SIMILARITY ANALYSIS IN MEDICAL IMAGE DATABASES

JACEK RUMIŃSKI

Department of Medical and Ecological Electronics,
Technical University of Gdansk,
Narutowicza 11/12, 80–952 Gdansk, Poland
jwr@biomed.eti.pg.gda.pl

Abstract: The review of methods of similarity analysis of medical images is presented. Feature extraction, feature representation and different concepts of image query algebra problems are described and discussed from the medical application point of view. New algorithms based on medical image regularity description and intensity description are proposed. As a conclusion a Java application "ObrazMed" for content based medical image analysis is presented.

Keywords: image retrieval, image analysis, descriptors, query algebra

1. Introduction

With the rapid advance in multimodal medical imaging and PACS systems, it is increasingly important to retrieve images from medical databases by visual features either locally or over computer networks. Textual annotation of medical images, mainly constructed from diagnostic results, may be a source of mistakes, when made without reference to validated similar cases. The latter are only useful when they contain not only patient records but also images. Images, selected through a query "by example", have been previously described and classified by an expert (medical doctor or computer system) and may be compared with a new case. This requires making new types of queries to a medical database, e.g. retrieve all tumours having a certain location, a certain extent and certain tissue characteristic in males of a certain age. We want to propose a new Content–Based medical Image Query (CBIQ) system, which can be useful to assess the quality of diagnosis. It can also be used, as an intelligent medical assistant, or act as a support tool, in a pure information browser, for guiding a search by content in electronic medical image archives. The following advantages of the proposed system can be achieved:

- an improvement of the quality of diagnosis;
- a provision of intelligent support for diagnosis;
- an interactive training and education;
- an improved usefulness of international medical image databases.

The proposed system is also designed for a medical doctor. A medical doctor, using the software environment (e.g. as a part of a PACS system, or in the internet context) can delineate certain structures, can specify a ROI in an image and with chosen criteria he can ask the system to find the most similar cases in local or world–wide medical image databases. As a result the MD will be able to compare his case with the case description of a number of most similar images. Additionally, by using a Visual Object Catalogue (VOC), some suggestions for diagnosis can be proposed (expert system). Query criteria allow the MD to specify different parameters e.g.: number or range of most similar images to be found, features of Visual Objects (e.g. value, shape, etc.), spatial relations between objects (e.g. tumour within the neighbourhood of the left lung).

In the context of the growing popularity of internet and other telecommunication networks used for remote data consultation, "automatic search engines" will play an important role in retrieving and structuring the information presented to end users. Currently, those search engines have capabilities limited to "search for text key words". The advanced search engines will include capabilities of video and image consultation by content, based on features derived from the images themselves. *An appropriate data structure* allowing the manipulation of image objects should support the needed flexibility and the time limitations in data access. For medical images, a relevant format for the syntactic data representation is DICOM. For the *search by content functionality*, the definition of appropriate features and their syntactic description, as envisaged in MPEG7, is of primary importance. The following achievements of the application are expected:

- verification of diagnosis;
- generating more objective and better diagnosis;
- promoting international collaboration for diagnosis and research purposes;
- integrating databases of medical images;
- improved application of existing databases;
- improved training of medical students and young medical specialists.

To achieve all these objectives the special research have to be perform. The most important problems are: image representation, feature extraction and description, decision methods and query algebra. The query algebra is a fundamental framework for Content Based Image Query system and defines set of possible algorithms.

2. CBIQ queries

The objective of CBIQ is to efficiently find and retrieve those images from a medical database that are similar to the query image. There are two forms of CBIQ [1, 2]:

a) K–nearest neighbour query, which retrieves the K images that are most similar to the query image, and

b) a range query, which retrieves all images that are within a fixed similarity bound, $s > T$, from the query image.

Both forms differ only in the last part of querying process. CBIQ is different from a typical database query in that it uses a similarity search. Based of extracted description of visual features similarity measures are generated by the system. Using similarity measure values queries decide which images should be selected. To perform a specified query from a CBIQ queries algebra the following items should be defined: an image, an image representation, a feature representation and extraction, a feature spatial relations, a region representation and extraction, similarity measures, decision rules. Figure 1 and Figure 2 present new algorithms of medical image data analysis implemented as a CBIQ system.

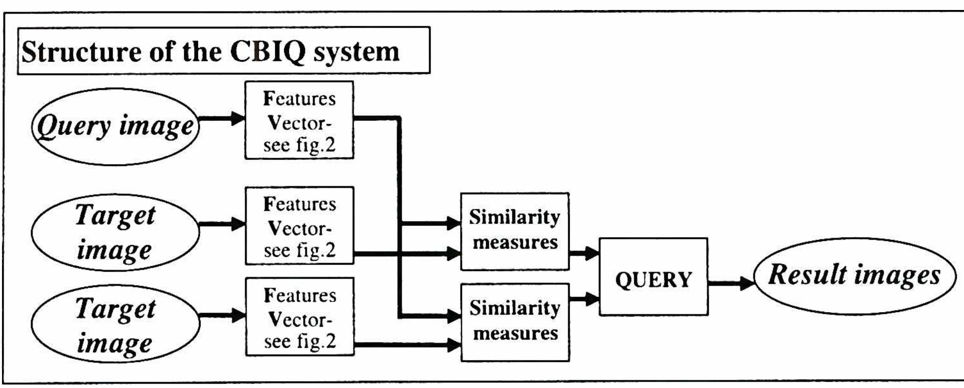

Figure 1. *Structure of the proposed CBIQ system*

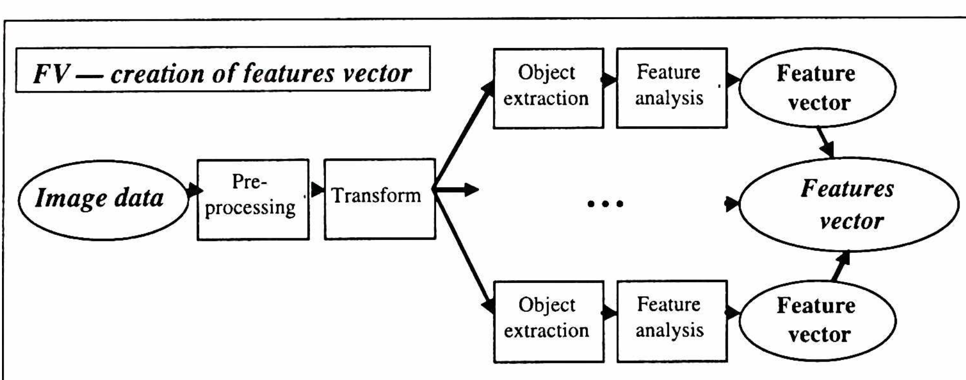

Figure 2. *Creation of the feature vector*

3. Image and feature representation

3.1 Image representation

Understanding of an image representation and description requires proper definition of the image. There are different image definitions. For the goal of this paper we define the image as: a 2–dimensional distribution of the light on the image plane [3]. A digital image is a digital representation of this distribution. Image forming process maybe direct or indirect. Indirect imaging is a process of generating measurement data used further for constructing the image data (e.g. tomography). Direct imaging is a process of direct image data generating. Image presentation is a control of 2–dimensional distribution of light with image data by special hardware (display, printer, etc.). Taking into account only one point from image plane it is possible to measure (absorbed or emitted) radiation wave length. Human perception of this radiation with given wave length is called colour. Human vision system uses three channels to detect radiation. Final result of detected light is a combination of three components. It is possible to create a colour model as a linear, weighted combination of three elementary wave lengths known as primary colours:

$$colour = r \cdot R + g \cdot G + b \cdot B,$$

where *RGB* — primary colours, *rgb* — tristimulus values (weights).
The most popular colour space used today *sRGB* is an example of representation of all possible linear combinations of primary colours defined for given *RGB* standard. In practical implementation in computer systems we operates with matrixes of weights of *RGB* colour model scaled from original (0, 1) to discrete values (0, 255). This means that for *rgb*(0, 0, 0) no radiation will be generate (no colour — black) but for *rgb*(255, 255, 255) maximal radiation with full primary colours will be generated (white). When $r = g = b$ generated colours are shades of grey. Greyscale is composed then by 256 colours. This conclusion is extremely important in case of medical imaging where original image data are usually deep more than 8 bits/element (mainly 12 bits/element and more) so it is impossible to present an image based on original image values. The solution is to scale original pixel values to 8 bits and/or to define a window, which chose 256 subset from full image data set. In case of choosing k levels from n level set it is possible to present $(n + 1) - k$ images (e.g. CT image data have 4096 levels which for 256 levels display gives 3847 images)! Taking into account possible scaling operation of more or less than 256 levels to 256 levels and windowing operation the number of possible images is huge! There is a problem to be solved to proper represent medical image for a CBIQ system. There are no clear solutions in literature for this problem. Only a few papers on a medical CBIQ system have been proposed [4, 5]. None of them deals with this problem because they use only specific medical images e.g. CT lung images — gives fix window width, usually resampled to 8 bits/element. There are two possible solutions for a medical CBIQ system which should be consider:

1. medical image presentation: variable window and scale;
   medical image analysis: original pixel data;
2. medical image presentation: fixed window and scale for given domain;
   medical image analysis: chosen part of original pixel data.

The second solution is hard to implement in multimodality medical image querying because window and scale have to be defined for each domain and each modality. The first solution is much better in terms of analysis but it is hard to see the real image content.

3.2 Image description

In the image formation process objects are describe by a carrier of information e.g. light, X–rays, etc. Task of the computer vision is the reverse process — extraction of objects from the image. The most popular methods are: pattern recognition, image segmentation, a region detection.

3.2.1 Region extraction strategies

Objects (patterns, segments, regions, etc.) are described by feature vectors using image analysis tools. In the CBIQ systems feature vectors are constructed for an entire, original image. In the CBRQ systems regions represent the image objects, which are described by feature vectors. Feature representation is then very important for object extraction (colour segmentation, value segmentation, texture segmentation, etc.) and description in the feature vector.

There are many possible strategies for region extraction in CBRQ [2]. The least complex method involves manual extraction. Images are evaluated by people, and the pertinent information is identified visually. For example, this approach is explored in the IBM QBIC system [6]. However, the manual extraction of regions and objects is extremely tedious and time–consuming for large image and video collections. Several recent techniques have been devised to facilitate the manual region extraction process, such as the use of active contours and snakes [5] to automatically improve the coarse descriptions of region boundaries that are obtained manually.

Alternatively, blocks of regions can be extracted by fixed segmentation of the images. Using this type of segmentation, improved image retrieval performance can be improved. However, in general, it is difficult to pick the scale and locations at which images should be blocked. Furthermore, fixed block segmentation is not invariant to shift or scale changes.

A third technique involves image segmentation. Many techniques have been proposed for segmenting images [1]. Segmentation technique, which uses colour pairs and extended version which perform foreground object colour extraction [7]. Celenk developed an image segmentation technique, which performs a clustering of colours [8]. The process determines simple structures and surfaces. Colour histogram thresholding has been investigated extensively for segmenting colour images. Others investigated one recursive histogram thresholding technique to derive a colour space that is most suited for region segmentation [1].

In order to extract region from images that are used in the spatial and image feature query system, techniques based on histogram back–projection have been proposed [1]. The authors use HBP to detect known objects within images. HBP was later investigated by Ennesser and Medioni for devising a system which solves the "Find Waldo" puzzles by computer [9].

3.2.2 General method used for region extraction [1]

Pattern recognition

Pattern recognition is the process of inferring the class ω_k of an observation h_k^v. There typically exists a finite set of K possible classes $\Omega = \{\omega_0, \omega_1, ..., \omega_{k-1}\}$ where ω_k are the classes and Ω is the set of classes.

In a closed–world assumption, the K classes are mutually exclusive and complete. Associated with each ω_k is a measurement vector h_k^v, which is a collection of observations about class ω_k. In general, the measurement vector h_k^v defines a vector point in the M–dimensional space.

The objective of pattern recognition is to provide the mapping from the h_v's into the ω's through a decision function $d(h_v)$ as follows: $h_v \xrightarrow{d} \omega$.

Typically, pattern recognition is accomplished by computing the distance of the unknown pattern to each class. Then, the unknown pattern is assigned to the nearest class k. This is accomplished as follows, given the unknown pattern with feature measure h^v, allocate h^v to class k if for each $t \neq k$: $Dv(h^v, h_k^v) \leq Dv(h^v, h_t^v)$, where Dv is the measure of dissimilarity of the type v features, (i.e., colour, value, texture).

Pattern recognition is not suited for the problem of extracting regions in a collection of unconstrained medical imagery because:

1. the set of classes must be determined a priori, and has fixed size. For example, when a new pattern class is added, the prior results of a pattern recognition procedure are invalidated;
2. the pattern recognition system is limited to using one feature measurement for all classes. It is not possible to define a texture measure for some classes and a colour measure for others.

Image segmentation

Image segmentation is the process of partitioning an image $I[x, y]$ into non–overlapping segments Sk. Segmentation requires that the set of partitions $\{Sk\}$ is non–overlapping and complete.

Image segmentation does not require that the pattern classes are known. In general, the objective is to separate the image into homogeneous regions. For example, segmentation methods have been developed that perform fixed block segmentation using spatial quad–trees and region merging [1, 4]. Other techniques, such as region growing [10], edge detection [4], texture segmentation [1, 11–14] and colour segmentation [1, 4, 7] have also been developed for segmenting images.

However, segmentation is not a well–defined problem in the case of unconstrained medical images. The pursuit of homogeneous regions is ill–posed for unconstrained images since the definition of the term "homogeneous" is application and domain–dependent. Furthermore, for unconstrained images there is no way of establishing the ground–truth segmentation in order to evaluate the correctness of any given segmentation.

There are multiple ways to choose the partitions to satisfy the constraints of image segmentation and it is hard to decide which is the best.

Region detection

The problem of region detection can be defined as the determination of the existence and location of a model within an image. The problem is formulated as follows: given image $I[x, y]$ and model M, detect the model in $I[x, y]$.

The problem of region detection is related to pattern recognition in that model patterns are used. However, the difference is that there is:

1. no restriction on the number of models or collection of models that are used. The complete set of models does not need to be defined a priori;
2. each model may use a different feature measurement type v. Some model patterns are defined in terms of colour and others in terms of texture.

When a library of models is used to extract regions of differing patterns from an image, the problem resembles that for image segmentation: a set of regions Sk is extracted from the image. However, the difference is that, here, the extracted regions may overlap and may not necessarily completely partition the image.

The most popular methods of region detection are histogram based methods. In the case of images, the probabilistic transition mechanism generates an observation as each point in the image. Measures for these observations in terms of colour/value, texture and shape are presented further in this paper. In this way, the features of regions are represented by histograms. These histograms correspond directly to the transition mechanism probability distributions. In the general two–class detection problem, the histogram for the model hq and for non–model ht is defined. The simple ratio of histograms which formulates the likelihood detection rule is defined as follows:

$$\frac{hq}{ht} > T, \tag{1}$$

where T is the decision threshold.
Different algorithms based on this measure are defined:

a) *a general (confidence) histogram back–projection HBP*

Histogram back–projection is a class of algorithms designed to detect within images the regions with feature–histograms that are similar to a given model histogram.

In Swain and Ballard's formulation, HBP determines the most likely location of a specific histogram within an image $I[x, y]$ [1]. By back–projecting the quotient of the query histogram and the image histogram, the location of the query histogram is found. More specifically, given query histogram hq and image histogram ht, for each level l the new index is calculated: $s[l] = \min(hq[l]/ht[l], 1)$. Then, each point in the image is replaced by the corresponding confidence score, $I'[x, y] = s[k]$, where $k = I[x, y]$. After convolving $I'[x, y]$ with a blurring mask, the location of the peak value corresponds to the most likely location of the model histogram within the image.

In small image retrieval applications, the HBP may be computed at the time of query to find objects within images [1, 15]. However, for large collections it is not feasible to compute the back–projection at time of query. A faster colour indexing method is needed.

b) *quadratic confidence back–projection*

In quadratic confidence HBP hc is generated by taking the ratio of the model histogram hq and target histograms ht as follows:

$$hc[m] = \frac{\sum_n hq[m]A_{m,n}}{ht}, \tag{2}$$

where $A_{m,n}$ defines the similarity of the histogram elements by m and n. Each point in the image is redefined as: if $k = I[x, y]$, then $I'[x, y] = hc[k]$. The new image $I'[x, y]$ is smoothed and thresholded as previously.

c) *binary set back–projection*

In binary set back–projection, the target image points are assigned 1 or 0 depending on membership in the model binary set St (binary sets are described further). In this way, this method is extremely simple as follows:

$$\text{if } k = I[x, y], \text{ then } I'[x, y] = St[k]. \tag{3}$$

The image $I'[x, y]$ is then smoothed to reduce noise and thresholded to reveal the most likely locations of St.

d) *single element quadratic back–projection*

Single element quadratic back–projection combines quadratic confidence HBP and binary set back–projection method. The target image points are weighted by similarity to the back–projection element m, where $St[m] = 1$ and $St[k] = 0$ for each $k \neq m$ such that:

$$\text{if } k = I[x, y], \text{ then } I'[x, y] = A_{m,k}. \tag{4}$$

The image $I'[x, y]$ is then smoothed to reduce noise and thresholded to reveal the most likely locations of St.

e) *local histogramming*

In local histogramming, the smoothing operation and back–projection steps are combined. In this case, for each point in the target image $I[x, y]$ a local neighbourhood histogram hl is computed and its distance to the model histogram hq determines whether each image point belongs to the model. Given region L centred at point (x, y) and the local neighbourhood histogram kl the original image is transformed to: $I'[x, y] = d(ht, hl)$, where d is distance between histograms.

3.3 Region representation

Region of interest (ROI) can be represent in two forms. In the first one — region is a homogenous (in sense of colour, texture, shape,.etc.) part of the image, is usually treated as an object described by a feature vector. The second possibility is that a region is a inhomogeneous part of the image so has to be described by regions in the first sense.

Another aspect of a ROI representation deals with a region size. If the given ROI is not rectangular it can be represent by the Minimum Bounding Rectangle (MBR) area or by filtered MBR area to original size of the region.

3.4 Region description

The step following after object extraction (region extraction) is the characterisation of the regions. The description of a region may include many aspects. For instance:

- radiometry (value, irradiance, tristimulus values);
- texture properties;
- geometry (shape, size, position, orientation);
- relations with other regions.

Often, it is desirable that a parameter describing one aspect is unaffected by other aspects. For instance, shape parameters should be invariant to position, scale, and orientation. Texture parameters should not depend too much on shape. Parameters (features) construct feature vectors, which are used for similarity measure in a query process.

3.5 Feature representation

Radiometry of the image is a measure of information carrier acquired during an image formation process. Well known representations of radiometry is value, colour and intensity. Most applications of a CBIQ system are based on a colour as a query feature. When information carrier is visible light then recorded colour is a property of the object. Databases used to demonstrate different CBIQ systems were usually based on such colour images.

3.5.1 Colour

Colour was defined in section 3.1. The traditional representation of colour in computer system is based on stored tristimulus values of the *RGB* colour space.

Since colour distances in *RGB* colour space do not reflect the actual human perceptual colour distance, many researchers convert and store the image in another colour space. Then most known colour spaces used for CBIQ applications are: Hue Saturation Value, CIE LUV, CIE LAB. Another popular transformations of original *RGB* colours are: colour differences and colour contrast.

Presented further feature representations and transforms do not cover all possible representations. There are only examples used by researchers for CBIQ systems.

Colour transforms

— *CIE LUV colour space [4].* The perceptually uniform CIE LUV colour space can be chosen, such that close distances in the colour space correspond to close distances for the user's perception. This property holds almost everywhere in this space, except for some hues (Red exhibits a larger range than Green and Blue; colours close to grey are poorly discriminated). The computation of a perceptually meaningful colour distance between two generic points in the CIE LUV space, requires to evaluate the length of the shortest path linking the two points. Due to its complexity to increase performance the number of colours has to be reduced to a small set of reference colours Ci. This can be done by clustering method, e.g. *K*–mean algorithm. The number of reference colours should be chosen so that the distance between two generic colours belonging to the same cluster is well approximated by the Euclidean distance. Some experiments [4] show that 128 colours suffice to archive a reasonable compromise between accuracy and computational effort.

— *Hue, Saturation, Value [1, 16].* Hue–saturation–value (HSV) colour space may be treated as a cone: for a given point (h; s; v), h and sv are the angular and radial co–ordinates of the point on a disk of radius v at height v; all co–ordinates range from 0 to 1. Points with small v are black, regardless of their h and s values. The cone representation maps all such points to the apex of the cone, so they are close to one another. The Cartesian co–ordinates of points in the cone, (sv cos(2_h); sv sin(2_h); v), can now be used to find colour differences. This encoding allows us to operationalize the fact that hue differences are meaningless for very small saturations (those near the cone's axis). However, this scheme ignores the fact that for large values and saturations, hue differences are more perceptually relevant than saturation and value differences.

— *Colour differences and intensity [16].*The luminance of a pixel at position (x, y) maybe defined as:

$$L = \frac{3R + 6G + B}{10}, \tag{5}$$

where R, G, B are colours of the pixel at (x, y) scaled in the range [0,1]. Chosen coefficients correspond to the "NTSC" red, green and blue CRT phosphors of 1953 and are standardised in ITU–R Recommendation BT. 601–4 (formerly CCIR Rec. 601). To compute non–linear video *luma* (measure of brightness but it is not CIE luminance) from non–linear red, green and blue:

$$luma = 0{,}299 \cdot R + 0{,}587 \cdot G + 0{,}114 \cdot B. \tag{6}$$

Additionally the following colours difference signals are in use:

$$Cr\text{–}g = 0.5 \cdot (R - G + 1) \tag{7}$$

and

$$Cy\text{–}b = 0.25 \cdot (R + G - 2B + 2). \tag{8}$$

Another proposed solution for intensity and colour differences representation defines the new values at a colour pixel based on the *RGB* values of an original pixel as follows [17]:

$$\begin{aligned} C\,1 &= (R + G + B)/3, \\ C\,2 &= (R + (max - B))/2, \\ C\,3 &= (R + 2 \cdot (max - G) + B)/4. \end{aligned} \tag{9}$$

Here *max* is the maximum possible value for each colour component in the *RGB* colour space. For a standard 24–bit colour image, $max = 255$. Clearly each colour component in the new colour space ranges from 0 to 255 as well. This colour space is similar to the opponent colour axes:

$$\begin{aligned} RG &= \mathrm{R} - 2 \cdot G + B, \\ BY &= -R - G + 2 \cdot B, \\ WB &= R + G + B. \end{aligned} \tag{10}$$

Besides the neurological correlation properties of such an opponent colour space, one important advantage of this alternative space is that the $C\,1$ axis, or the intensity, can be more coarsely sampled than the other two axes on colour correlation.

— *Colour contrast [4].* Colour contrast image may be construct based on original *RGB* colour space in the following way. Given *c* as a colour vector in the *RGB* colour space it is possible to calculate an angle between two colour vectors. Let's the second colour vector, *cm*, be the average colour vector computed in a window of *W* width around the pixel represented by the first colour vector. Then the colour contrast image may be constructed by a set of angles:

$$C(x,y) = \frac{2}{\pi} arc\cos\left(\frac{c(x,y)\cdot cm(x,y)}{|c(x,y)|\cdot|cm(x,y)|}\right), \tag{11}$$

where average colour vector *cm* is:

$$cm(x,y) = \frac{1}{W^2} \sum_{i=x-\frac{W}{2}}^{x+\frac{W}{2}} \sum_{j=y-\frac{W}{2}}^{j+\frac{W}{2}} \left(\frac{c(x-i, y-j)}{|c(x-i, y-j)|} \right). \tag{12}$$

Value

In medical applications colour is not usually an object property (only some colour medical images like those in dermatology, ophthamology). Stored image data are related to information carrier used during the image formation. For example information carrier for CT are X–rays, for thermography infrared radiation, etc. In general medical image stores values which construct image data. Only a few methods of value representations and transform exists including original values, scaled values, value contrast image. It is possible to define value contrast image in similar way as for colour contrast image [3]:

$$C(x, y) = w \cdot (I(x, y) - Im(x, y))/(I(x, y) + Im(x, y)). \tag{13}$$

Common

Presented methods of colour and value representations produced new image matrix data based on original data. Another approach is based on the reduced representation of image data. The most popular methods are:

— *Histogram and first order statistical parameters like mean, expectation, variance, standard deviation, skewness, kurtosis, entropy.*

The histogram of an image is a plot of the colour/value levels versus the number of pixels at that value. The shape of the histogram presents information about the nature of the image, or region if we are considering an object within the image. For example, a very narrow histogram implies a low contrast image, a histogram skewed toward the high end implies a bright image, and a histogram with two major peaks, called bimodal, implies an object that is in contrast with its background.

The well known histogram features are statistically based, where the histogram is used as a model of the probability distribution of the colour/value levels. These statistical features represent information about the characteristics of the colour/value level distribution for the image or region. The first–order histogram probability, $P(g)$, is defined as [3]:

$$P(g) = \frac{V(g)}{N}, \tag{14}$$

where N is the number of pixels; in the image or region (if the entire image or region is under consideration then $N = K \cdot L$ for $K \cdot L$ image), and $V(g)$ is the number of pixels at colour/value level g. As with any probability distribution, all the values for $P(g)$ are less than or equal to 1, and the sum of all the $P(g)$

values is equal to 1. The features based on the first–order histogram probability are the mean, standard deviation (variance), skewness, kurtosis, energy, and entropy.

The mean: is the average value, so it tells something about the general colour/ value of the image and is defined as:

$$\mu = \sum_{g=0}^{L-1} gP(g) = \sum_{row} \sum_{column} \frac{I(row, column)}{N}, \tag{15}$$

where L is the number of colour/value levels.

The second form of the equation, is a sum over the rows and columns corresponding to the pixels in the image or region under consideration.

The standard deviation, σ, which is the square root of the variance, gives information about the contrast. It describes the spread in the data, so a high contrast image will have a high variance, and a low contrast image will have a low variance. It is defined as follows:

$$\sigma = \sqrt{\sum_{g=0}^{L-1} (g-\mu)^2 P(g)}\ . \tag{16}$$

The skewness measures the asymmetry about the mean in the colour value levels distribution. It is defined as:

$$\mu_3 = \frac{1}{\sigma^3} \sum_{g=0}^{L-1} (g-\mu)^3 P(g)\ . \tag{17}$$

There is another method used to calculate skewness in more computationally efficient way:

$$\mu_3' = \frac{\mu - \max(g)}{\sigma}\ . \tag{18}$$

Next feature, kurtosis is a measure of the tail of the histogram; long–tailed histograms correspond to spiky regions. It is defined as:

$$\mu_4 = \frac{1}{4} \sum_{g=0}^{L-1} (g-\mu)^4 P(g) - 3\ . \tag{19}$$

The energy measures of how the colour/value levels are distributed:

$$E = \sum_{g=0}^{L-1} [P(g)]^2\ . \tag{20}$$

The energy measure has a maximum value of 1 for an image with a constant value, and it gets increasingly smaller as the pixel values are distributed across more different level values. The larger the energy value, it is easier to compress the image data. If the energy is high, the number of colour/value levels in the image is few, that is, the distribution is concentrated in only a small number of different colour/value levels.

The entropy is a measure of how many bits are required to code the image data and is given by:

$$H = -\sum_{g=0}^{L-1} P(g)\ln[P(g)]. \tag{21}$$

As the pixel values in the image are distributed among more colour/value levels the entropy increases. This metric tends to vary inversely with the energy.

There are other histogram features (mainly second order histogram features based on a joint probability distribution model) which are defined. However there are used for texture representation and are described further.

Two dimensional histogram has been proposed by researchers for thresholding purposes. The first dimension is traditional colour/value level histogram. The second histogram is constructed as a distribution of average values in given neibourhood of each pixel. 2–D histogram can be described by represented by features and used for the QBIC system. The advantage of such solution is possibility of local distribution comparison.

— *Binary sets*

Binary sets are constructed in two ways. The first method [1] is based on image histogram. In this method the binary set (vector) store 1 for each colour/value position if corresponding histogram value is greater than defined threshold, otherwise 0. The second method construct binary vector in the following way [4]: if a reference colour/value is present in the region, its corresponding entry in the binary vector is set to 1, otherwise 0.

— *Multiresolution representation of the image*

Pyramid transform of images is often in use to increase performance of the query process. Features are constructed as parameters of a low resolution representation of image and histogram parameters of this representation.

Spatial frequency domain

A two dimensional Fourier transform on an image maps spatial data into the spatial frequency domain. In this domain it is possible to extract features, spectral features, where the primary metrics is power:

$$Power = \left|F(u,v)\right|^2, \tag{22}$$

where $F(u,v)$ is the Fourier transform of $I(c,r)$.

Calculation of power in specific part (region) of the image can be used as a s–f feature of the image, however there were no reports found on applications of spectral power and other Fourier descriptors (mainly shape descriptors) in CBIQ systems.

Spatial — spatial frequency domain

The wavelet transform with different basis functions is used to present the image in the s–s/f domain. The most popular wavelets used for CBIQ systems are Haar's basis and Daubechies' basis. Version of Haar's construction can be written as follows [18]:

the mother wavelet:

$$\psi(x) = \begin{cases} 1, & x \in [0,0.5) \\ -1, & x \in [0.5,1) \\ 0, & otherwise \end{cases}, \tag{23}$$

the scaling function:

$$\phi(x) = \begin{cases} 1, & x \in [0,1) \\ \\ 0, & otherwise \end{cases}. \tag{24}$$

Daubechies' basis is defined as:

the mother wavelet:

$$\psi(x) = 2^{1/2} \sum_k g(k)\,\phi(2\ x - k), \tag{25}$$

the scaling function:

$$\phi(x) = 2^{1/2} \sum_k h(k)\,\phi(2\ x - k), \tag{26}$$

and

$$g(k) = (-1)^k h(1-k). \tag{27}$$

Low pass filter coefficients $h(k)$ have to satisfy following conditions: orthogonality, normality and regularity.

After wavelet transform given by:

$$W_{m,n} = \int_{-\infty}^{+\infty} f(x)\,\psi_{m,n}(x)dx, \tag{28}$$

where m, n are integers and

$$\psi_{m,n}(x) = 2^{-m/2}\psi(2^{-m}x - n), \tag{29}$$

is a family of real orthonormal bases obtained through translation and dilation of the mother wavelet; multiresolution s–s/f representation of image (region) is created. Created set of subimages with the resolution reduced by a factor of 2 from original up to given level m, represent different frequency bands of the image. The lower frequency bands in the wavelet transform usually represent object configurations in the images; the higher frequency bands represent texture and local colour/value variation. Values in the particular bands (subimages) are used as feature sets. Additionally histogram features are used to represent these subimages for the fast comparison between the template (query object) and the target.

3.5.2 Texture

Texture is a well–researched property of image regions, and many texture descriptors have been proposed, based on co–occurrence matrix, Fourier transform domain, random field models, local linear transforms and multiresolution methods.

Whereas colour is a point property, texture is a local–neighbourhood property. It does not make sense to talk about the texture of zebra stripes at a particular pixel without specifying a neighbourhood around that pixel. In order for a texture descriptor to be useful, it must provide an adequate description of the underlying texture parameters and it must be computed in a neighbourhood, which is appropriate to the local structure being described.

Local (directional) differences methods of texture representation

It is possibly to describe texture region by histogram features. Some histogram feature, were described earlier, however they cannot express spatial texture characteristics.

The first set of histogram–based texture features is construct on colour/value co–occurrence matrices. Let $p(k, l, d)$ denote the joint probability of two pixels k and l lying at distance d in the image. This probability can be easily estimated from an image by counting the number Nkl of occurrences of the pixel values (k, l) lying at distance d in the image. Let n be the total number of any possibly joint pairs lying at distance d in the image. Then co–occurrence matrix elements are given by [3]:

$$c_{kl} = p(k, l, d) = \frac{Nkl}{n} d \ . \tag{30}$$

The co–occurrence matrix C has dimension $L \cdot L$, where L is the number of colours/values in the image. Co–occurrence matrices carry very useful information about spatial texture organisation. If the texture is coarse, their mass tends to be concentrated around the main diagonal, i.e at element $-c_{kl}$, $|k - l| < t$. If the texture is fine, co–occurrence matrix values are much more spread. If texture carries strong directional information, the co–occurrence matrices C tend to have their mass in the main diagonal, for displacement vector d corresponding to the texture direction. Several texture descriptors have been proposed to characterise texture with the co–occurrence matrix:

a) correlation

$$R = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (k)(l) c_{kl} \; ; \tag{31}$$

b) covariance

$$C = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (k-\mu)(l-\mu) c_{kl} \; ; \tag{32}$$

c) normalised covariance (correlation coefficient)

$$r = \frac{1}{\sigma^2} \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (k-\mu)(l-\mu) c_{kl} = \frac{C}{\sigma^2} \; ; \tag{33}$$

d) energy

$$E = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (c_{kl})^2 \; ; \tag{34}$$

e) entropy

$$H = -\sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (c_{kl}) \ln(c_{kl}) \; ; \tag{35}$$

f) inertia

$$I = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} (k-l)^2 c_{kl} \; ; \tag{36}$$

g) local homogeneity

$$L = \sum_{k=0}^{L-1} \sum_{l=0}^{L-1} \frac{1}{1+(k-l)^2} (c_{kl}) \; ; \tag{37}$$

h) maximum probability

$$p = \max_{k,l}(c_{kl}) \, . \tag{38}$$

Another set of histogram–based texture features is construct on colour/value differences. Let $d = (d1, d2)$ be the displacement vector between two images pixels, and $g(d)$ the colour/value level difference at distance d:

$$g(d) = | I(r,c) - I(r + d1, c + d2) | \, . \tag{39}$$

Only one histogram $p(g, d)$ exists for each distance d. It contains valuable information about the spatial organisation of image colours/values. If an image region has coarse texture, the histogram $p(g, d)$ tends to concentrate around $g = 0$ for small displacements d. If the region texture is thin, $p(g, d)$ tends to spread, even for small displacement vectors d that exceed the texture grain size. Several texture measures can be extracted from histogram colour/value level differences. Some of them are identical as previously described for colour/value histograms features with $P(g) = p(g, d)$. Additionally two measures are defined:

a) contrast

$$CON = \sum_{g=0}^{L-1} g^2[p(g,d)]; \tag{40}$$

b) inverse difference moment

$$IDM = \sum_{g=0}^{L-1} \frac{1}{1+g^2}[p(g,d)]. \tag{41}$$

The third set of histogram–based texture features is construct on colour/value run length statistics. A run length l of pixels having equal intensity f in a direction θ is an event denoted by (l, f, θ). The run lengths reveal both directionality and coarse-ness of image texture. Coarse textures tend to produce long runs at specific direction θ. Usually four colour/value level run length matrices for $\theta = 0°, 45°, 90°, 135°$, are computed for each image (region). Let $N(l, f, \theta)$ denote the number of events (l, f, θ) in an image. Let Nr be the total number of existing runs and let Tr denote the double sum:

$$Tr = \sum_{k=0}^{L-1} \sum_{l=0}^{Nr-1} N(l, f_k, \theta). \tag{42}$$

The ratio $r(l, f, \theta) = N(l, f, \theta)/Tr$ is the histogram of the colour/value runs at a specific direction θ. The following features can be calculated from the colour/values run lengths:

a) short–run emphasis

$$RF_1 = \sum_{k=0}^{L-1} \sum_{l=0}^{Nr-1} \frac{r(l, f_k, \theta)}{l^2}; \tag{43}$$

b) long–run emphasis

$$RF_2 = \sum_{k=0}^{L-1} \sum_{l=0}^{Nr-1} l^2 r(l, f_k, \theta); \tag{44}$$

c) colour/value distribution

$$RF_3 = \sum_{k=0}^{L-1} \left[\sum_{l=0}^{Nr-1} r(l, f_k, \theta) \right]^2; \tag{45}$$

d) run–length distribution

$$RF_4 = \sum_{l=0}^{Nr-1}\left[\sum_{k=0}^{L-1} r(l, f_k, \theta)\right]^2; \tag{46}$$

e) run percentages

$$RF_5 = \frac{Tk}{Tp}\sum_{k=0}^{L-1}\sum_{l=0}^{Nr-1} r(l, f_k, \theta), \tag{47}$$

where Tp is the number of pixels in the picture.

Gradient–based methods [16]

The texture descriptors for this method arise from the windowed second moment matrix derived from the gradient ΔI of the image intensity. The gradient is computed using the first difference approximation along each dimension. This operation is often accompanied by smoothing.

Defined scale σ controls the size of the integration window around each pixel within which the outer product of the gradient vectors is averaged and is define to be the width of the Gaussian window within which the gradient vectors of the image are pooled. The second moment matrix for the vectors within this window, computed about each pixel in the image, can be approximated using

$$M_\sigma(x, y) = G_\sigma(x, y) \cdot (\nabla I)(\nabla I)^T, \tag{48}$$

where $G_\sigma(x, y)$ is a separable binomial approximation to a Gaussian smoothing kernel with variance σ^2.

At each pixel location, $M_\sigma(x, y)$ is a 2×2 symmetric positive semidefinite matrix; thus it provides us with three pieces of information about each pixel. Rather than work with the raw entries in $M_\sigma(x, y)$, it is more common to deal with its eigenstructure. Consider a fixed scale and pixel location, let $\lambda 1$ and $\lambda 2$ ($\lambda 1 \geq \lambda 2$) denote the eigenvalues of $M_\sigma(x, y)$ at that location, and let ϕ denote the argument of the principal eigenvector. When $\lambda 1$ is large compared to $\lambda 2$, the local neighbourhood possesses a dominant orientation, as specified by ϕ. When the eigenvalues are comparable, there is no preferred orientation, and when both eigenvalues are negligible, the local neighbourhood is approximately constant.

In order to select the scale at which $M_\sigma(x, y)$ is computed, i.e. to determine the function $\sigma(x, y)$, a local image property known as the *polarity* can be used. The polarity is a measure of the extent to which the gradient vectors in a certain neighbourhood all point in the same direction. The polarity at a given pixel is computed with respect to the dominant orientation ϕ in the neighbourhood of that pixel. For ease of notation, a fixed scale and pixel is used. The polarity can be define as:

$$p = \frac{|E_+ - E_-|}{E_+ + E_-}, \tag{49}$$

The definitions of E_+ and E_- are:

$$E_+ = \sum_{(x,y)\in\Omega} G_\sigma(x,y)[\nabla I \cdot \bar{n}]_+ , \tag{50}$$

and

$$E_- = \sum_{(x,y)\in\Omega} G_\sigma(x,y)[\nabla I \cdot \bar{n}]_- , \tag{51}$$

where $[q]_+$ and $[q]_-$ are parts of their argument, n is a unit vector perpendicular to ϕ, and Ω represents the neighbourhood under consideration.

After selecting scale for each pixel, three texture descriptors are assigned to each pixel. The first is polarity p, the second anisotropy $a = 1-\lambda 2/\lambda 1$ and the third normalised texture contrast $c = 2\sqrt{\lambda 1 + \lambda 2}$.

Spatial–Spatial frequency representation of texture [1]

Often texture regions are represented by a set of multiresolution subimages generated by the wavelet transform. As it was described previously values and quntizied values of subimages are used as feature vectors. Constructed histograms and calculated histograms features are also used as texture features.

3.5.3 Geometry (shape, size, location, orientation)

Object shape is usually described by two groups of method: boundary (contour) description and statistical moments.

In the first group shape boundary may be represented as an ordered set of boundary interest points. In case where the shape is polygonal, the interested points would be chosen as the vertices on the polygon. If curved shapes are present, then points on the boundary where maximal local curvature occurs would be chosen. A variety of techniques exist for the robust identification of interest points including edge detectors, multiresolution detectors, and other. Points vector construct feature vector which is used for query process.

Considering the image plane as a complex plane consisting of a real axis and an imaginary axis: $z = x + jy$, contour function can be presented as $z(s) = z(s + P)$, where s is the running arc length relative to the start point ($x0$, $y0$) and P is the perimeter of the region. Then by inverse Fourier transform the complex amplitudes Zk of harmonic functions with frequencies that are multiples of $1/P$ are calculated. These harmonics are Fourier descriptors (FDs) and are used to construct feature vector [3].

Most applications of the CBIQ systems used family of parameters called moments to represent shape properties. The moments of order $p+q$ of region represented by the bitmap b are [3]:

$$M_{p,q} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} n^p m^q b_{n,m} . \tag{52}$$

The first order moments and are related to the balance point (x', y') of the region: $x' = M_{1,0} / M_{0,0}$ and $y' = M_{0,1} / M_{0,0}$. The point (x', y') is called the centre of gravity or centroid. In order to make the description independent on position, moments can be calculated with respect to the centroid. The results are the so–called central moments:

$$\mu_{p,q} = \sum_{n=0}^{N-1} \sum_{m=0}^{M-1} (n - x')^p (m - y')^q b_{n,m}. \tag{53}$$

Usually a set of moments and parameters based on them (e.g. principal moments, orientation of the region, eccentricity, etc.) are used to construct feature vector for the query process.

Other geometry based parameters, which are often used in the QBIC applications are:

a) point locations (e.g., point [spot] detectors, multiresolution contrast based point detectors, etc.);
b) line location, length, an angle between lines (e.g., edge detectors);
c) polygon centroid (as location), area and perimeter (e.g., FDs, moments, etc.);
d) object to object relation (e.g., distance between region centroids);
e) bounded location i.e. location of object in given part of the image.

4. Similarity measure and retrieval effectiveness

4.1 Similarity measures

In order to compare feature vectors different similarity measures are defined. The main task of this process is to sort all possible feature vectors (for all features and for all targets) according to the similarity measure value. Later, according to defined query, the defined number or set of values is chosen. Chosen feature vector is identified with the target region or image by unique index (or name, number identifier, URL, etc.).

Let Vq be the query feature vector and Vt the target feature vector. Then the following set of distance measures (minimal distance → maximal similarity) can be defined [1, 3, 4]:

1. *Minkowski–form distance*

   The first class of dissimilarity measures is based upon the Minkowski–form distance metric:

$$d_{q,t}^{r} = \sum_{m=0}^{M-1} | Vq[m] - Vt[m] |^{r} . \tag{54}$$

   where M is the number of features in feature vector, and r is an integer number. When r equals 2 this Minkowski–form metric is known as Euclidean distance. When $r = 1$ and $|q|$ is absolut value of q then this Minkowski–form metric is known as city–block distance.

2. *Chessboard distance*

   This metric looks for a maximal difference value between vector elements:

$$d_{q,t} = \max_m (| Vq[m] - Vt[m] |) . \tag{55}$$

3. *Vector intersection*

   The vector intersection is the application of Minkowski–form distance. When the query feature vector (q) size is less than the target feature vector (t) size then the intersection of vectors Vq and Vt can be defined as:

$$d_{q,t} = 1 - \frac{\sum_{m=0}^{M-1} \min(Vq[m], Vt[m])}{| Vq |}, \tag{56}$$

   or in symmetric in vectors form:

$$d_{q,t} = 1 - \frac{\sum_{m=0}^{M-1} \min(Vq[m], Vt[m])}{\min(| Vq |, | Vt |)}, \tag{57}$$

   where $|Vq|$ is the sum of vector element. This kind of distance measure is especially used when the feature vector is the feature histogram. When a size of vector Vq and Vt are equal then distance is defined by Minkowski–form formula.

4. *Hamming distance*

$$d_{q,t} = \frac{\sum_{m=0}^{M-1} (Vq[m] - Vt[m])}{\sum_{m=0}^{M-1} (Vq[m]) \sum_{m=0}^{M-1} (Vt[m])} . \tag{58}$$

5. *Quadratic distance measure*

   A quadratic–form distance metric is used in the IBM QBIC system for colour histogram–based image retrieval [6]. It is reported that quadratic–form distance metric between colour histograms provides more desirable results than "like–bin" only comparisons between colour histograms. The quadratic form distance is given by:

$$d_{q,t} = (Vq - Vt)^T \mathrm{A} (Vq - Vt), \tag{59}$$

   where $\mathrm{A} = [a_{i,j}]$ and $a_{i,j}$ denotes similarity between elements with indexes i and j. The quadratic metric distance is a true–distance metric when $a_{i,j} = a_{j,i}$ (symmetry) and $a_{i,i} = 1$.

6. *Mahalanobis distance*

$$d_{q,t} = (Vq - Vt)^T C^{-1}(Vq - Vt), \tag{60}$$

where C is the covariance matrix. Computation of this metric is faster when covariance matrix is diagonal.

7. *Correlation coefficient*

The correlation coefficient is independent of both amplitude fluctuations and baseline changes, and produces an output between −1 and 1. Mathematically, the correlation coefficient is defined as:

$$\rho = \frac{\sum_{m=0}^{M-1}(Vq[m] - V'q)(Vt[m] - V't)}{\sqrt{\sum_{m=0}^{M-1}(Vq[m] - V'q)^2}\sqrt{\sum_{m=1}^{M-1}(Vt[m] - V't)^2}}, \tag{61}$$

where $V'q$ and $V't$ are mean values of Vq and Vt.

In proposed algorithms we used an Euclidean distance measure as it is fast and operates on all feature vectors.

4.2 Retrieval effectiveness

It is important to decide which query method is the best and for which conditions. Retrieval effectiveness can be estimated by two metrics: recall and precision. Recall signifies the proportion of relevant images in the entire database that are retrieved in the query. Precision is the proportion of the retrieved images that are relevant to the query. More precisely, let A be the set of relevant items, let B be the set of retrieved items, and a, b, c and d are given as follows.

— a = retrieved and relevant (detection);
— b = retrieved and not relevant (false alarm);
— c = not retrieved and relevant (miss);
— d = not retrieved and not relevant.

Results a, b, c, d are well know in medical applications like a set of indexes: True Positive, True Negative, False Positive and False Negative.

Recall and *precision* are defined by the following conditional probabilities:

$$recall = P\,(B|A) = a/(a + c), \tag{62}$$

$$precision = P\,(A|B) = a/(a + b). \tag{63}$$

The measures of recall and precision require that the relevance to the query of each item in the database is established ahead of time. For example this can be done for each query by subjectively assigning one of three values to each image in the database: relevant = 1, partially or marginally relevant = 0:5 and not relevant = 0.

The relevance value corresponds to the probability that the item is relevant to the query.

5. Implementation

Most of methods presented in this paper where implemented in our application [21]. With Java programming language the modern medical image processing and analysis system was created, which enable to operate on original medical images (original values). The user can point data space with Uniform Resource Locator (URL), which can be a file, a directory, a database (by JDBC) or a DICOMDIR file. Our Java application — "ObrazMed" — support a user with a GUI for interactive definition of a ROI in a query image. Defined ROI is used as a template subimage for a differential analysis with every target image. Algorithms operate on the original or modified (according to different representations of medical images e.g. contrast image, morphological laplacian, etc.) pixel values. As a result of operations a list of similarity indexes is created. The system user has to define, a priori, a threshold value to choose range of similar images. All algorithms are grouped into three classes: original pixel value based, region based and texture based. For each algorithm appropriate representation of a medical image is created. Average time consumed for an image features calculation (made once for image in the database) and query is less that one second for Sun Ultra 30 (128MB RAM). These preliminary results were obtained with database with more that 200 images (this small database is under continuous development) with the recall about 0.9 and

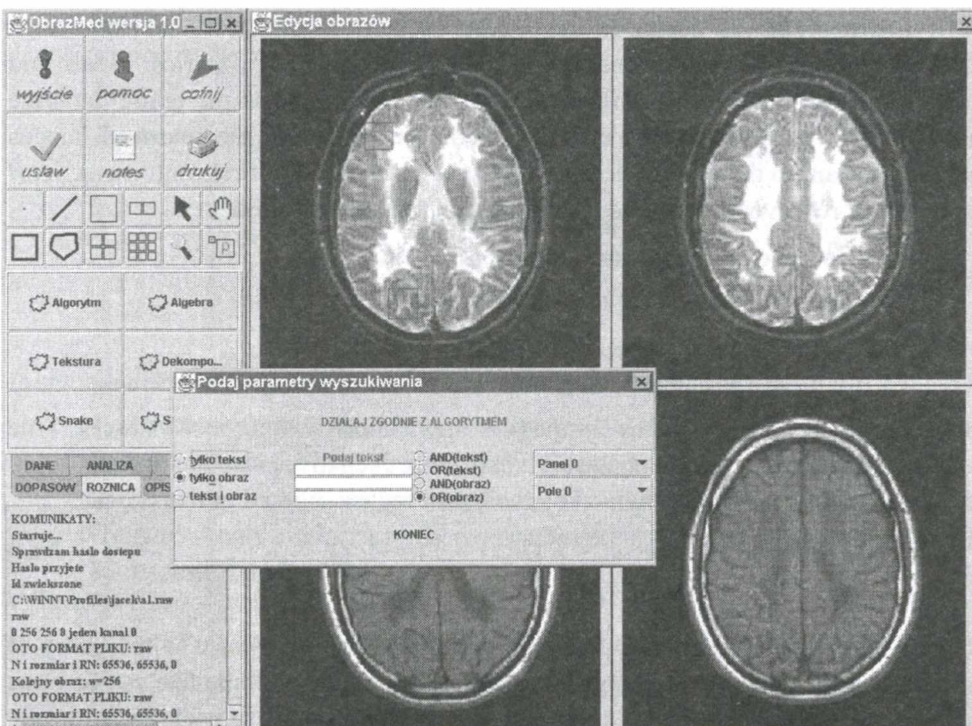

Figure 3. *Screen–shot of the ObrazMed Java application for content based medical image retrieval*

the precision grater than 0.8. In the near future spatial queries and wavelet multiscales representations will be implemented in the system.

Content–based image search is time consuming so some modifications have to be performed to increase effectiveness. Proposed solutions introduce the pyramid method of image processing (to reduce number of candidates for the most time consuming operations) and additionally text search (based on text in description modules of DICOM information objects). Other functionality and features of this new medical image analysis system are:

— RAW, GIF, JPG format support;
— image filtering with convolution masks;
— morphological operators;
— graphical overlays;
— histogram presentation and operations;
— different colour look–up tables;
— enhanced image algebra (special command window);
— multi–images presentation in the one window;
— image scaling.

References

[1] J.R. Smith, *Integrated Spatial and feature image systems: retrieval, analysis and compression*, PhD dissertation, Columbia University 1997

[2] N. Nes and M. Kersten, *The Acoi Algebra: A Query algebra for image retrieval systems*, BNCOD'16, cardiff, 77–88, July 1998

[3] F. van der Heijden, *Image–Based Measurement Systems*, Wiley&Sons 1995

[4] E. Vicario ed., *Image Description and Retrieval*, Plenum Press 1998

[5] ASSERT system — *http://rvl1.ecn.purdue.edu/v1/CBIR/CBIRASSERT.html*

[6] M. Flickner, H.S. Sawhney, J. Ashley, Q. Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, *Query by image and video content: The QBIC system*, IEEE Computer 28(9), 23–32, 1995

[7] M. Celenk, *A color clustering technique for image segmentation*, Computer Vision, Graphics and Image Processing (CVGIP), vol. 52 1990

[8] T.–S. Chua, S.–K. Lim and H.–K. Pung, *Content–based retrieval of segmented images*, Proc. ACM Intern. Conf. Multimedia, October 1994

[9] S. Cooper, *Find Waldo Now*, Little, Brown and Company, 1988

[10] J.M.H. Du Buf, *Abstract process in texture discrimination*, Spatial Vision, vol. 6 1992

[11] J.M.H. Du Buf, M. Kardan and W. Spann, *Texture feature performance for image segmentation*, Pattern Recog., 23(3/4), 1990

[12] B.B. Chaudhari and N. Sarkar, *Texture segmentation using fractal dimension*, IEEE Trans. Pattern Anal. Machine Intell., 17(1), January 1995

[13] R. Chellappa, R.L. Kashyap and B.S. Manjunath, *Model–based texture segmentation and classification*, in: C.H. Chen, L.F. Pau and P.S.P. Wang, editors, Handbook of Pattern Recognition and Computer Vision, 277–310, World Scientific Publishing Company, River Edge, NJ, 1993

[14] D. Dunn and W.E. Higgins, *Optimal gabor filters for texture segmentation*, IEEE Trans. Image Processing, 4(7), July 1995

[15] F. Ennesser and G. Medioni, *Finding waldo, or focus of attention using local color information*, IEEE Trans. Pattern Anal. Machine Intell., 17(8), August 1995

[16] C. Carson, S. Belongie, H. Greenspan and J. Malik, *Region–Based Image Quering*, CVPR'97 IEEE Workshop on Content–Based Access of Image and Video Libraries, Puerto Rico, 1997

[17] N.W. Campbell, W.P.J. Mackeown, B.T. Thomas and T. Troscianko, *Interpreting image databases by region classification*, Pattern Recognition (Special Edition on Image Databases) vol. 30, no. 4, April 1997

[18] J.Z. Wang, G. Wiederhold, O. Firschein and S.X. Wei, *Content–based Image indexing and searching using Daubechies' wavelets*, International Journal on Digital Libraries, 1(4), 311–328, Springer–Verlag 1998

[19] A. Gupta, *Visual information retrieval: A virage perspective*, Technical report, Virage, San Mateo, CA, 1995

[20] A. Pentland, R.W. Picard, and S. Sclaroff, *Photobook: Content Based Manipulation of Image Databases*, chapter 2, 43–75, Kluwer, Academic publishers, 1996

[21] J. Rumiński, *Content based medical image analysis*, Proceedings of Computer Assisted Radiology and Surgery Congress, International Congress Series, Elsevier 1999