# Improved System for Content Based Image Retrieval Based on Pyramid Decomposition in the Spectrum Domain

Agata Manolova[1], Roumen Kountchev[2], Irina Aleksieva[3]

***Abstract*** – **One of the most important problems concerning the e-client management of large databases is the creation of methods for content-based image retrieval. The goal of this study is to improve the system for fast image retrieval, using pyramid image decomposition in the spectrum domain developed in [1]. The image retrieval is performed on the basis of evaluation of the multi-layer distance between the compared images. This approach permits the use of a recursive algorithm with relatively low computational complexity.**

*Keywords* – **Content based image retrieval, Inverse Differential Pyramid, Colour image database.**

## I. Introduction

The rapid advancements in multimedia technology have increased the relevance that repositories of digital images are assuming in a wide range of information systems. Effective access to such archives requires that conventional searching techniques based on external textual keywords be complemented by content-based queries addressing appearing visual features of searched data [2], [3]. Content-based image retrieval (CBIR) could be described as a process frame work for efficiently retrieving images from a collection by similarity. The retrieval relies on extracting the appropriate characteristic quantities describing the desired contents of images. In particular, different techniques were identified and experimented with to represent the content of single images according to low-level features, such as color [4], [5], texture [7], [8], shape [9], [10], and structure [11].

The current work is divided in three major steps:

The first part consists of image preprocessing for the needs of the algorithm;

The second stage of the proposed work is the development of the content based image retrieval algorithm based on the Inverse Differential Pyramid (IDP);

The last part includes the implementation of the algorithm on C++ and testing it on databases – small with 301 images divided in four semantic types and extended database with 4468 images divided in 53 semantic types and conclusions.

[1]Agata Manolova and [2]prof. Roumen Kountchev are with the Faculty of Telecommunications at TU - Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: amanolova@tu-sofia.bg, rkountch@tu-sofia.bg .

[3]Irina Aleksieva is with University Joseph Fourier, 621 avenue Centrale , Domaine Universitaire, 38400 St Martin d'Heres, France.

## II. Development of the Incremental Algorithm for Image Comparison with IDP

### A. Image analysis and preprocessing

Image representation based on transforming the image in another function is a very suitable choice which allows to obtain easily comparable objects and also to do the comparison on a multilevel scale.

There are some necessary conditions for the IDP algorithm depending on the image dimensions and the spectral coefficients as described in [12]. The first condition of the preprocessing is that the image should be in greyscale i.e. there should be a conversion from the RGB or other color space to the YCbCr model where we keep the Y component. The use of the YCbCr color space is not random. It is widely used in image and digital photography systems. The YCbCr color space has the advantage of low correlation between its components. The second important condition is that all the images in the database must have the same height / weight m ($m = 2^n$ and $n \in IN$). Unfortunately few images meet this requirement. There are various ways to handle that problem depending on the database. The simplest way is to resize the image with an interpolation of the pixels in order to obtain the right dimensions. As the standard method for image interpolation is not suitable for this system due to a change in the general forms of the image which perturbs the transform, the proposed algorithm for modified image interpolation can provoke a slight twist in the general forms of the image but usually it is insignificant for the transform. An important perturbation can occur just in cases where the width and the height strongly differ [13]. This algorithm is based on floor and ceiling functions. If the image size is WxH where $W < m$ and $H < m$ (which is mostly the case), the rows and the columns are separately treated. The first step is to compute the scale between the old width (resp. height) and the new one. Supposing the old width is W and the new is w then the ratio is W/w. Then the width coordinate of every pixel from the original image is multiplied by the ratio and the floor function of this multiplication is put in an accumulator. The accumulator receives the values of every pixel of the original image that have obtained the same floor. Computing the value of the new pixel simply demands the mean value of all the pixels in the accumulator as shown on the figure 1. Compared to the standard interpolation approach the results with the modified implementation method present much less distortion. This algorithm is computationally non expensive and easy to implement.

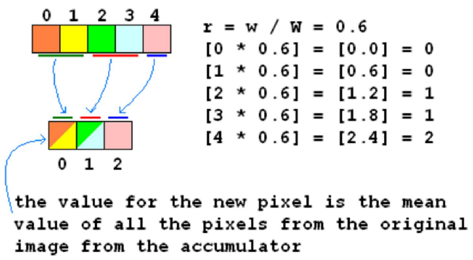


Fig. 1 Modified Interpolation algorithm scheme

*B. Image decomposition with Inverse Differential Pyramid*

The idea behind this approach is transforming the images from the database to obtain different levels (called layers) providing simplified information about the image. The method used for this decomposition is called Inverse Differential Pyramid (IDP), [1], [12] and it comprises the following steps:

1. The original image [B] with size $m^2$ ($m=2^n$) pixels is presented as a sum of components, presenting the image decomposition in the spatial domain in correspondence with the relation:

$$\left[B(2^n)\right] = \left[\tilde{B}_0(2^n)\right] + \sum_{p=1}^{n-1} \tilde{E}_{p-1}(2^n) + E_{n-1}(2^n) \tag{1}$$

where p = 0, 1, 2, ..., n-1 is the sequence number of the decomposition component.

2. The components of the decomposition (1) are defined as follows:

i. Image decomposition with Inverse Difference Pyramid in (1) for p=0 is presented with the matrix:

$$\left[\tilde{B}_0(2^n)\right] = \left[T_0(2^n)\right]^{-1}\left[\tilde{S}_0(2^n)\right]\left[T_0(2^n)\right]^{-1} \tag{2}$$

where $[T_0(2^n)]$ is the matrix, of the used 2D linear orthogonal transform, in this case the Discrete Cosine Transform (DCT). There are infinity of function transform and the use of every transform depends on the approach. Since this study is about image comparison the applied transform is DCT which is a transform commonly used in image manipulation. It is also a non-loss transform meaning that if DCT is applied to an image and then we run over the IDCT (Inverse Discrete Cosine Transform) the result will be the entry image. It is possible to have a slight loss of information due to a computer approximation error but it is insignificant.

$$\left[\tilde{S}_0(2^n)\right] = m_0(u,v)S_0(u,v)$$
$$m_0(u,v) = \begin{cases} 1 \text{ for } (u,v) = (0,0),(0,1),(0,2),(1,0), \\ \qquad\qquad (1,1),(1,2),(2,0),(2,1) \\ 0 \text{ elsewhere} \end{cases} \tag{3}$$

The choice of the binary mask $m_0(u,v)$ is based on previous studies about the IDP algorithm efficiency [12]. $S_0(u, v)$ are the elements of the spectrum matrix, calculated in accordance with the transform:

$$\left[S_0(2^n)\right] = \left[T_0(2^n)\right]\left[B_0(2^n)\right]\left[T_0(2^n)\right] \tag{4}$$

The values of the elements build the lowest layer (p=0) of the spectrum pyramid.

ii. The next components of the decomposition (1) for p = 1, 2, 3, ..., n-1 are represented with the matrix:

$$\left[\tilde{E}_{p-1}(2^n)\right] = \begin{bmatrix} \left[\tilde{E}^{1}_{p-1}(2^{n-p})\right] & \left[\tilde{E}^{2}_{p-1}(2^{n-p})\right] & -- & \left[\tilde{E}^{2^p}_{p-1}(2^{n-p})\right] \\ \left[\tilde{E}^{2^p+1}_{p-1}(2^{n-p})\right] & \left[\tilde{E}^{2^p+2}_{p-1}(2^{n-p})\right] & -- & \left[\tilde{E}^{2^p+p}_{p-1}(2^{n-p})\right] \\ --- & --- & & --- \\ \left[\tilde{E}^{4^p-2^p+1}_{p-1}(2^{n-p})\right] & \left[\tilde{E}^{4^p-2^p+2}_{p-1}(2^{n-p})\right] & -- & \left[\tilde{E}^{4^p+1}_{p-1}(2^{n-p})\right] \end{bmatrix} \tag{5}$$

$$\left[\tilde{E}^{k_p}_{p-1}(2^{n-p})\right] = \mathrm{IDCT}\left(\tilde{S}^{k_p}_{p}(2^{n-p})\right)$$
$$\text{for } k_p = 1,2,\ldots,4^p \tag{6}$$

$$\left[\tilde{S}^{k_p}_{p}(u,v)\right] = m_p(u,v)S^{k_p}_{p}(u,v)$$
$$m_p(u,v) = \begin{cases} 1 \text{ for } (u,v) = (0,0),(0,1),(0,2),(1,0), \\ \qquad\qquad (1,1),(1,2),(2,0),(2,1) \\ 0 \text{ elsewhere} \end{cases} \tag{7}$$

$$\left[E_{p-1}(2^{n-p})\right] = \begin{cases} \left[B(2^n)\right] - \left[\tilde{B}_0(2^n)\right] \text{for } p = 1 \\ \left[E_{p-2}(2^{n-p}) - \left[\tilde{E}_{p-2}(2^{n-p})\right]\right] \\ \text{for } p = 2,3,\ldots,n-1 \end{cases} \tag{8}$$

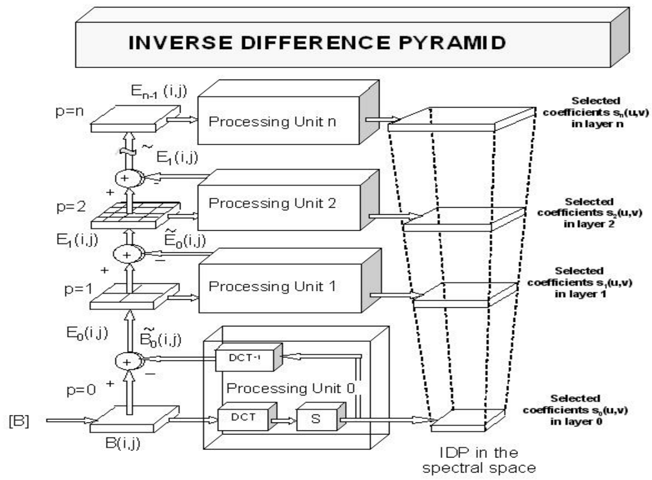


Fig. 2 IDP algorithm

*C. Multilevel image distance*

In this study the images are compared in the frequency domain (after the application of the DCT). As we know, not all the coefficients in the image-corresponding matrix in the frequency domain bare significance to the object information

i.e. most of them equate to 0. Furthermore we know where are located the 8 most important coefficients (top left corner).
Therefore when we compare two objects on a current pyramid layer comparing the two matrices for all terms is unwise because it would not give us further lore (and it also could provide expensive in terms of time and memory). Thus we could compare only 8 coefficients per entry block. This give us 8 comparisons for level 0, 8 x 4 coefficients for level 1, 8 x $4^2$ for level 2, etc to the last level n where we have $8x4^n$.
So the total length of the comparator vector is $8\times\frac{4^{n-1}}{3}$ coefficients.
The Multilevel Image Distance computation is done by levels. On each level is computed the difference between the coefficients of that level using the Euclidian distance. The difference of the coefficients of the query image and the current image from the database we compare to is then summed:

$$D_{r,q}\left[\widetilde{E}_{r-1,q}\left(2^{n-r}\right),\widetilde{E}^{k}_{r-1,q}\left(2^{n-r}\right)\right]= \\ =\sum_{u-0}^{2^{n-r}}\sum_{v=0}^{2^{n-r}}\left|m_{r,q}(u,v)\right|\left[S_{r,q}(u,v)-S^{k}_{r,q}(u,v)\right] \quad (10)$$

for q=1,2,…,$4^r$ and r=1,2,…, n-1

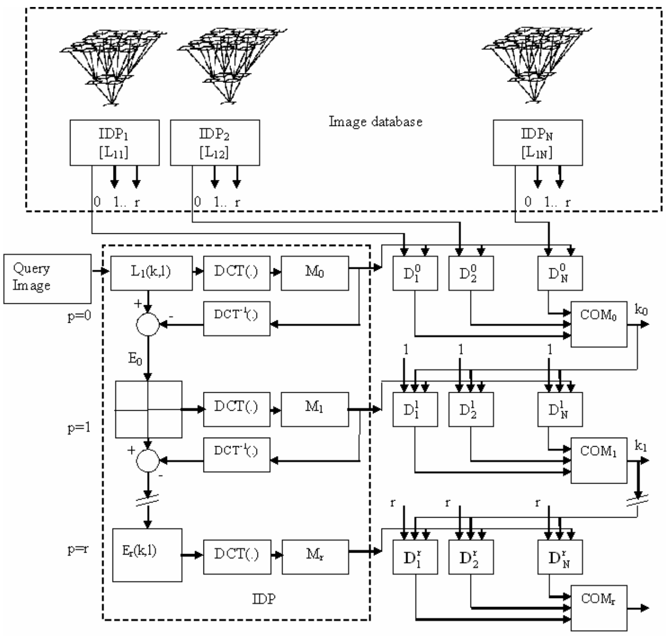

Fig. 3 Multi-layer coarse-to-fine similarity measure

*D. CBIR algorithms based on IPD*

1. The Incremental Algorithm

Image comparison can be expensive in terms of time and memory. Thus we need a flexible algorithm which focuses on similar images and ignores the rest of the database, enclosing the database to a smaller set at every step/level of the pyramid. Such algorithm can be an incremental algorithm:
If N is the total number of images in the database and we are looking for the 3 to 5 closest images:

a. Compute $M_{in1}$ (a vector containing the $N_1$ images closest to the query image at level 1 where $N_1 < N$);

b. Compute $M_{in2}$ (a vector containing the $N_2$ images closest to the query image at level 2 where $N_2 < N_1$);

……………………………………………

n. Compute $M_{inn}$ (a vector containing the $N^{(p)}$ images closest to the query image at level p-1 where $N^{(p)} < N^{(p-1)}$);

Compute $M_{in}$ (the vector containing the 3 or 5 final images closest to the query image at level p)
The computational complexity of this algorithm is quite low. There are operations $\sum_{i=1}^{p}N^2(i)+2\sum_{i=1}^{p}N(i)$ at all, which is a polynomial algorithm cost.

2. The Prototype search

There are several possible approaches when choosing the prototype. In this study the prototypes were random; hence the database classes were highly homogenous.
Supposing that there are q image classes s.a. [$C_1$,…, $C_q$], we could choose couple of representative images par class (the prototypes, j << number images per class) called keys: [$K_1(C_1)$,…, $K_j(C_1)$]; [$K_1(C_2)$,...,$K_j(C_2)$]; …; [$K_1(C_q)$,…,$K_j(C_q)$]. Instead of comparing every query image to the whole database we could just compare them to the keys and keep the r most interesting classes (i.e. where the r << q) and then compare the query image with all the images belonging to the selected classes to find the closest matches [10].

## III. Experimental Results

There were two image databases used in the study. The first database consisted in 301 color images divided in 4 classes and a second one - a much larger set of 4468 images grouped in 53 classes. For both the IDP algorithm and the comparison algorithms had to be chosen a suitable platform, the programming language is C++, we have decided to use the Qt4 toolkit and ImageLib for the FT functionalities.

On the big database were experimented different approaches of the incremental algorithm. The image retrieval to the last level (5 levels) proved to be not always necessary and furthermore not constantly the most precise. However this is easily explained by the diversity of the images in the database. The upper levels of the pyramid are far to detailed to give accurate information with a highly heterogeneous collection.
The results in Table I were obtained by testing every image in the database and an evaluation of the retrieval accuracy. Ever image was tested halting the incremental algorithm on five different levels from 0 to 4. The results show that it is not necessary to run the algorithm to the last level.

TABLE I
RETRIEVAL BY PYRAMID LEVELS

| Image Level | 0 | 0 and 1 | 0, 1 and 2 |
|---|---|---|---|
| Retrieval accuracy | 78 % | 81 % | 82 % |

On Figure 4 some visual results are presented for three different levels. The first image is the query image and the number below the image is the distance between the current image and the query image.

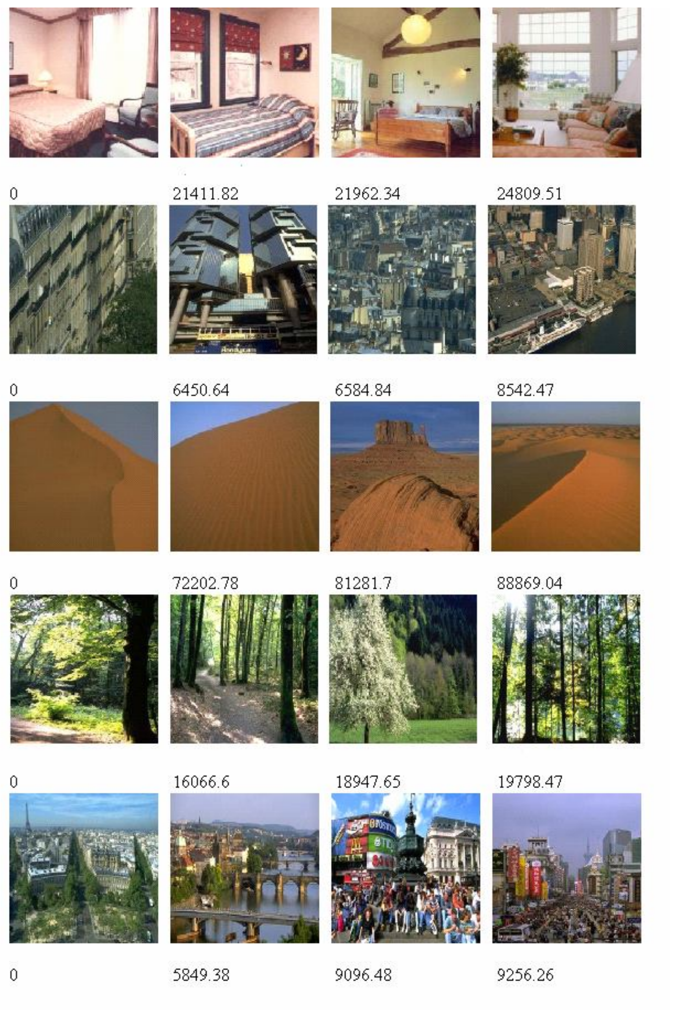


Fig. 4 CBIR results

## IV. Conclusion

The presented method for efficient image retrieval is based on new image decomposition in the spectrum domain (the Inverse Difference Pyramid). The method permits the evaluation of the images similarity to be performed in the consecutive decomposition layers and to compare parts of the images with increasing resolution, which corresponds to different image scaling. The main advantage of the presented approach is the relatively low computational complexity and the capability to avoid difficulties, due to cropping or other kinds of image editing. All the retrieval computations are done in real time. The time for the retrieval of one query image and displaying the results for the small base without prototypes is 0.07s for 5 levels. The time for the same base but with a prototype search for the same levels is 0.002s. The time for the big base for five levels is 0.9s.

The search speed is one of the most important qualities of the presented method. As a result of the IDP decomposition the search process is significantly accelerated when compared with the well-known Exhaustive Search Algorithm (ESA) [10], for which the search is based on the use of a 3-dimensional color feature vector. For the evaluation were calculated the numbers of operations, necessary for the successful performance of the two algorithms.

Moreover, it proves that the computational complexity can be reduced using the incremental algorithm with a halt condition, or in a case of well-defined classes a prototype search.

All this puts the developed software on a market competitive level and qualifies it as industrially useful.

## Acknowledgement

The authors would like to thank the Gipsa-Lab for the support and the opportunity to make this study. There also should be special gratitude to Anne Guerin-Dugue for the guidance and the ideas. This work was financed by a project grant of the National Fund for Scientific Research of the Bulgarian Ministry of Education and Science by the contract VU-I-305.

## References

[1] R. Kountchev, A. Manolova, "General Approach for Fast Image Retrieval from Image Database using Consecutive Iterations in the Transform Domain", Proc. of the Technical University - Sofia, Bulgaria, Vol. 55, 2005, pp. 167-176.

[2] A. Gupta and R. Jain, "Visual information retrieval", 1997

[3] A. Del Bimbo, *Visual Information Retrieval*, Academic Press, San Francisco, CA, 1999.

[4] A. Smeulders, M. Worring, S. Santini, A. Gupta, and R. Jain, "Content based image retrieval at the end of the early years", IEEE Trans. on Patt. Anal. and Machine Intelligence, 22, 1349–1380, 12.2000.

[5] T. Gevers, *Principles of Visual Information Retrieval*, M. Lew, Ed., ch. Color in image search engines. Springer-Verlag, Heidelberg, February 2001, 11–48.

[6] R. Schettini, G. Ciocca, and S. Zuffi, *A Survey of Methods for Color Image Indexing and Retrieval in Image Databases*, ch. Color imaging science: Exploiting digital media, R. Luo and L. Mac Donald, Eds., JohnWiley & Sons, New York, 2001.

[7] N. Sebe and M. Lew, *Texture Features for Content-Based Retrieval*, ch. Principles of visual information Retrieval. Springer-Verlag, Heidelberg, 2001.

[8] D. Hoiem, R. Sukthankar, H. Schneiderman, and L. Huston, "Object-based image retrieval the statistical structure of images", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Washington, DC, IEEE Computer Society, Vol. 2, June 2004, pp. 490–497.

[9] Andrew P. Berman, Linda G. Shapiro, "Efficient Content-Based Image Retrieval: Experimental Results", IEEE Workshop on Content-Based Access of Image and Video Libraries (1999).

[10] Cawkell Tony, "Image indexing and retrieval by content", Information Services And Use 20, no. 1 (2000): 49-58.

[11] Roumen Kountchev, Veronique Haese-Coat, Joseph Rosnin, *Inverse Pyramidal Decomposition with multiple DCT*, Signal processing: Image communication 17 (2002) 201-218 Elsevier 2002.

[12] Meijering, Erik, "A chronology of interpolation: from ancient astronomy to modern signal and image processing", Proceedings of the IEEE 90 (3): 319342, DOI 10.1109/5.993400 (2002).