox becomes obvious.

Below we present the main features of the application:

- Two ways to run the application: batch and interactive
- Usage of different color spaces in which to characterize the histogram with the theory of moments
- Choice of the moment sets to be considered when computing the similarity between the query image and the images of the database.
- User-definable similarity/distance function
- User-definable length of the query result list

IMAGE DATABASES

The synthetic test case - The MOVI Collection

This database is designed to provide useful data for computer vision algorithms and is particularly well suited to image classification and indexing algorithms.It contains several sequences of images. In each sequence what varies between the images is precisely described. This variation factor may be simple, for example a single parameter like the intensity of the main light source, or composite (several factors vary together). The aim of each sequence is to allow testing if algorithms are sensitive to some parameters, and to measure how much sensitive they are. This permits a real evaluation of algorithm performance, and a more objective comparison of algorithms.

The natural test case- The GOODSHOT Collection

This database contains around 10.000 images grouped in folders reflecting the same type of images. The difference versus MOVI is that while MOVI is an "artificial" database, GOODSHOT is comprised of images grouped into conceptual categories/folders.

RESULTS

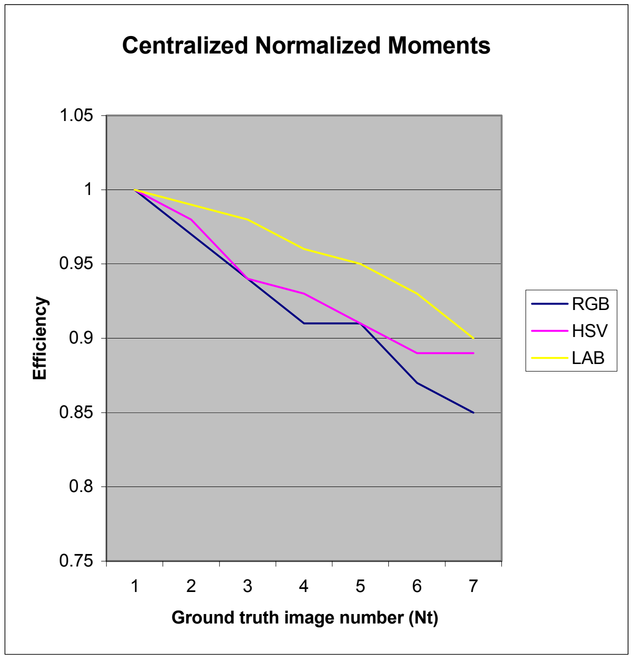
We made batch runs of the algorithm on the whole database, comparing each image of the database with the others. The table below presents the average efficiency for an incremental length of the query result list. The moments we used are centralized normalized and the Manhattan distance for each color space.So for each value of the length and for each color space we issued 595 queries.

Color Space	Query Result List Length						
	1	2	3	4	5	6	7
RGB	1	0,97	0,94	0,91	0,91	0,87	0,85
HSV	1	0,98	0,94	0,93	0,91	0,89	0,89
LAB	1	0,99	0,98	0,96	0,95	0,93	0,90

The numbers in the table represent average efficiencies of queries performed on a database with respect to the length of the result list and the chosen color space. Let us consider N the number of images in the database. We thus performed:

$$N_r = N * E * L$$

queries, where E is the number of color spaces considered and L is the number of result list lengths considered for each query sequence. In our case, $N_r = 595 * 3 * 7 = 12495$ queries. The efficiency is computed by incrementally taking Nt = 1 … 7.



This type of query where the length of the result list is limited to less than 10 images gives an approximation on the user's perspective form utilizing the application.

However, if we also take into account larger values for Nt (the ground truth), like the number of images in a folder and we also vary the distance function we obtain lower value efficiencies as presented din the tables below:

Centralized Normalized Moments	
Euclidean Metric (p=2)	
Color Space	**Efficiency (%)**
RGB	**53**
HSV	**72**
YcbCr	**44**
CIELa*b*	**15**

In what follows, we present some results obtained form the utilization of Zernike and Invariant Moments as opposed to centralized normalized ones.

CONCLUSIONS

The theory of moments is a good method for the characterization of distribution functions, thus the high applicability to histograms. The domain specific literature provides efficiency numbers similar to what we already have obtained. Retrieval efficiencies of 90% and higher are obtained by also taking into consideration other features of the image such as the texture, shape or decomposition of the image in wavelets.

The next phase in the development of this project would be to investigate the influence of each individual moment, as opposed to the technique we are currently using, to compare a full set of 1st, 2nd and 3rd order moments.

REFERENCES

[1] **John C Russ** - Image Processing, CRC Press, USA [handbook published in cooperation with IEEE PRESS 1999].

[2] **Aurel Vlaicu** – Digital Image Processing, "Microinformatica" Edition House, Cluj-Napoca, Romania.

[3] **Theo Gevers, Arnold Smeulders** – "PicToSeek : Combining Color and Shape Invariant features for image retrieval ", IEEE Transactions on Image Processing Vol.9, No.1, January 2000

[4] **M. Stricker, M. Orengo**, " Similarity of color images", Proc of SPIE : Storage and Retrieval for Image and Video Databases III, Vol. 2420, pp 381-392, Feb. 1995

[5] **Eakins J P** (1993) "Design criteria for a shape retrieval system" Computers in Industry 21, pp167-184

[6]. **Swain M.J, Ballard D.H – "**Color Indexing" – International Journal of Computer Vision, vol7, No.1, pp 11-23 (1991).

[7] **Clément A., Vigouroux B.** – "Un histogramme compact pour l'analyse d'images multi-composantes", GRETSI 2001, Toulouse (France), 10-13 sept. 2001.