Transactions on Machine Learning
and Data Mining
Vol. 2, No 1 (2009) 3-22
© ISSN: 1865-6781 (Journal),
ISBN: 978-3-940501-11-0,
IBaI Publishing ISSN 1864-9734

ibai Publishing
www.ibai-publishing.org

Content-Based Image Indexing and Retrieval in an Image Database for Technical Domains

Mark Ewald

Institute of Computer Vision and Applied Computer Sciences
Kohlenstr. 2, 04109 Leipzig, Germany

Abstract. The availability of a variety of sophisticated data acquisition instruments has resulted in large repositories of imagery data in different applications like non-destructive testing, technical drawing, medicine, museums and so one. Effective extraction of visual features and contents is needed to provide meaningful index of and access to visual data. In this paper, we propose an image database architecture which can be used for most industrial problems. The image database is able to handle structural representation of images. Indexing is possible object based, spatial relation based, and by a combination of both. The query can be a textual query or a image content based query. We propose how the image query is processed, how similarity based retrieval is performed over images and how the image database is organized. Results are presented based on an application of ultra sonic images from non-destructive testing.

Keywords: Image Database, Query-by-Image-Content, Structural Similarity Measure, Indexing, Learning

1 Introduction

The availability of a variety of sophisticated data acquisition instruments has resulted in large repositories of imagery data in different applications like non-destructive testing, technical drawing, medicine, museums and so on. Effective extraction of visual features and contents is needed to provide a meaningful index of and access to visual data[1]. Most existing approaches to image indexing and retrieval use the textual keyword [2][3]. Search and retrieval are performed on the keyword records and the associated images are retrieved after the textual search is complete. The main drawback of such an approach is that different users may have different interpretations for one image content and therefore, they may use different words to describe the image.

The vocabulary used in describing image contents is usually domain-dependent since definition of standard vocabulary and taxonomy for the different information contained in an image like texture, shape and spatial location is still a research topic [4], which is not considered away from image processing

All this has brought to the attention of researchers the concept of Query-by-Image Content (QBIC) [5]. The QBIC concept allows to index and retrieve an image by image content rather than textual keywords and makes the system more independent by the drawbacks mentioned above.

Hildebrandt [6] identifies three different types of QBIC: spatial index based method, feature based indexing methods, object based methods and pictorial based methods. In spatial index based method is used the location of an object or a scene whereas feature based methods use information like color, texture or statistic of a scenic in the image or the whole image[7][8]. Object based indexing methods use information like color, texture, location and shape of an object [9]. Pictorial based indexing methods use a symbolic image which is an array representing the spatial relations among objects [10]. Besides that, there are also used image keys for indexing image databases[11].

To enable such queries it is necessary to combine methods from pattern recognition (to detect and represent the content based features) and database techniques (to efficiently index and retrieve the relevant images based on those features) as well as learning capabilities to include new data entries.

As a complement to text based methods, the content-by-image methods that is (1) posed visually by giving a sample image and finding others similar to it; (2) based on computed features like gray level, texture, shape and spatial location, and not exclusively on keywords or text; and (3) based on similarity as opposed to exact match so that a set of “like” or similar images can be returned and ranked accordingly[9].

In this paper, we describe an approach which combines both concepts, the concept of textual queries with query-by-image content, for image indexing and retrieval. The approach allows an user to use both types of queries: to index by textual description if no proper images for indexing are available and to index by image content if the content based query can be processed from an available image. For the textual keywords is the user provided by the system with standard vocabulary for shape, size, gray level and spatial location, which he can inputted to the system via a keyword mask. The same semantic content can be processed from the image by image processing and signal-symbol transformation unit. This ensures high flexibility, completeness and consistency in the textual description and the usage of the same algorithm for similarity determination and image retrieval.

The signal-symbol transformation unit and the feature extraction unit is generic. The image processing unit in the system uses domain dependent algorithm. This requires that the user specify the domain he is considering before indexing and retrieval of images, but allows automatic processing of image queries.

The system considers two main data types: objects and scenes. Objects are described by their shape, size, location and gray level. A scene comprises of objects and their spatial relation to each other which makes up a high-level description of an image. The high-level description is a structural representation of the image realized as attributed graph.

Such a description is sufficient for most technical domains like mechanical objects, defect images of welding seams, ultra sonic images of vessels, fingerprint images [12] and technical drawings [13] .

In Section 2, we describe the overall architecture of our image database. The content based query features are described in Section 3. The similarity measure as well as the algorithm used for similarity are presented in Section 4. Section 5 deals with the index structure. Retrieval is presented in Section 7. Finally, we show our results in Section 8 and give conclusions in Section 9.

2 Overview of Image Database

Our image database mainly consists of 5 functional units (see Figure 1):

- the image database itself with the index structure,
- the image processing and interpretation unit with the image query construction unit,
- the textual user interface,
- the automatic image query construction unit,
- the similarity determination and retrieval unit, and
- the learning unit for index structure updating.

Two query types are possible: textual keywords and image content based queries. For inputting the textual queries the user is provided by the system with a vocabulary for image description, see Fig. 6. The same terms are used by the signal-to-symbol transformation unit which automatically transforms the numeric features extracted from the image into a symbol.

The content based query is automatically processed from the image by the image processing algorithm and feature extraction unit. This results in another problem with image databases caused by pattern recognition. An accurate automatic detection and recognition algorithm never works for all kind of images. Such an algorithm is rather domain dependent than generic. Therefore, Lee et al [15] suggest providing an image database with standard procedures for image enhancement, image analysis and feature extraction. The application of these procedure to the image is left to the user. He can process the query in an interactive fashion with the help of the procedures provided by the system. However, this requires knowledge of the user about image processing technology. He needs to be trained on image processing and pattern recognition technology in order to understand what effect an image processing and pattern recognition procedures produces on the image.

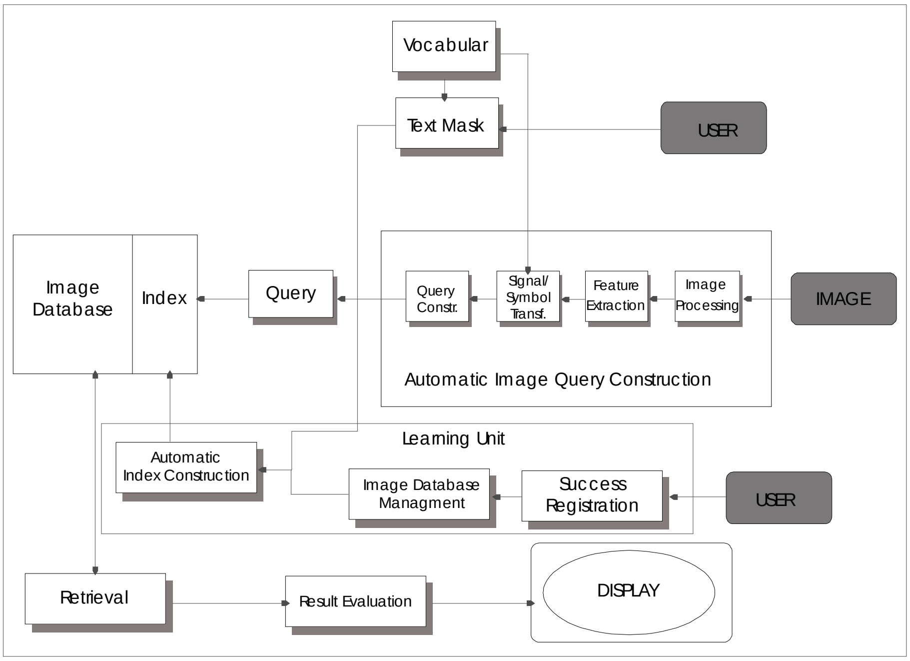

Fig. 1. Architecture of image database

Such an approach is not appropriate in most technical and medical domains where the image databases are used in day-to-day practice. It is more sufficient to provide the system with domain dependent image processing and pattern recognition algorithm for time efficiency purposes.

Therefore, our system is equipped with domain dependent image processing facilities, described in Section 9 on the special domain. That requires that the user specify the domain he is working on to the system. However, this is only if images from more than one application are contained in the database. Based on that information, the system selects the proper image processing algorithm for automatic image query processing which had been developed and installed beforehand. Such an approach ensures no interaction with the user which makes it easier for him to employ the database.

After the numeric features are transformed to symbolic features, the high-level representation of the image is formed and used for query. The high-level representation is an attributed graph. The similarity measure should work for that kind of representation and should allow exact and partial match retrieval.

The index structure is automatic constructed from the high-level representation of the image. It should allow to index the whole image as well as part images and single objects.

New images not contained in database should easily be incorporated into the image database as well as into the index structure. Therefore, a learning unit observes the success or failure of the database and activates the automatic index construction unit for incremental learning of index structure.

The result of the retrieval process is shown on display to the user.

An entry in the image data base consists of: non-image information like date of image acquisition, sensor parameter and so on, the high-level representation of the image and the image itself, see Fig. 2.

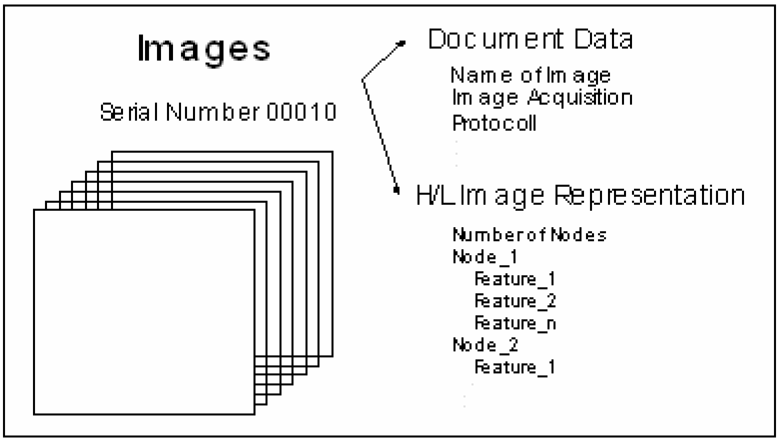

Fig. 2. Data structure

3 Content Based Query Features

The system considers two main data types: objects and scenes. Objects have features such as gray level, location, size and shape. The scenes comprises of objects and their spatial relation to each other and make up the high-level description of the image.

Generally, an automatic image processing algorithm will consist of the following steps: image preprocessing, image segmentation (meaning labeling of object pixel and background pixel), morphological operations for noise reduction, labeling of objects, contour following method, numeric feature extraction and signal-symbol conversion.

Once the object has been labeled (see Fig. 3) all the next operations are generic enough to get used for processing of other kind of images.

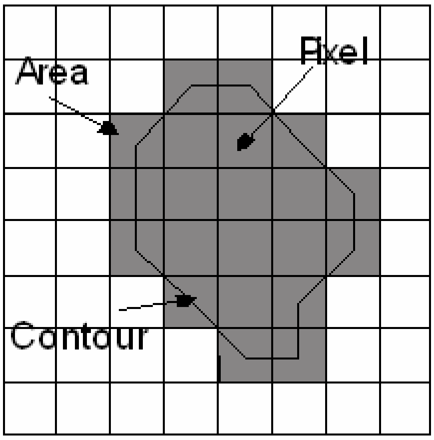

Fig. 3. Labeled object and object contour

3.1 Features

The computed features are the following:

Gray level:
From the area inside the extracted object boundary the mean gray level is computed and taken as the gray level feature for the object. We quantize the gray level space into k levels and associate a symbolic term with each level like white, very light gray, light gray, gray, dark gray, very dark gray, black.

Size:
The size of an object is computed from the area ***A*** inside the extracted object boundary.

Shape:
The shape feature is computed based on the following formulae:

$$F = 10 \bullet \frac{A}{u^2}$$

with *u* for contour length.

The following symbols are associated to the values of *F:* round, longelongated, nonround.

This is a simple shape measure which cannot describe complex objects but is accurate enough for most technical applications. If there is a need for more complex shape measure we can change our measure to other measures in our system, e.g. moment based shape measure [16] or fractal dimension[17] .

Location:
The centroid for an object is calculated and the coordinates s_x, s_y associated to that pixel are taken for the location.

$$s_y = \frac{1}{A}\sum_{i=y\min}^{y\max} w_i \cdot i \qquad s_x = \frac{1}{A}\sum_{y\min}^{y\max} w_i \cdot s_{xi} \;,$$

with w_i the number of object points in line i and s_{xi} the x-coordinate of the centroid of the *ith* line see Fig. 4.

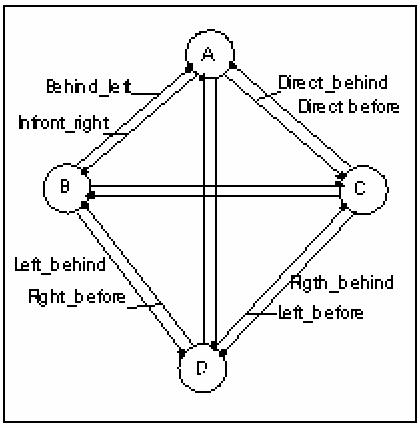

Fig. 4. Model for Relations

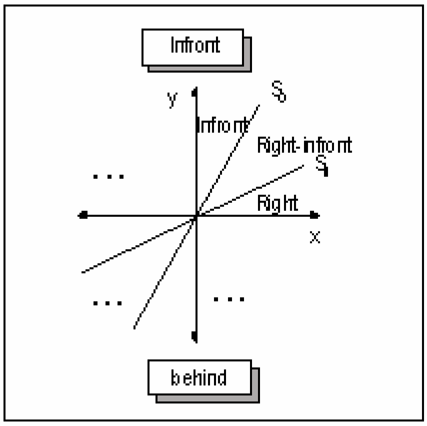

Fig. 5. Model for Spatial Relations

Spatial Location:
For expressing the spatial relation in a qualitative manner like "above" or "above left" we need a functional model for space, see Fig. 5.

In the above described example, we can think of a coordinate system that is zero in the center of mass of object A and aligned to the beam angle. Then we can describe "behind" and "above". The 4 square of the coordinate system give the specialization "left_behind", "right_behind" and "left_infront", "right_infront". We can shift the coordinate system from one object to another object. Then, we look from that focal point to the spatial relations to all other objects in the image. There are various levels of granularity[18] :

Projection

disjointness	no_contact	
tangency		
overlap	contact	no_projection_info
inclusion		

Orientation

0. no orientation
1. a. left right
   b. back in_front
2. back, left, right, in_front
3. back, left, right, in_front, ..., right_back, left_back , right_in_front,..

Note that the meaning of back, e.g. varies depending on the level of granularity. The line, for example S_0 and S_U , gives the interval for the spatial attribute right_in_front that will play a role in the later described similarity. An abstraction for two spatial attributes, one is "right" and the other is "right_in_front", would be "right , in_front".

For the representation of "more_left_behind", we need to quantize our model or a representation of a fuzzy area [19].

3.2 High-level Representation of the Image

The high-level description of an image comprises of objects, their object features, and the spatial relation between the objects. The intern representation is a attributed graph like shown in Figure 4. The graph is defined as:

Definition 1:

W ... set of attribute values
  e.g.: W = {"dark_grey", "left_behind", "directly_behind", ...}
A ... set of all attributes
  e.g.: A = {shape, object area, spatial_relationship, ...}
b: $A \rightarrow W$ partial mapping, called attribute assignments
B ... set of all attribute assignments over A and W.

A graph G = (N, p, q) consists of
  N ... finite set of nodes
  $p : N \rightarrow B$ mapping of attributes to nodes
  $q : E \rightarrow B$ mapping of attributes to edges, where $E = (NxN)\backslash I_N$ and I_N is the Identity relation in N.

The nodes are the objects and the edges are the spatial relation between the objects. Each object has attributes which are associated to the corresponding node within the graph.

The explicit specification of the set of edges can be abandoned. The spatial relation between two objects is determinable between two objects any time. That means, that there are edges between the nodes of a graph. The assumed symmetry of the set of edges is redundant with regard to spatial relations, but is advantageous for the part isomorphism algorithm. The attribute assignment of the opposite direction can be done without any problem by negation of edge labels, e.g. "/behind = in_front". Note, the set of edges is N*(N -1).

This representation allows us to consider only objects, the spatial relation between objects, or the whole image (see Section 4).

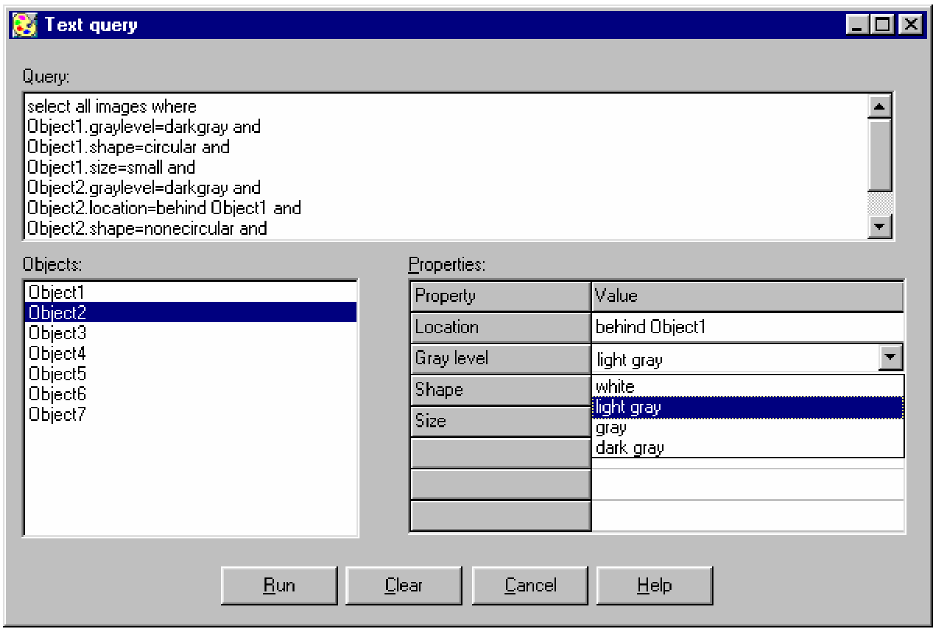

Fig. 6. Text query

4 Similarity Measure for Content-Based Retrieval

It should be possible to determine similarity in three different way: 1. Similarity among the spatial relation of objects, 2. Similarity of object features, and 3. Similarity of spatial relation and object features. All this can be done using the following similarity measure.

4.1 Comparison of Spatial Location between Objects

We may define our problem of similarity as a problem of finding structural identity or similarity between two structures. If we are looking for structural identity, we need to determine isomorphism. That is a very strong requirement. We may relax this requirement by demand partial isomorphism.

Based on partial isomorphism, we can introduce a partial order over the set of graphs:

Definition 2:

Two graphs $G_1 = (N_1,p_1,q_1)$ and $G_2 = (N_2,p_2,q_2)$ are in the relation $G_1 \leq G_2$ iff there exists a one-to-one mapping $f: N_1 \rightarrow N_2$ with

(1) $p_1(x) = p_2(f(x))$ for all $x \in N_1$
(2) $q_1(x) = q_2(f(x), f(y))$ for all $x,y \in N_1$, $x \neq y$.

Is a graph G_1 included in another graph G_2 then the number of nodes of graph G_1 is not higher than the number of nodes of G_2.

4.2 Similarity Measure for Spatial Location and Object Features

Similarity between attributed graphs can be handled in many ways. We propose the following way for the measure of closeness.

In the definition of part isomorphism we may relax the required correspondence of attribute assignment of nodes and edges in that way that we introduce ranges of tolerance:

If $a \in A$ is a attribute and $W_a \subseteq W$ is the set of all attribute values, which can be assigned to a, then we can determine for each attribute a mapping:

$$\text{distance}_a : W_a \rightarrow [0,1].$$

The normalization to a real interval is not absolute necessary but advantageous for the comparison of attribute assignments.

For example, let a be an attribute a = spatial_relationship and

W_a = {behind_right, behind_left, infront_right, ...}.

Then we could define:

distance$_a$ (behind_right, behind_right) = 0
distance$_a$ (behind_right, infront_right) = 0.25
distance$_a$ (behind_right, behind_left) = 0.75 .

Based on such distance measure for attributes, we can define different variants of distance measure as mapping:

distance : $B^2 \rightarrow R^+$

(R^* ... set of positive real numbers) in the following way:

$$\text{distance}(x,y) = 1/D \sum_{a \in D} \text{distance}_a (x(a), y(a))$$

with D = domain (x) $\cap$ domain(y).

Usually, by the comparison of graphs not all attributes have the same priority. Thus, it is good to determine a weight factor v_a and then, define the distance as following:

$$\text{distance}(x,y) = \sum_{a \in D} v_a * \text{distance}_a (x(a), y(a))$$

For definition of part isomorphism, we get the following variant:

Definition 3:

Two graphs $G_1 = (N_1, p_1, q_1)$ and $G_2 = (N_2, p_2, q_2)$ are in the relation $G_1 \leq G_2$ iff there exists a one-to-one mapping f: $N_1 \rightarrow N_2$ and threshold's C_1, C_2 with

(1) distance($p_1(x)$, $p_2(f(x))$) $\leq C_1$ for all $x \in N_1$

(2) distance($q_1(x,y)$, $q_2(f(x), f(y))$) $\leq C_2$ for all $x,y \in N_1$, $x \neq y$.

There is another way to handle similarity is the way the L-sets are defined and particularly the inclusion of K-lists:

Given C a real constant, $n \in N_1$ and $m \in N_2$. $K(n) \subseteq_C K(m)$ is true iff for each attribute assignment b_1 of the list K(n) attribute assignment b_2 of K(m) exists, such that distance(b_1,b_2) $\leq$ C.
Each element of K(m) is to assign to different element in list K(n).

Obviously, it is possible to introduce a separate constant for each attribute. Depending on the application, the inclusion of the K-lists may be sharpened by a global threshold:

If it is possible to establish a correspondence g according to the requirements mentioned above, then an additional condition should be fulfilled:

$$\sum_{(x,y) \in g} \text{distance}(x,y) \leq C_3 \quad (C_3 \text{ - threshold constant}).$$

Then, for the L-set we get the following definition (see also Sect. 5.1) :

Definition 4:

$L(n) = \{m \_ m \in N_2, \text{distance } (p_1(n), p_2(m)) \leq C_1, K(n) \subseteq_C K(m) \}$.

In step 3 of the algorithm for the determination of one-to-one mapping, we should also consider the defined distance function for the comparison of the attribute assignments of the edges. This new calculation increases the total amount of effort, but the complexity of the algorithm is not changed.

For basic introduction to graph theory and graph grammars, see [20] and [21]. For other similarity concepts see [13] and [14].

5 Algorithm for Determining Similarity between Structural Representations

Now, consider an algorithm for determining the part isomorphism of two graphs. This task can be solved with an algorithm based on [20]. The main approach is to find an overset of all possible correspondences f and then exclude non promising cases. In the following, we assume that the number of nodes of G_1 is not greater than the number of nodes of G_2.

A technical aid is to assign to each node n a temporary attribute list K(n) of all attribute assignments of all the connected edges:

$$K(n) = (a \mid q(n,m) = a, m\ N \setminus \{n\}) \quad (n \in N).$$

The order of list elements has no meaning. Because all edges exist in a graph the length of K(n) is equal to 2*(|N|-1).

For demonstration purposes, consider the example in Fig. 8. The result would be:

$$K(X) = (bl, bl, br)$$
$$K(Y) = (br, br, \underline{bl}).$$

In the worst case, the complexity of the algorithm is $O(\mid N \mid^3)$.

In the next step, we assign to each node of G_1 all nodes of G_2 that could be assigned by a mapping f. That means we calculate the following sets:

$$L(n) = \{ m \mid m \in N_2, p_1(n) = p_2(m), K(n) \subseteq K(m)\} .$$

The inclusion $K(n) \subseteq K(m)$ shows that in the list K(m) the list K(n) is included without considering the order of the elements. Does the list K(n) multiple contains an attribute assignment then the list K(m) also has to multiple contain this attribute assignment.

For the example in Fig. 7 and Fig. 8 we get the following L-sets (see table 1):

$L(X) = \{A\}$
$L(Y) = \{B_1\}$
$L(Z) = \{C\}$
$L(U) = \{ D, B_2\}$.

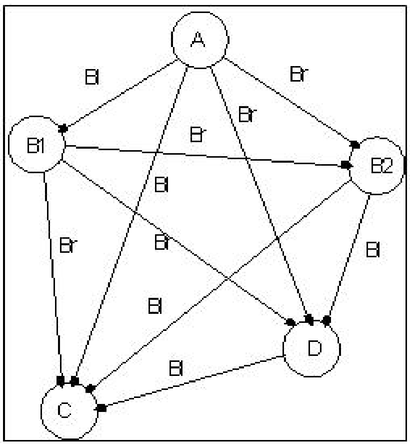

Fig. 7. Graph of Image_1

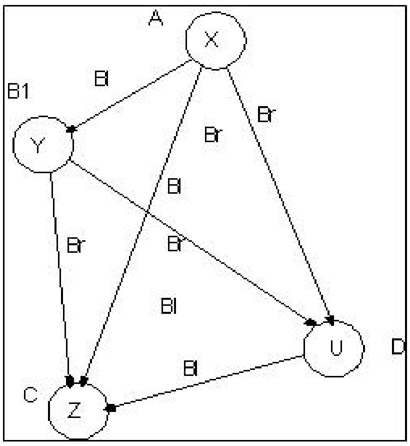

Fig. 8. Graph of Image_2 and results of subgraph Isomorphism to Image_1

N_1	f_1	f_2
X	A	A
Y	B_1	B_1
Z	C	C
U	D	B_2

Table 1. Demonstration of L-Sets

We did not consider in this example the attribute assignments of the nodes.

Now, the construction of the mapping f is prepared and if there exists any mapping then must hold the following condition:

$$f(n) \in L(n) \qquad (n \in N_1) .$$

The first condition for the mapping f regarding the attribute assignments of nodes holds because of the construction procedure of the L-sets. In case that one set L(n) is empty, there is no partial isomorphism.

Also, if there are nonempty sets, in a third step is to check if the attribute assignments of the edges match.

If there is no match, then the corresponding L-set should be reduced to:

for all nodes n_1 of G_1
    for all nodes n_2 of $L(n_1)$
        for all edges (n_1, m_1) of G_1
        if for all nodes m_2 of $L(m_1)$
            $p_1(n_1,m_1) \neq p_2(n_2,m_2)$
        then $L(n_1) := L(n_1) \setminus \{n_2\}$

If the L-set of node has been changed during this procedure, then the examinations already carried out should be repeated. That means that this procedure should be repeated until none of the L-sets has been changed.

If the result of this step 3 is an empty L-set, then there is also no partial isomorphism. If all L-sets are nonempty, then some mappings f from N1 to N2 have been determined. If each L-set contains exactly only one element, then there is

only one mapping. In a final step, all mappings should be excluded, which are not of the one-to-one type.

For example, let us compare the representation of pore_1 and pore_2 in Fig. 7 and Fig. 8. In step 3, the L-set of pore_1 will not be reduced and we get two solutions, shown in Fig. 9 and Fig. 10.

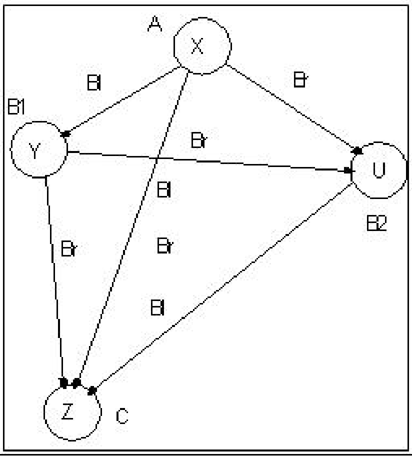

Fig. 9. Second result of subgraph Isomorphism of Image_1 and Image_2

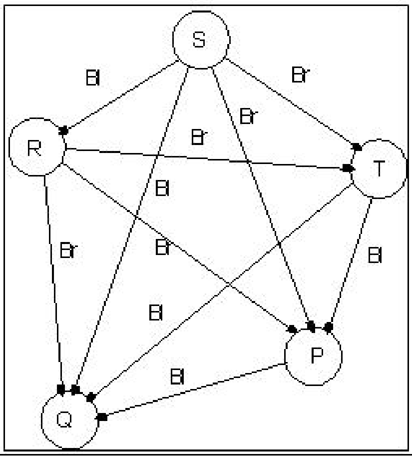

Fig. 10. Graph of image_3 and results for subgraph Isomorphism to Image_2

If we compare the representation of pore_1 and pore_3, a L-set of pore_1 also contains two elements: L(U) = {T, P}

However in step 3, the element T will be excluded if the attribute assignments of the edges (U,Y) and (T,R) do not match when node U is examined.

If the L-set of a node has been changed during step 3 then the examinations already carried out should be repeated. That means that step 3 is to repeat until there is no change in any L-set.

This algorithm has a total complexity of the order $O (|N_2|^3, |N_1|^3 * |M|^3)$. $|M|$ represents the maximal number of elements in any L-set ($|M| \leq |N|$).

6 Index Structure

The initial image database may be built up by existing data entries. Therefore; a nonincremental learning procedure is required in order to built the index structure. When using the system, new cases may be stored in the data base. They should be integrated into the already existing image database. Therefore, we need an incremental learning procedure [22].

Elements in the index structure are representations between graphs. We have considered similarity based on partial isomorphism as an important relation between these graph. It is possible to organize the index structure as directed graph because of this characteristic.

In the following, we will define the index structure of the image database as a graph that contains the image graphs described above in the nodes:

Definition 5

H is given, the set of all image graphs.
A index graph is a Tupel IB = (N,E,p), with

(1) $N \subseteq H$ set of nodes and

(2) $E \subseteq N^2$ set of edges.
This set should show the partial isomorphism in the set of nodes, meaning it should be valid

$x \leq y \Rightarrow (x,y) \in E$ for all $x,y \in N$.

(3) $p: N \rightarrow B$ mapping of image names to the index graph.

Because of the transitivity of part isomorphism, certain edges can be directly derived from other edges and do not need to be separately stored. A relaxation of top (2) in definition 5 can be reduced storage capacity.

7 Learning of Index Structure

Now, the task is to build up the graphs of IB in a supergraph by a learning environment.

Input is:

Supergraph IB = (N, E, p) and
image graph $x \in H$.

Output is:

modified Supergraph IB' = (N', E', p')
with $N' \subseteq N \cup \{x\}$, $E \subseteq E'$, $p \subseteq p'$

At the beginning of the learning process or the process of construction of index graph N can be an empty set.

The attribute assignment function p' gives the values (p'(x), (dd)) as an output. This is an answer to the question: What is the name of the image name that is mirrored in the image graph x?

The inclusion

$$N' \subseteq N \cup \{x\}$$

says that the image graph x may be isomorphic to one image graph y contained in the image database contained image graph y, so $x \leq y$ and also $y \leq x$ hold. Then, no new node is created, which means the image database is not increased.

The algorithm for the construction of the modified index structure IB' can also use the circumstance that no image graph is part isomorph to another image graph if it has more nodes than the second one.

As a technical aid for the algorithm there are introduced a set N_i. N_i contains all image graphs of the image database IB with exactly i nodes. The maximal number of nodes of the image graph contained in the image database is k, then it is valid:

$$N = \bigcup_{i=1}^{k} N_i$$

The image graph which has to be included in the image database has l nodes ($l \rangle 0$). By the comparison of the current image graph with all in the image database contained graphs, we can make use of transitivity of part isomorphism for the reduction of the nodes that has to be compared.

Algorithm

E' := E;
Z := N;
for all y∈ N_l
if x ≤ y **then** [IB' := IB; return];
N' := N ∪ {x};
for all i with 0 < i < l;
    for all y ∈ N_i \ Z;
    for all y ≤ x **then** [Z := Z \ {u _ u ≤ y, u ∈ Z};
        E' := E' ∪ { (y,x)}];
for all i with l < i ≤ k
    for all y ∈ N_i \ Z
    if x ≤ y **then** [Z := Z \ {u _ y ≤ u, u ∈ Z };
        E' := E' ∪ { (x,y)}];
p' := p ∪ { (x, (dd : unknown))};

If we use the concept of Sect. 4 for the determination of similarity, then we can use the algorithm of Sect. 4 without any changes. But we should notice that for each group of image graphs that is approximately isomorphic, the first occurred image graph is stored in the case base. Therefore, it is better to calculate of every instance and each new instance of a group a prototype and store this one in the index structure of the image database.

8 Retrieval

When an image is given as query to the system, first of all the image graph is constructed by the feature extraction unit. The query can be the high-level representation from the whole image or only one node representing one object. This representation is given to the image database as a query. The question is: Is there any similar case in the image database?

This question is answered by matching the current case through the index hierarchy. First the image representation with the same number of nodes like the query image representations is determined (see algorithm in Sect. 6), then between the remaining set of images having similar representation and the query the part isomorphism relation is determined. The output of the system are all images y in the image database which are in relation to the query x as follows:

$$(x \le y \vee y \le x) \wedge \left\| N_x \right| - \left| N_y \right\| \le d$$

where d is a constant which can be chosen by the user of the system and N_x and N_y are the sets of nodes of the graph x resp. Y.

Figure 11 shows an example of a index structure and the relation of current a query to images in image database.

The user will see these images on display.

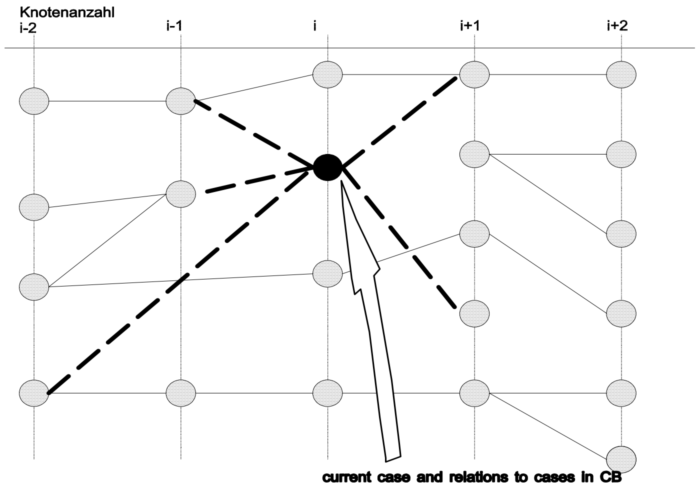

Fig. 11. Schematic Description of the Retrieval Process

9 Results

We used our system for the storage of ultra sonic images from non-destructive testing. The images represent a industrial metallic component having a defect (crack or hole) inside of the component. The images were taken by a SAFT ultra sonic imaging system.

Each entry represents an inspection of an industrial part, consisting of:

- an image acquisition protocol: sensor parameters, the parameters of the amplifier,
- a protocol about the type or the characteristic of the component,
- the information about the defect type,
- and the image.

It is necessary to keep these images for recourse purposes as certificate and also, for hard inspection cases. Than an image from the difficult inspection problem is compared to the other inspection problems contained in the image database to find the right interpretation of the defect type.

The certification of the condition of technical components, buildings and other industrial parts is one of the main purposes why images need to be stored in image databases.

Figure 12 shows a ultra sonic image of crack inside a flat metallic component as gray level image. The image gets posterized for viewing purposes and is then displayed as color image on the display of the database, see Fig. 15. From the original image a binary image is obtained by thresholding technique. Preprocessing is done by morphological operators like dilation and erosion and afterwards the objects are labeled by the contour following procedure [23]. The results after the image processing steps are shown for the segmentation in Figure13 and for the object labeling in Figure 14. The resulting high-level description get displayed graphically in the image query, see Figure 15. Queries can be: the high-level representation shown in Figure 15, only the spatial relation between the objects without considering the attributes, and the attributes for one object.

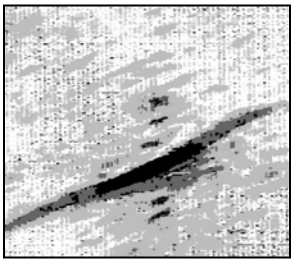

Fig. 12. Original Ultra Sonic image

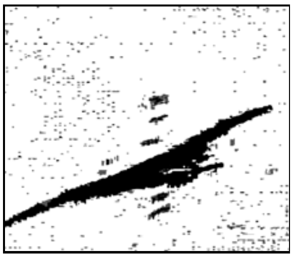

Fig. 13. Segmented image

Fig. 14. Processed Image Query

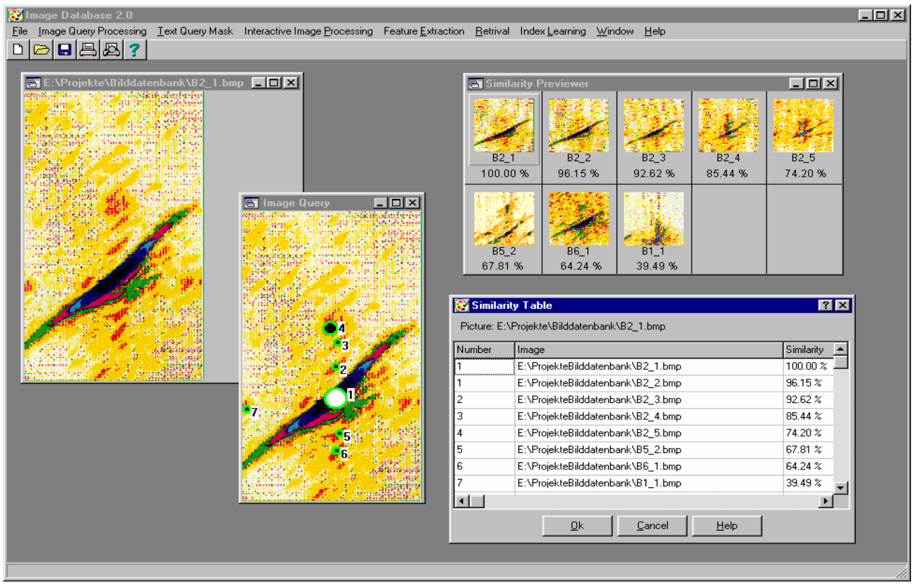

Fig. 15. Printscreen of the Image Retrieval System

By pressing the button “query processing” from the current image, the query is automatically processed and presented to the user on the display. The similar images are shown in a preview on display. The values for similarity are shown in another frame to the user ranked accordingly. If the user selects one of the objects by mouse click in the query image, only images having similar objects are retrieved. If he wants to consider only the spatial relation he needs to specify this in the menu “interactive query”.

The system was implemented on PC with C++.

10 Conclusion

In our paper, we proposed an image database architecture which can be used for most industrial problems. The image database is capable of handling structural representation of images. Indexing is possible object based, spatial relation based, and with a combination of both. The query can be a textual query or a image content based query. For the latter, the system is provided with image processing and pattern recognition facilities. By signal-to-symbol transformation the processed image content is automatically transformed into symbols which are the same the user can use for textual query. This enables the user to understand the image content and ensures that the same algorithm can be user for indexing and similarity retrieval.

The proposed a similarity measure for structural representations is fundamental and flexible enough to get used for a class of different problems. We described the similarity measure in detail and showed the different degree of freedom the similarity measure allows. The algorithm for similarity determination is of polynomial order and fast enough for the considered class of applications.

The index structure is organized in a hierarchical fashion based on the relation of sub graph isomorphism. The index structure can be automatically incrementally learned which allows to incorporate new images in a easy fashion.

Finally, the performance of the system is presented based on ultra sonic images from non-destructive testing.

References

[1] Chang, S.F.: Content-Based Indexing and Retrieval of Visual Informations, IEEE Signal Processing Magazine 14 (4), 45-48 (1997)

[2] Meghini, C.: An image retrieval model based on classical logic. In: Proceedings of the 18th annual international ACM SIGIR conference on Research and development in information retrieval. pp. 300-308, ACM (1995)

[3] Whalen, Th., Lee, E.S., Safayeni, F.: The retrieval of images for image databases: trademarks. Behaviour and information technology, vol. 14 (1), 3-13 (1995)

[4] Rao, A.R.: A Taxonomy for Texture Description and Identification, Springer (1990)

[5] Mehrotra, R.: Integrated Image Information Managment: Research Issues. In: Proc. of the SPIE, vol. 2488, pp. 168-177 (1995)

[6] Hildebrandt, J., Irving, C., Tang K.: An Implementation of Image Database Systems using differing indexing method. In: Proc. of the SPIE, vol. 2606, pp. 246-257 (1995)

[7] Samadani, R., Han, C., Katragadda, L.K: Content based event selection from satellite images of the aurora. In: Proc. of SPIE, vol. 1908, pp. 50-59 (1993)
[8] Zhang, Q.L., Chang, S.K., Yau, S.S.T.: A unified approach to iconicindexing, retrieval, and maintenance of spatial relationships in image databases. Journal of Visual Communiction and Image Representation 7 (4), 307-24 (1996)
[9] Lee, D., Barber, R., Niblack, W.: Indexing for Complex Queries on a Query-By-Content Image Database. In: Proc. of the ICPR`94, Jerusalem Oct.9-13 1994, Israel, vol. 1, p. 142-146 (1994)
[10] Papadias, D., Sellis, T.: A Pictorial Query-By-Example Language. Journal of Visual Languages and Computing 6, 53-72 (1995)
[11] Lew, M.S., Huijsmans, D.P, Denteneer, D.: Optimal keys for image database. In: Del Bimbo, A. (Ed.) Proc. of the 9th ICIAP`97, Vol. II. LNCS, Vol. 1311 pp. 148-155, Springer Verlag (1997),
[12] Johannesen, F.R., Raaschou, S., Larsen, O.V., Jürgensen, P.: Using Weighted Minutiae for Fingerprint Identification. In: Perner, P., Wang, P., Rosenfeld, A. (Eds.) Proceedings of SSPR'96: Advances in Structural and Syntactical Pattern Recognition. LNCS, vol. 1121 pp. 289-299, Springer Verlag (1996)
[13] Bunke, H., Wakimoto. K., Tanaka, S., Maeda, A.: A similarity retrieval method of drawings based on graph representation. Systems and Computer in Japan 26 (11), 100-109 (1995)
[14] Vosselman, G.: Relational Matching. LNCS, vol. 628, Springer, Heidelberg (1992)
[15] Lee, D., Barber, R., Niblack, W., Flickner, M., Hafner, J., and Petkovic, D.: Queryby Image Content using multiple objects and multiple features: User Interface Issues. In ICIP, Austin (1994)
[16] Zamperoni, P.: Feature Extraction. In: Maitre, H., Zinn-Justin, J. (Eds.) Progress in Picture Processing. pp.123-184, Elsevier Science (1996)
[17] Mandelbrot, B.B.: The Fractal Geometry of Nature. W.H. Freeman and Company (1977)
[18] Hernandez, D.: Relative Representation of Spatial Knowledge: The 2-D Case. Report FKI-135-90, TU Muenchen (Aug. 1990)
[19] Schirra, J.R.: A Contribution to Reference Semantics of Spatial Prepositions: theVisualization and its Solution in VITRA. SFB 314, Ki-Wissensbasierte Systeme, Dec. (1990)
[20] Schlesinger, M.I.: Mathematical Tools of Picture Processing (in Russian). Naukowa Dumka, Kiew (1989)
[21] Harary, F.: Graph Theory. Addison-Wesley (1972)
[22] Perner, P., Pätzold, W.: An Incremental Learning System for Interpretation of Images. In: Dori, D., Bruckstein, A. (Eds.) Proc. of SSPR: Shape, Structure and Pattern Recognition, pp. 311-323, World Scientific Publishing Co (1995)
[23] Niemann, H.: Pattern Analysis and Understanding. Springer Verlag (1990)