БЪЛГАРСКА АКАДЕМИЯ НА НАУКИТЕ . BULGARIAN ACADEMY OF SCIENCES

КИБЕРНЕТИКА И ИНФОРМАЦИОННИ ТЕХНОЛОГИИ · Том **4**, № 2
CYBERNETICS AND INFORMATION TECHNOLOGIES · Volume **4**, No 2

София . 2004 . Sofia

Fast Image Retrieval by the Tree of Contours Content*

Dimo Dimov

Institute of Information Technologies, 1113 Sofia
E-mail: dtdim@iinf.bas.bg

Abstract: *The paper presents a method for fast access by content into database (DB) of images (IDB). The proposed method is based on the image tree of contours defined for graphic images, as well as on the one-dimensional complex Fourier transform of the contours. In this way, problems connected with the image invariance to translation, rotation, scaling, reflection and intensity are currently solved. The most essential image data are ordered by importance in a key of fixed length, on which the fast access is performed using the well-known index access methods of a conventional DB management system (DBMS). Problems connected with possible image noise are also discussed. The method is tested on a IDB of about 4000 images of trademarks.*

Keywords: *content based image retrieval, fast image retrieval, tree of image contours, Fourier's transform of contours, image databases.*

1. Introduction

Regardless of recent advance in Artificial intelligence area and the object oriented approach in software as well, the application of conventional DBs, generally designed for textual, numerical and/or Boolean data, seems to sustain still for a long time. Now-a-days DBs, most of all of relational type, are currently extended to keep binary data records of tremendous volume, which interpretation as one- or multi-dimensional signals, for instance – images, speech, movies, etc., is most often left to the user's competence [2, 12, 23]. For instance, there is a large number of image DBs (IDBs), many of them available on Internet that need proper retrieval tools [20].

* This research was jointly supported by the following grants/funds: (i) the Grant No RC6/2004 (IIT No 0210143) of the ICT Development Agency at Bulgarian Ministry of Transport & Communication, (ii) the Grant No I-1306/2003 (IIT No 020057) of the National Science Fund at Bulgarian Ministry of Education & Science, and (iii) the BAS research funds of IIT disposal (IIT No 010056).

A number of systems could be encountered in world practice trying to meet the above challenge, e.g. TRADEMARK (1992), QBIC (IBM, 1993), CORE, Photobook (1994), Image Engine (Virage), Image Retrieval Ware (Excalibur), ARTISAN (1999), PictureFinder (2002), cf. aslo [6, 7, 19, 21, 23].

For instance, the early method of GEMINI [23] is mainly optimized for searching of 1D time series and consequently extended to images by an analogy. GEMINI defines a feature space for recognition generally on RGB-histograms that is obviously insufficient for an adequate image representation. An object-based approach of image retrieval by similarity [12] is based on a sophisticated segmentation method for cognitive contour parts of the pictures in the DB. In ARHOR, a multimedia system for historical documents, the pictures included in the DB of mixed documents are searched by special image tags [2]. ARTISAN, a system for marks retrieval, tries to concentrate on shape representation, especially on oftenly used shapes (lines, circles, ovals, etc) and on their colinearity, similarity and proximity as well [7]. At the same time PictureFinder candidly relies on the user's competence to label the images and to resolve the collisions at retrieval.

A relatively new area of Informatics, called Content Based Image Retrieval (CBIR), appeared recently to solve the problems of above practice. CBIR covers techniques for automatic (or automated) retrieval of image and/or video objects by features as color, texture, shape, movement, etc. [6, 19]. The now-a-days CBIR is often qualified as an "early" one, because of considering predominantly simple statistics (histograms) of these features [6]. The last are usually preferred to more sophisticated structures like contours, trajectories, etc., because of the well-known difficulties encountered by segmentation of graphical objects, even in cases of low level of image noise [19, 21].

A general characteristic of CBIR systems and available IDB as well is that they are developed over conventional DB Management Systems (DBMS), which methods for fast data access by indices are optimized basically for text, number and logical data. The well known extensions of these methods for multimedia data (R-trees, k-trees, etc. [19, 23]) do not account the CBIR specifics of image recognition, because they are developed generally for GIS (Geographic Information Systems) technologies. In this way, CBIR systems tend to a sequential data access, what is unacceptable, especially in cases of large IDB. In this way the interest in CBIR naturally arises towards approaches better adapted to the index access techniques of conventional DBMS.

The basic idea herein is to develop a fast access method for searching images by their content and the method to be easily implemented into conventional DB of images for an efficient retrieval. At this stage, images are considered most of all graphic images (gray scaled or colored), and their content is considered the images themselves or their sketches.

2. Image search strategy

The proposed strategy is based on the "fast retrieval by key" paradigm of conventional DBMS for texts. The idea is to represent the essential image content as a well-structured key of fixed length.

2.1. Databases of images

There are two classical approaches of using conventional DB for keeping images recently [2, 3, 9, 12, 19, 20, 23], namely:

- by keeping a table of pointers to (or of the names of) the image files that are written anywhere outside the DB, or
- by keeping the images themselves into a table of BLOB (Binary Large OBject) fields.

Both the approaches rely on the convenience of searching the images using conventional access methods of DBMS, namely, sequential access methods, relative sequential ones, and especially the faster ones – index-sequential (or parallel) ones, direct hash-access ones, associative ones, etc. [11, 13, 14, 23]. All fast access methods work using the convention of a "key" and give the search result as follows:

$$(1)\qquad \langle \textit{the key value of the retrieved object (record)} \rangle \leftharpoondown \\ \leftharpoondown \langle \textit{the key value of the searched object (record)} \rangle$$

where "$\langle i \rangle \leftharpoondown \langle ii \rangle$" means "$\langle i \rangle$ is the greatest value less than or equal to $\langle ii \rangle$".

The key usually corresponds to a feature (or a combination of features) defined for all DB objects of a given type. As a rule, the key fields should be simple fields of fixed and not very large length.

Because of the contemporary object-oriented DBs inherit the well spread relational DBs, which in their turn inherit the old-fashioned DBs of network type, and all of them possess as a minimum a kind of index-sequential access method as well, the choice for experiments here is made on:

- a DB of relational type to keep images;
- an index-sequential access method to realize the search strategy here proposed.

Most of the practical cases of efficient DB image access rely on simple description of the searched content that is relatively easy to be derived automatically from the input image, and because of that suffer of low recognition ability. More sophisticated methods for image content derivation are, as a rule, not well suitable to general access methods of conventional DBMS, and consequently are high computer power consuming, i.e. not efficient enough.

Besides efficiency, (in processing speed and noise tolerance), the assess method proposed here aspires also to the conventional DBMS cannon – DB objects to be accessed as a whole, and eventual small differences to be considered as admissible noise, i.e. in a tolerance of the DB sensitivity.

2.2. Images of interest

Graphic images that will be of main interest hereinafter can be considered as reproduced by a small number of colors or half tones (i.e. gray intensities), and the respective areas are of relatively not very small size. Besides, these images should be well capable for centralized localization, i.e. the essential graphics to be in the "middle" of the image frame, while the area near the frame to be filled only with background color (or half-tone). Examples of similar images are illustrated in Fig. 1. They are intensively used in patent offices' activities, e.g. for registration of companies, firms, etc., and are popular by the names "hallmarks", "trademarks", or simply "marks" [10, 15].

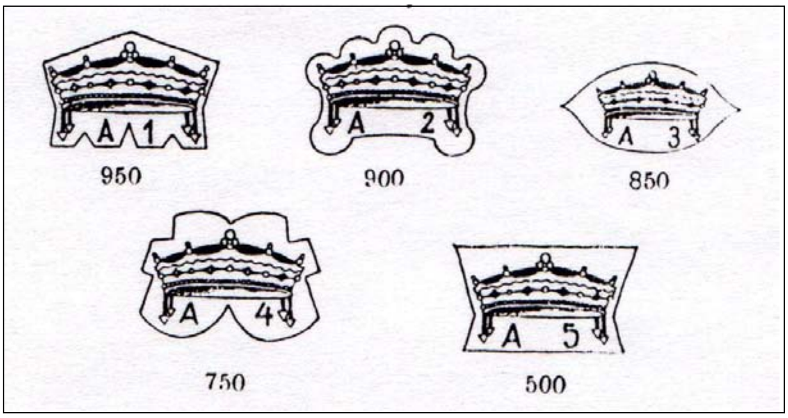

Fig. 1. Image examples suitable to the method proposed

2.3. Requirements to the image key definition

- The key should consist of the (most) essential information of the image.
- This key information should be of significantly smaller volume than the image itself.
- The key information volume should be written in a 1D array of fixed length.
- The key information should be structured in a way that essential parts to appear in frontier positions.
- The image key is not obligatory unique, i.e. there could exist more than one corresponding image for a given key length.
- All DB images of one and the same key should be considered as equivalent images.
- The key length should be chosen in such a way that the image key to be sufficiently informative.

Thus, the following intuitive definition about keys can be written:

Definition 1. The *image key* or simply the *key* should consist of the most essential image information structured in a descending order into a one-dimensional (1D) array of fixed length.

2.4. Tree of contours

The representation of a given image is popular as a scalar function $g(x, y)$, $(x, y) \in X \times Y$ characterizing the intensities (or colors) over a rectangular pixel area $|X \times Y|$, defined by the image dimensions – the width X and the height Y (in pixels).

The same image function can be also considered as given by a set $G \equiv \{g_i\}$ of iso-curves g_i, $g_i \equiv g(x, y) = i$, each of them defined by an intensity level $i \in I$, I is the set of image intensities. Besides, each iso-curve is a closed and non-self-crossing curve that bounds a one-connected area of pixels. In general, the number $|I|$ of intensity levels depends on the resolution ability on intensity (or color), what is chosen at the stage of image capturing, for instance $I_{max} = 256$ levels of gray, or $I_{max} = 3*256$ RGB levels, etc. In many practical cases, the number $|I|$ of essential (or really presented) levels scarcely depends on I_{max}, i.e. $|I|$ is constant, and as a rule, $|I| << I_{max}$.

Thus, and without more details, we can consider the images of interest as composed by several one-connected regions (i.e. without any holes), placed one over another, and each of them respectively filled by its color (or half-tone). This representation can be arranged in such a way that each two regions to be either entirely included one other or non-crossing ones. Besides, its contour and filling color (or tone) uniquely represents each region.

In this way, a given image can be represented by its “coloured” contours structured in a tree of inclusion, and this representation will be called hereinafter “tree of contours” of the image, cf. Fig. 2. Something more, this representation can be considered as an isomorphism that is obvious in the case of two colors only, and is of main interest hereinafter.

Furthermore, the colour data could be dropped out the tree of contours, because of a supposition that the images of interest are preliminary binarized, i.e. converted into two tones – black and white (object vs. background). The “filling colour” here is changing alternatively form level to level of the tree, i.e. if start with “white” the image will considerably be a “positive”, else if start with “black” it will be a “negative”.

Finally, considering the image key definition, the following can be intuitively said for the essential image data and their ordering by importance:

– Given contour is as much important as close situated to the root of the tree of contours.

– The most important contour among those of a given level should be the one of greatest area, or perhaps the curliest one, i.e. the one of the longest perimeter per an area unit, etc.

– The root can be only one if the image frame is considered also as contour. Of course, it does not mean that the root is an important contour; but its primary successors should be.

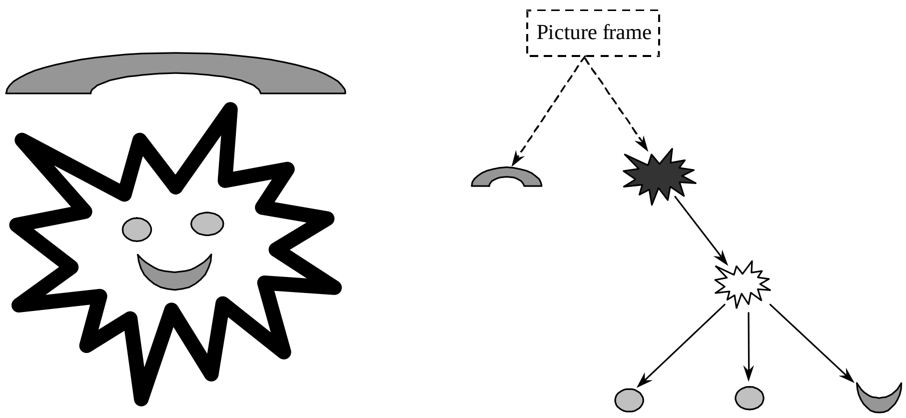

Fig. 2. A graphics and its tree of contours

Obviously, the image representation by the tree of contours is isomorphic to the bitmap representation. In this line of thinking, we will consider a *noise* all the differences between the original image and the reverse image (i.e. that can be obtained after respective filling of contours).

3. Algorithm of the proposed method

Described in brief the proposed method for fast image retrieval can be outlined algorithmically by the following steps:

A1. Image binarization playing as tone/colour intensity normalization
A2. Creation of the tree of contours
A3. Complex one-dimensional Fourier's transform for each image contour
A4. Translation, scale and reflection normalization for the contours
A5. Composition of the spectra matrix ordering the contours by importance
A6. Derivation of the image key items by the spectra matrix

Illustrations of these steps are given by Figs. 3-9 and 11 on a real example. More details follow hereinafter.

3.1. Preliminary image binarization (A1)

The input image (in BMP format) is first binarized and then represented by its set of contours, cf. Figs. 3, 4 and 5. The resulting tree of contours should be obtained at the next step A2.

We consider that in cases of the graphic images of interest, the shape is much more essential than the colouring. That's why the original images (true color or gray scale) are preliminary converted to black-and-white (B/W) ones, i.e. binarized.

On one hand, binarization facilitates the performance of contouring. So, if the input images are too noisy, then a sophisticated approach to binarization will be necessary, e.g. like [25].

On the other hand, binarization is also considered as a process of image intensity normalization. It should be extended to segmentation by areas of mean tones (or colors) for more appropriately to exploit the idea of the tree of contours, cf. [1].

From another viewpoint, we may consider the input picture as composed only by a background (the root) and either an object (i.e. all contours below the root) or a set of objects (each one initiated from a contour of first level).

From one more viewpoint, we can base on the object being derived by a precise binarization but considering its full tone (or colour) spectrum inherited by the original image.

Fig. 3. An original mark image

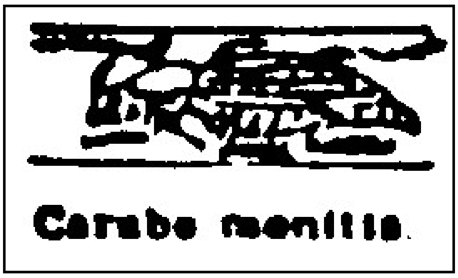

Fig. 4. The binarized image

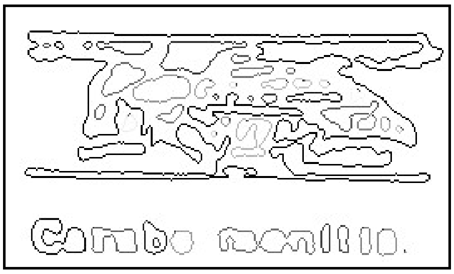

Fig. 5. All contours of the image

All these interpretations are possible extensions at this step of the method.

For reasons of simplicity for the experiments here, we use the global histogram optimization approach of O t s u [16] for preliminary binarization of images of interest.

3.2. Creation of the tree of contours (A2)

The contouring procedure, let's call it *TraceBMP()* can be designed using well known approaches, for instance following the ever green reading of P a v l i d i s [17]. Most of the local troubles pointed there, e.g. for the pixel connectivity, for possible coincidences of contour parts, etc., can be resolved using solutions fro computer graphics, e.g. following F o l e y & v a n D a m [8].

To save memory [2, 17], the contours are represented by a Freeman's code instead of Cartesian coordinates only. Each contour C is represented by the pair (c_0 , C^*) where c_0 is the start vertex given absolutely (in Cartesian coordinates), and C^* is the rest chain of vertexes given relatively (in Freeman's codes). More precisely, $C^* \equiv (c_i \mid i = 1, \ldots, L)$, $c_i \in \mathbf{F}$, $\mathbf{F}$ is the code set for the corresponding Freeman vectors – from a current pixel to its surrounding pixels. L is the length of C^*, consequently the number of all vertexes is $L+1$, because of contours are definitely closed (i.e. the start and final vertexes coincide). There are two practical choices for $\mathbf{F}$, either $\mathbf{F}=\mathbf{F}_4$ for 4 basic directions, which vectors are of length 1, or $\mathbf{F}=\mathbf{F}_8$, $\mathbf{F}_8$ – an extension of $\mathbf{F}_4$ with extra 4 diagonal directions, which corresponding vectors are of length $\sqrt{2}$.

Which version to choose, $\mathbf{F}_4$ or $\mathbf{F}_8$, is not a principal problem because of the contours' smoothing performed by necessity in the next step A3. Anyway, the choice of 8 directions brings more precise approximation of contours but leads to extra troubles due to the pixel connectivity already mentioned.

It is important to outline that the colour data ("black" or "white") can be omitted here, because of their alternative change following the order of contours' inclusion. At the same time, other intermediate results as contour areas, perimeters, etc., are quite useful and should be kept in the tree nodes for speediness.

3.3. Complex one dimensional Fourier transform of contours (A3)

The interpretation of a 2D curve as complex function of a real argument is very convenient for emphasizing the essential graphic features, especially by exploitation of the well known fast algorithm for 1D discrete Fourier transform, i.e. the so called Fast Fourier Transform (FFT). For instance, D u d a & H a r t [5] suggest the first dozen of the frequencies, while D i m a u r o [4] assures that only the first 4–5 of them are just enough for visual recognition of the ten Arabic digits, even of being hand-written ones.

As a facilitation to this approach, our image contours are always closed and non-self-crossing curves. Each contour can be discretely represented by a polyline that bounds a one-connected polygon. In order to exploit easy the FFT, it remains preferable, of course, to redraw each contour in a way that its new length (i.e. number of points) to equal a power of 2.

Instead of using a canonical 1D Fourier transform [4], the decomposition into frequencies is interpreted here most of all as complex Fourier series, which discrete version, by the way also leads to a FFT performance, cf. Figs. 6 and 7. This interpretation can easy assure us in the translation, rotation and scale invariance of the frequency contour representation of the image, cf. also [18, 22].

The processing at this stage, let's call it *FourierSpectrum()*, can be described in brief by the following 3 sub-steps.

(A3.1) *Fourier is Spectrum()* considers each image contour from the tree, cf. Figs.6, 6a, 6b, as a one-dimensional complex function $Z(t)$ of a natural argument t, $t = 0, 1, \ldots, T-1$, T being the number of contour points. To obey the correct FFT conditions, each contour data are subjected to the two preliminary conversions:

a) the relative representation (the Freeman's chain at A2) is first converted to Cartesian coordinates, i.e. to the complex representation $C(l) = x(l) + j\times y(l)$, $l = 0, 1,\ldots, (L-1)$, L is the number of integer points $(x(l), y(l))$ and j – the complex unit ($j^2 = -1$);

b) the complex representation C is smoothed to the desired regular representation $Z(t)$, $t = 0, 1,\ldots T-1$, where the period T is computed as $T = 2^k \geq L$, k is the minimal integer about that, and the distances between each two consecutive points $(x(t), y(t))$ and $(x(t+1), y(t+1))$ to be one and the same.

(A3.2) *Fourier is Spectrum()* gives the contour's Fourier spectrum as a complex series $F(\omega) = \mathrm{Re}(F(\omega)) + j{*}\mathrm{Im}(F(\omega))$, $\omega = 0, 1,\ldots, T-1$. A geometric interpretation of $F(\omega)$ in the type $F(\omega) = |F(\omega)|{*}\exp(j{*}\mathrm{Arg}(F(\omega)))$ is illustrated in Figs. 7, 7a, 7b.

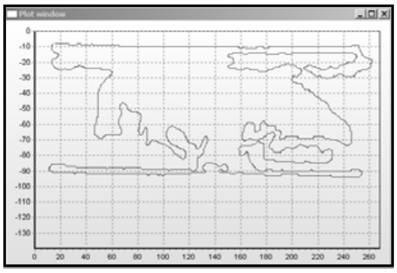

Fig. 6. The first contour $Z_0(t)$, t=0?T_0

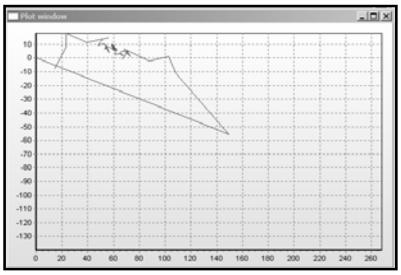

Fig. 7. The Z_0 frequency vectors F_0 (ω)

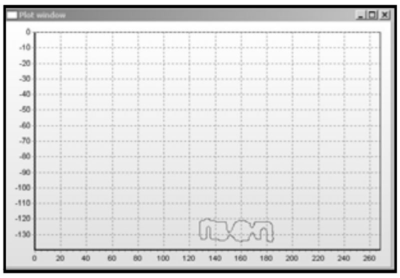

Fig. 6a. The next contour $Z_1(t)$, $t = 0, 1, \ldots, T_1$

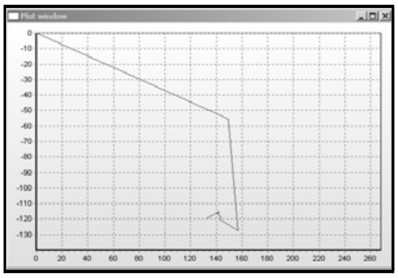

Fig. 7a. The Z_1 frequency vectors F_1 (ω)

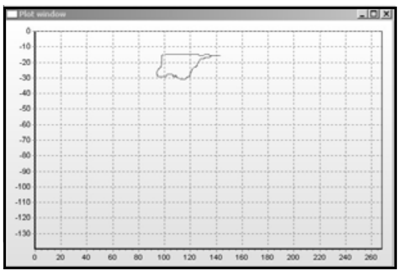

Fig. 6b. The next one $Z_2(t)$, $t = 0, 1, \ldots, T_2$
…and so on to the last contour of the tree.

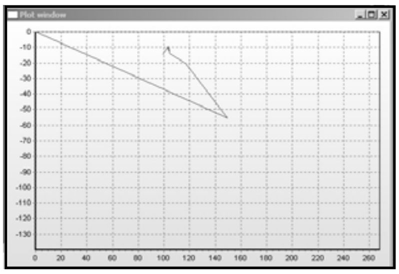

Fig. 7b. The Z_2 frequency vectors F_2(ω)
…and so on.

(A3.3) To assure the rotational invariance, only frequency modules, i.e. the radius-vector lengths $|F(\omega)|$ are used to represent respective contour of the tree, cf. Figs. 8, 8a, 8b.

3.4. Translation, scale and reflection normalization for the contours (A4)

The *SpecialSpectrum()* procedure performs the general image-normalization over the current module spectrum $|F(\omega)|$, namely by three steps.

(A4.1) Centralization of the Cartesian co-ordinates of the image by moving the origin into the point $(x_0, y_0) \equiv x_0+j^*y_0$. The translation vector (x_0, y_0) can be preliminary computed over the binarized image at step A1, for instance as a center of gravity (CoG), either of the whole object, or of its significant contour, etc., cf. also Figs.7, 7a, 7b, where the vectors $F_{(.)}(\omega)$, $\omega = 0$, are (and should be) equal.

(A4.2) Scaling the frequency modules by a predefined coefficient *Norm*. *Norm* can be an appropriate constant, for instance, preliminary calculated for to transform the base contour size into a universal image size denoted here as NORMSIZE. For the experiments here, NORMSIZE is chosen equal to 128.

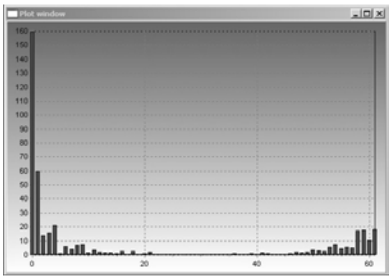

Fig. 8. The Z_0 frequency modula spectrum

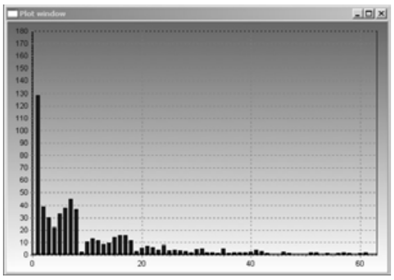

Fig. 9. The Z_0 special spectrum

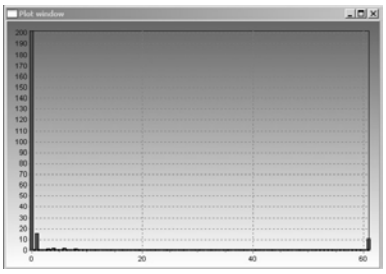

Fig. 8a. The Z_1 frequency modula spectrum

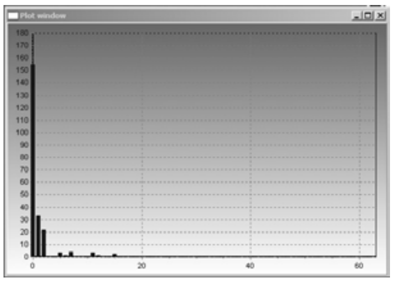

Fig. 9a. The Z_1 special spectrum

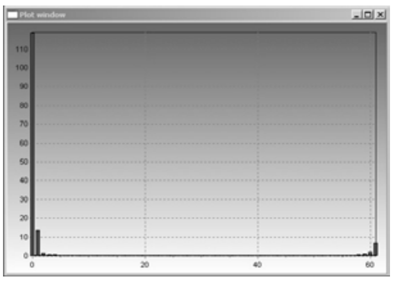

Fig. 8b. The Z_2 frequency modula spectrum
…and so on to the last contour of the tree

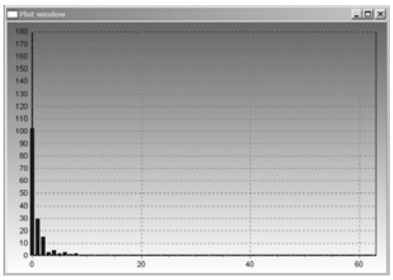

Fig. 9b. The Z_2 special spectrum,
…and so on

(A4.3) A reflection normalization ignoring the encircling direction of contours (clock-wise or v.v.) – by reordering the frequency series regularly from both ends to the middle.

Additionally, the length of each frequency series is shortened using a predefined bottom limit for the essential frequency modula values.

The resulting frequency series is called here *special spectrum* $SF(k)$, $k = 0, 1,\dots, K-1$, of the respective contour, where most often $K<<T$, cf. also Figs. 9, 9a, 9b. Besides, the length K can be still manipulated for additional shortening, so that $SF(k) < \varepsilon$, for each $k > K$, where ε is a small constant chosen for the concrete application.

3.5. Composition of the spectra matrix (A5)

A preliminary ordering of contours by their "importance" can be prepared still at step A2, using already computed contour areas, perimeters, etc. More accurate ordering can be also performed as a simple preliminary step at A3, for evaluating either the "zero"-frequency vectors (i.e. the CoG's, cf. step A4.1) or the first ones (i.e. the radius vectors of the (almost) greatest modules), etc.

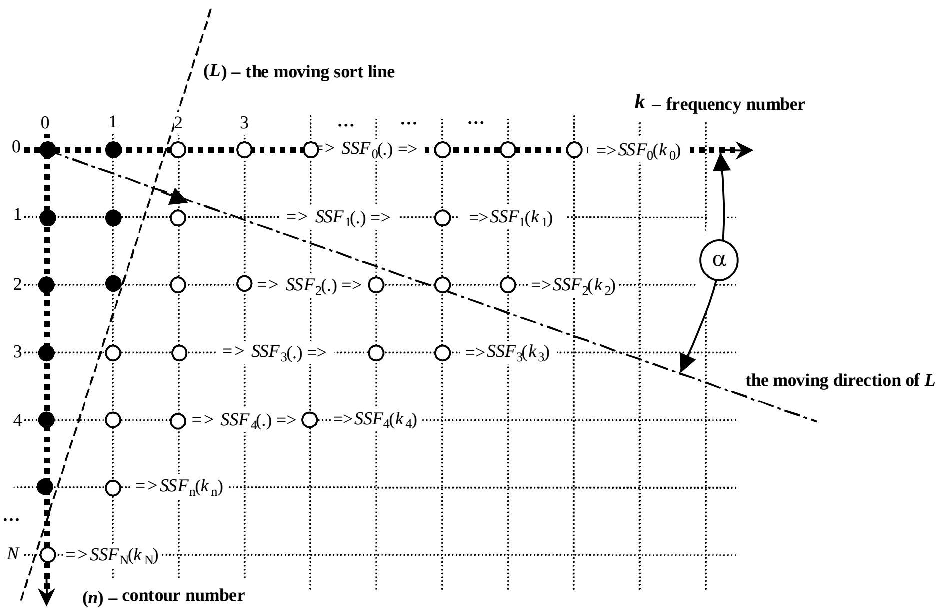

Fig. 10. The super spectra matrix $SSF_{(.)}(.)$, and its sorting scheme producing the IDB key

The chosen way of ordering, namely the "greater" contours at front positions and the "smaller" ones at last places, is used for arranging the contours, i.e. their special spectra, in a matrix called here *super spectra matrix* $SSF_{(.)}(.)$, $SSF_n(k) < \varepsilon$, $n = 0, 1, \dots N$, for each $k > k_n$, where N is the number of all contours, and k_n is the effective spectrum length of the n-th contour. The general presumption for this matrix construction is – the more essential a given contour frequency modulus the closer it should be situated to the matrix origin, cf. also Fig. 10.

3.6. Derivation of the image key items (A6)

A procedure called *CommonSupremeSpectrum()* performs the final reordering in the frames of the super spectra matrix $SSF_{(.)}(.)$ of the previous step A5. Its result is denoted *common supreme spectrum* $CSS(m)$, $m = 0, 1, \ldots, M-1$, where M is the preliminary chosen (and fixed) length of IDB keys.

The idea of *CommonSupremeSpectrum()* is to combine all the $SSF_n(k_n)$ spectrum values, ordering them in a way that well corresponds to our imagination for contours' importance, i.e. the bigger and more sophisticated contours – at front positions. Let us imagine the memory occupied by $SSF_{(.)}(.)$ as a *SSITEM*-array of different length rows, as illustrated in Fig. 10. Let a line L is initially situated as shown here at the beginning $SSF_0(0)$, with a normal angle α, $\alpha \in (0, \pi/2)$, so that all the *SSF*-values to lie on one and the same side of L. Besides, let L is moving to the infinity in the chosen direction of α. Then the order of the *SSF*-values popping-up at the other side of the moving L, will correspond "good enough" to our idea for the image key creation. To be precise it should be also added that the eventual software collisions of simultaneously popping-up of *SSF*-values have to be processed by a similar way.

An optimal value of α can be evaluated for the given IDB by averaging the ordering rule over all the images of IDB. This means that the optimal α will define a triangle of maximal total on this way averaged $SSF_{(.)}(.)$. Experimental results on this rule are illustrated in Fig. 11.

The resulting keys of the illustrative image (cf. Fig.3) for 3 different α can be compared visually in Figs. 11, 11a, 11b, 11c. The choice $\alpha \approx 72°$ is made here for easier quantitative reasoning, i.e. for $\text{ctg}(\alpha)=1:3$.

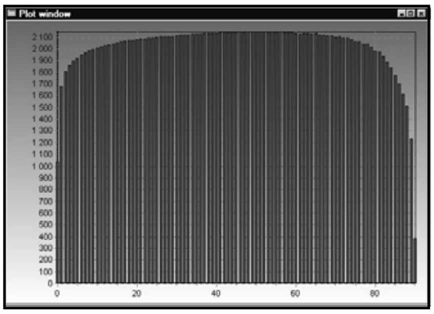

Fig. 11. Optimization of α for test IDB

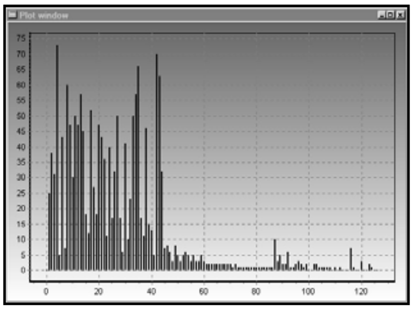

Fig. 11a. The IDB key obtained by $\alpha=0°$.

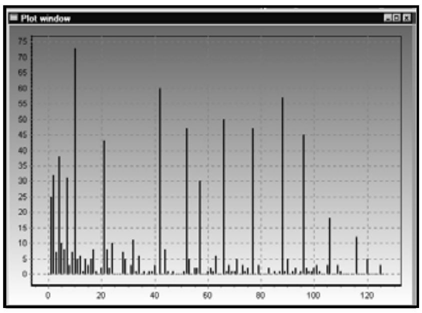

Fig. 11b. The IDB key obtained by $\alpha= 72°$

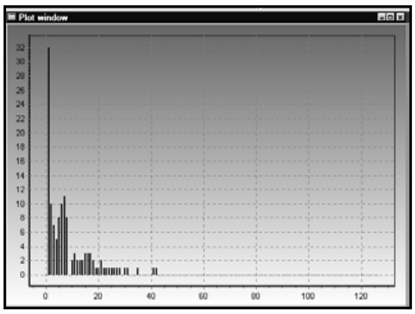

Fig. 11c. The IDB key obtained by $\alpha= 90°$

Finally, it should be said for the image key created in the above way that the size of each key item should correspond to the memory item size used by the respective access method of the DBMS keeping the IDB. For the experiments here, the key item size *SSITEM* is chosen a byte and the key length equals M=510 bytes.

4. Experiments

The proposed method is tested on an IDB of about 4000 images of trademarks. The retrieval time per image is about a few seconds (3-5 s) and could be further diminished.

The Experimental Image Retrieval System (EIRS) developed is written on Borland C Builder v.5.0. Respectively, the DBMS used is Paradox. An IBM compatible PC has been used for development and test – CPU Intel Celeron 330MHz, MM 192MB, HDD 20GB. More details about the EIRS and the test IDB can be found in [26, 27].

Figs. 3-9 and 11 are produced using the tools of EIRS.

5. Discussion of results

The proposed method has been tested for translation, rotation, scaling, reflection and intensity invariance. Randomly chosen images from the IDB have been manipulated in the above sense before passing them to the search engine of EIRS. Accidental misses (no hits) have been met in a few cases, especially when the searched image was manipulated in a way that initial and resulting pixel lattices differ, for instance because of non-integer translation vector, or of rotation angle differing 0mod(π/4), or of non-integer scaling factor, etc.

More exhaustive experiment has been performed to check the rotation invariance for all images of the test IDB. For instance, for a rotation angle of 7°, about 70 of images fails, that is about 1.8 % of all images. By another angle of rotation most of the same images fails again, as could be expected.

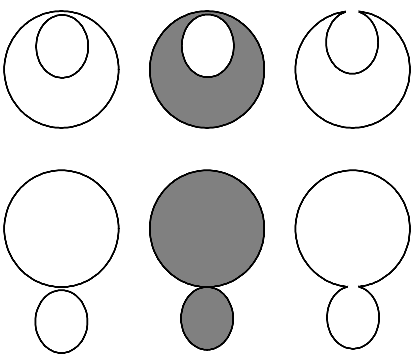

Fig. 12. Possible confusion with a couple of neighboring contours

The reason for these "confusing" experiments can be discovered in the relatively low resolution of the images. Scanning the same images again from the original sources and loading them into the IDB corrects the problem.

The explanation of this phenomenon is simple: there are no problems in search for the original image, but after the test rotation, a small "regular" noise is added to the image. If the image resolution is of low "dpp" (dots-per-pixel) of original scanning, then there is a danger that a couple of image contours will merge in a common contour or v.v., cf. Fig.12. And the last should be fatal for the method.

Fortunately, this phenomenon is easy to escape, scanning the images at higher resolution, especially at the stage of the initial loading of DB with images.

Of course, it would be better if the system evaluates automatically each input image for the above collision. And, this is not a very difficult task that could be included in step A3 of the algorithm.

In a few words, the proposed method deals well with a "regular" noise being in the limits defined simultaneously by the image resolution as well as by the complexity of the image "object". The way of achieving such a *fault tolerance* of the method overlaps this paper frame. It could be said only that the fault tolerance here lies on similar principles already described in [24] for textual DBs.

A disadvantage of the proposed method can be met in that it could not deal with an artificial noise of a great level as well as with a regular noise over the above-mentioned limits.

These remarks orientate the possible implementation of the current method in IDBs of relatively clean images that can be considered graphic images (gray scaled or colored) of a complexity relevant to the available scanner.

6. Conclusion

A fast access method for image retrieval by graphic content has been proposed. The method is based on the image tree of contours defined for graphic images, as well as on the one-dimensional complex Fourier transform for each of the contours. The image content for search is considered the normalized image graphics that should be well localizable into the input picture to search. The most essential image data are represented as a key of fixed length, on which the fast access is performed using the index access methods of a conventional DB management system. The method is experimentally proved.

The method current development can be applied as image search engine for conventional DB of images, for instance in information and image retrieval systems for marks, hallmarks, trademarks, postmarks, as well as for drawings, patterns, stamps, prints, and other types of half-toned or coloured graphics, in the patent offices practice, forensic expertise, industry and/or manufactory area, banking, etc.

At this stage of research and development, the proposed method power is restricted to the set of pictures (images) of well localizable essential graphics (gray scale or coloured) of the picture vs. the picture background. Besides, the essential graphics should not contain any great level of noise, and especially a rough "artificial" noise. The improvement of the method efficiency against such kind of noise will be the matter of near future work due to possible application for the necessities of Bulgarian Patent Office.

References

1. Beaulieu, J. -M., M. Goldberg. Hierarchy in picture segmentation: A stepwise optimization approach. – IEEE Trans. on Patt. Analysis and Mach. Intelligence, Vol. 11, 1989, No 2, 150-163.
2. Chandrinos, K. V., J. Immerker, P. E. Trahanias. ARHON: A multimedia database design for image documents. – In: Proc. of EUSIPCO'1998, 1705-1708.
3. Chan, D. Y. -M., I. King. Genetic algorithm for weights assignment in dissimilarity function for trademark retrieval. – In: Visual Information and Information systems. D. Huijsmans, A. Smeulders (Eds.). – L.N. in Computer Sci., Vol. 1614, Berlin, Springer Verlag, 1999, 557-565.
4. Dimauro, G. Digital transforms in handwriting recognition. – In: Fundamentals in HWR, NATO ASI Series "F". Vol. 124, 1994. S. Impedovo (Ed.). Berlin-Heidelberg, Springer Verlag, 113-146.

5. D u d a, R., P. H a r t. Pattern Classification and Scene Analysis. A Wiley-Interscience Publication, 1973.
6. E a k i n s, J. P. Towards intelligent image retrieval. – Pattern Rec. J., **35**, 2002, 3-14.
7. E a k i n s, J. P., K. S h i e l d s, J. B o a r d m a n. ARTISAN – a shape retrieval system based on boundary family indexing. 12 p.
http://citeseer.nj.nec.com/cs, keyword CBIR
8. F o l e y, J. D., A. v a n D a m. Fundamentals of interactive. – Computer Graphics, Vol. 2, London Amsterdam Sydney. Addison-Westley Publ. Co., 1982.
9. H u l l, J. J. A Database for hand-written text recognition research. – IEEE Trans. on PAMI, Vol.16, 1994, No 5, 550-554.
10. International Hallmarks on Silver, collected by TARDY. Paris, 1981, 429-523.
11. K h o s h a f i a n, S. Object Oriented Databases. John Wiley & Sons, Inc., 1993.
12. K u t i c s, A., N. N i k a j i m a, T. I e k i. An object-based approach to similar image retrieval. – In: IEEE-EURASIP'1999 Workshop on NSIP, Antalya, Turkey, 1999, paper No 161.
13. M a r t i n, J. Computer Database Organisation. Englewood Cliffs, New Jersey, Prentice Hall, Inc., 1975.
14. O z k a r a h a n, E. Database Machines and Database Management. New Jersey, Prentice Hall, Inc., 1986.
15. Official Bulletin of the Bulgarian Patent Office. ISSN 1310-179X, Index 20311, Vol.10, 2001.
16. O t s u, N. A threshold selection method from gray-level histograms. – IEEE Trans. on Systems, Man and Cybernetics, Vol. 9, 1979, No 1, 62-66.
17. P a v l i d i s, T. Algorithms of Comp.Graphics and Image Processing. Comp. Sci. Press, Inc., 1982.
18. R e i s s, T. H. Recognizing Planar Objects Using Invariant Image Features. Lecture Notes in Computer Science. Vol. 676. Berlin, Heidelberg, New York, Springer Verlag, 1993.
19. S m e u l d e r s, A. W. M., M. W o r r i n g, S. S a n t i n i, A. G u p t a, R. J a i n. Content-based image retrieval at the end of the early years. – IEEE Trans. on PAMI, Vol. 22, 2000, No 12, 1349-1380.
20. S t a n c h e v, P. L. Content based image retrieval systems. – In: Proc. of CompSysTech'2001, Sofia, 2001, P.1.1-6.
21.V a s c o n s e l o s, N., M. K u n t. Content-based retrieval from databases: current solutions and future directions. – In: Proceedings of ICIP'2001, Oct. 7-10, 2001, Thessaloniki, Greece Preprint: IEEE, Vol. 3, 6-9.
22. Y o n g R u i, A l f r e d C. S h e, T h o m a s S. H u a n g. Modified fourier descriptors for shape representation – a practical approach. – In: Proc. of First International Workshop on Image Databases and Multi Media Search , 22-23 August, 1996, Amsterdam.
23. Z a n i o l o, C., S. C e r i, C. F a l o u t s o s, R. S n o d g r a s s, V. S u b r a h m a n i a n, R. Z i c a r i. Advanced Database Systems. San Francisco, CA, Morgan Kaufmann Publ., Inc., 1997.
24. D i m o v, D. Natural language names recognition method, fault-tolerant to the most common mistypings. – In: Artificial Intelligence: Methodology, Systems, Applications, A. M. Ramsay, (ed.). Amsterdam, IOS Press, 1996, 320-329.
25. D i m o v, D. T., G. Y. G l u h c h e v. A locally adaptive binary-tree method for binarization of text images. – In: Progress in Handwriting Recognition, A.C. Downton, S. Impedovo, (eds.). London, World Scientific, 1997, 575-580.
26. D i m o v, D., G. G l u h c h e v et al. The computer system STEMB for verification of replicas of the Bulgarian state emblem. – Cybernetics and Information Technologies. Vol. 1, 2001, No 1, 95-106.
27. D i m o v, D. Experimental image retrieval system. Cybernetics and Information Technologies, Vol.2, No 2, 2002, 120-122.
28. D i m o v, D. Fast. Shape-based image retrieval. –In: Proc. of CompSysTech'2003, June 19-20, Sofia, 2003, 3.8.1-7,
http://ecet.ecs.ru.acad.bg/cst/Docs/proceedings/S3/III-8.pdf

Бърз достъп до изображения по графичното съдържание на контурите

Димо Димов

Институт по информационни технологии, 1113 София
E-mail: dtdim@iinf.bas.bg

(Р е з ю м е)

Предлага се метод за бърз достъп по съдържание в база данни (БД) от изображения. Методът се основава на дърво на контурите, дефинирано за графични изображения и едномерна комплексна трансформация на Фурие за всеки контур. Така се опростяват характерните проблеми, свързани с необходимото нормиране на изображенията спрямо транслация, ротация, мащабиране, отражение и интензитет. Съществените графични данни за всяко изображение се подреждат по значимост в ключов низ с фиксирана дължина. Бързият достъп до изображението се организира по този ключ чрез индексните методи за достъп, добре познати от конвенционалните системи за управление на БД (СУБД). Проблемите, свързани с възможния шум в изображенията, също се дискутират в доклада. Предлаганият метод е експериментиран върху БД от около 4000 изображения на фирмени марки.