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

Petra Perner

Institute of Computer Vision and Applied Computer Sciences
Arno-Nitzsche-Str. 45, 04277 Leipzig, Germany
Email: ibaiperner@aol.com

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 the paper, we proposed an image database architecture, which can be used for most industrial problems. The image database is able to handle structural representations of images. Indexing is possible object based, spatial relation based, and by a combination of both. The query can be a textual query or an image content-based query. We describe 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

In the 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 a 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 input 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 are generic. The image-processing unit in the system uses domain dependent algorithm. This requires from the user to 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 determination is presented in Section 4 and Section 5. Section 6 and Section 7 deal with the index structure. Retrieval is presented in Section 8. Finally, we show our results in Section 9 and give conclusions in Section 10.

2 Overview of Image Database

Our image database mainly consists of five 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. The same terms are used by the signal-to-symbol transformation unit, which automatically transforms the numeric features extracted from the image to 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 to provide an image database with standard procedures for image enhancement, image analysis and feature extraction. The application of these procedures 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, it requires by the user knowledge about the image processing technology. He needs to be trained on image processing and pattern recognition technology in order to understand what effect image processing and pattern recognition procedures produces on the image.

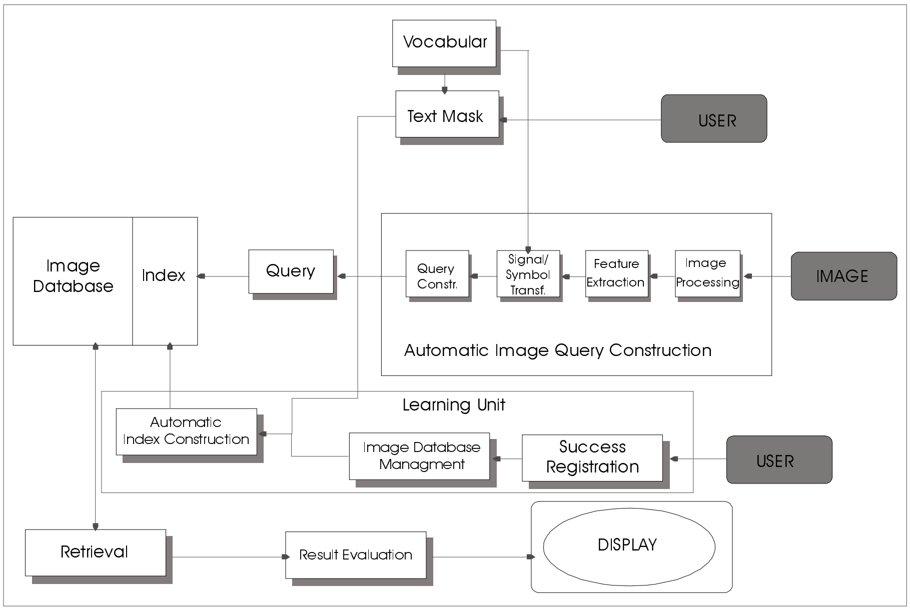

Fig. 1 Architecture of the Image Database

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 sufficient to provide the system with domain dependent image processing and pattern recognition algorithm for time efficiency purposes.

Therefore is our system equipped with domain dependent image processing facilities, described in Section 9 on the special domain. That requires from the user to specify to the system the domain he is working on but only in case that images from more than one application is contained in 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 indexing the whole image as well as part images and single objects.

New images not contained in database should easily be incorporated to the image database as well as to 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 of the image database consists of: non-image information like date of image acquisition, sensor parameter and so one, the high-level representation of the image and the image itself, see Fig. 2.

3 Content Based Query Features

The system considers two main data types: objects and scenes. Objects have features like 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.

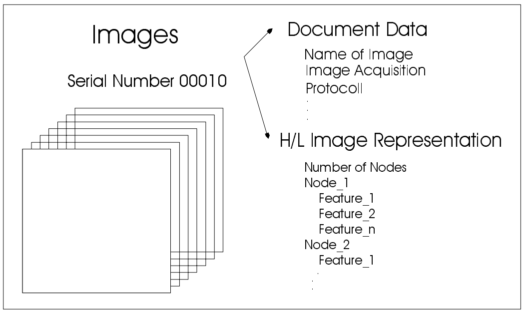

Fig. 2 Data Structure

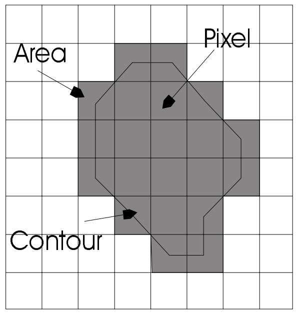

Fig. 3 Labeled Object and Object Contour

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 associates to each level a symbolic term 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 $\boldsymbol{A}$ 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, non-round.

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.

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

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

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 four 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].

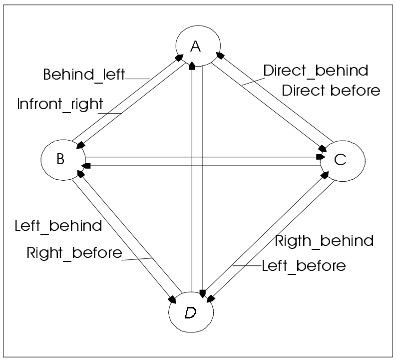

Fig. 4 Image Graph

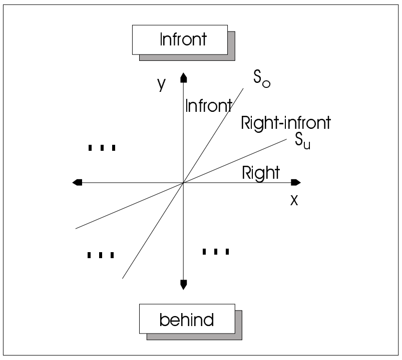

Fig. 5 Model for Spatial Relations

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 an 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 $E = (N\mathrm{x}N)\backslash I_N$
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 gets 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, the edges between the nodes of a graph do exist. The assumed symmetry of the set of edges is redundant according 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).

4 Similarity Measure for Content-Based Retrieval

Determination of similarity should be possible in three different ways: 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 by the following similarity measure.

4.1 Comparison of Spatial Location between Objects

We may define our problem of similarity as to find structural identity or similarity between two structures. If we ask for structural identity, we need to determine isomorphism. That is a very strong requirement. A relaxation of the requirements is to ask for part isomorphism.
Based on part 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*** a mapping:

$distance_a : W_a \rightarrow [0,1]$.

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:

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

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:

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

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 $x,y \in N_1$, $x \neq y$.

Another way to handle similarity is the way how the L-set's 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).

Obvious, it is possible to introduce a separate constant for threshold for every attribute. Depending on the application, the inclusion of the K-lists may be sharpen up by a global threshold:

> If it is possible to establish a correspondence g according to the requirements mentioned above, then an additional condition is to fulfill:
>
> $$\sum_{(x,y) \in g.} \text{distance}(x,y) \leq C_3 \quad (C_3 \text{ - threshold constant}) \ .$$

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 is also to consider the defined distance function for the comparison of the attribute assignments of the edges. The total effort is increasing by this new calculation 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 higher than the number of nodes of $G_{2.}$
A technical help 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 do exist in a graph the length of K(n) is equal to $2*(|N|-1)$.
For demonstration, consider the example in Fig. 7. The result would be:

K(X) = (bl, bl, br)
K(Y) = (br, br, <u>bl</u>).

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

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. 6 and Fig. 7, we get the following L-sets:

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

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:

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

The first condition for a mapping f regarding the attribute assignments of nodes holds because of the construction procedure of the L-sets. In case of one, set L(n) is empty then there is no part isomorphism.
In addition, in the case of nonempty sets in a third step it is to check if the attribute assignments of the edges do match.
If there is no match then the corresponding L-set is to reduce:

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 $L(n_1) := L(n_1) \setminus \{n_2\}$

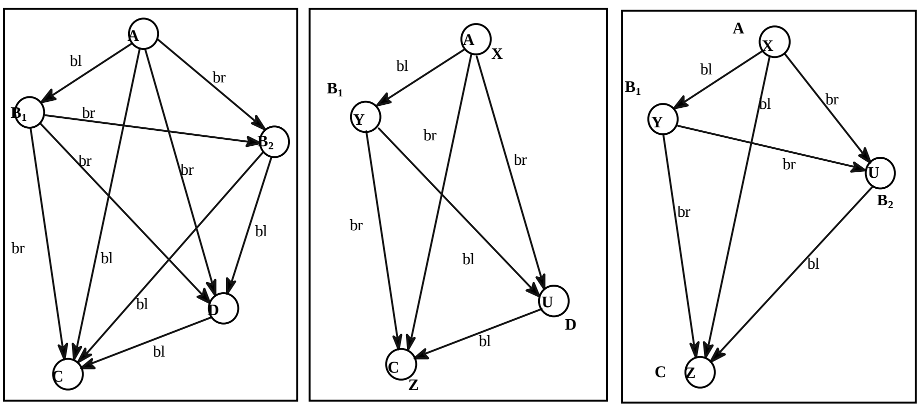

Fig. 6 Graph_1 Fig. 7 Graph_2 Fig. 8 Result

If during this procedure, the L-set of node has been changed then the examinations already carried out are to repeat. That means, this procedure is to repeat until none of the L-sets was changed.
Is the result of this step 3 an empty L-set then there is also no part isomorphism. If all L-sets are nonempty then some mapping's f form N_1 to N_2 are determined. If each L-set contains exactly only one element then there is only one mapping. In a final step all mappings are to exclude, which are not of the type one-to-one.

For example, let's compare the representation of pore_1 and pore_2 in Fig. 6 and Fig. 7. In step 3, the L-set of pore_1 will not be reduced and we get two solutions, shown in Fig. 7 and Fig. 8:

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

If we compare the representation of pore_1 and pore_3 a L-set of pore_1 contains also two elements:

L(U) = {T, P}

However, in step 3, the element T will be excluded because during examination of the node U the attribute assignments of the edges (U,Y) and (T,R) do not match.
If during step 3, the L-set of a node has been changed then the examinations already carried out are to repeat. That means, step 3 is to repeat until there is no change of 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 ($|M| \leq |N_2|$).

6 Index Structure

By the development and usage of the database, it is to consider that the database may be permanently changing during learning process. The initial image database may be built up by existing data entries therefore; a nonincremental learning procedure is required. During the use of the system, new cases may be stored in the database. They

should be integrated in the already existing image database. Therefore, we need an incremental learning procedure [22].
Elements in the index structure are representations between graphs. As an important relation between these graphs, we have considered similarity based on part isomorphism. Because of this characteristic, it is possible to organize the index structure as directed graph.
In the following, we define the index structure of the image database as graph that contains the before described image graphs in the nodes:

Definition 5

Given is H, 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 part 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. Formally, this task is to solve permanently:

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 as output the value (p'(x), (dd)). This is an answer to the question: What is the image name that is mirrored in the image graph x?
The inclusion $N' \subseteq N \cup \{x\}$ says that the image graph x can be isomorphic to one in the image database contained image graph y, so $x \leq y$ and also $y \leq x$ hold. Then, no new node is created that 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 isomorphic to another image graph if it has more nodes then the second one.

For technical help 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 〉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 similarity handling, then we can use the algorithm of Sect. 4 without any changes. However, 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 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 as query to the image database. 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

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

set of images having similar representation and the query the part isomorphism relation is determined. The output of the system is all images y in the image database, which are in relation to the query x as follows:
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 9 shows an example of an index structure and the relation of current a query to images in image database.
The user will see these images on display.

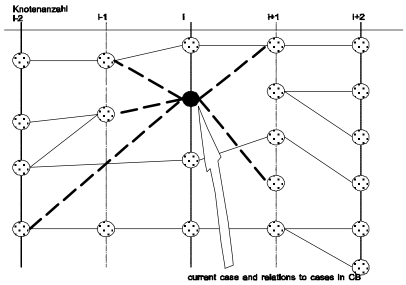

Fig. 9 Schematic Description of the Retrieval Process

9 Results

We used our system for the storage of ultra sonic images from non-destructive testing. The images represent an 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 10 shows an ultra sonic image of crack inside a flat metallic component as gray level image. The image are 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 contour following procedure [23]. The results after the image processing steps are shown for the segmentation in Figure11 and for the object labeling in Figure 12. The

resulting high-level description are displayed graphically in the image query, see Figure 16. Queries can be: the textual description of the image (see Fig. 13) or the high-level representation of the image (see Fig. 14).

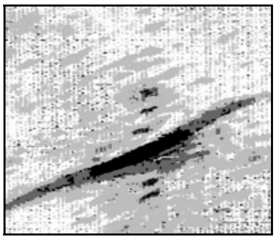
Fig. 10 Original Ultra Sonic Image

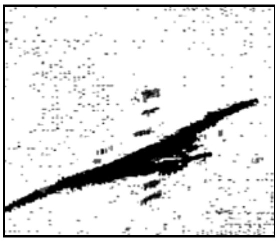
Fig. 11 Segmented Image

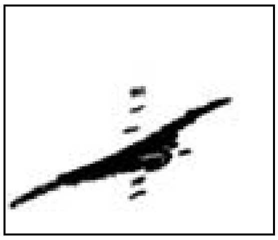
Fig. 12 Processed Image Query

By pressing the button "query processing" from the actual image the query is automatically processed and presented on the display to the user. 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++ Images.

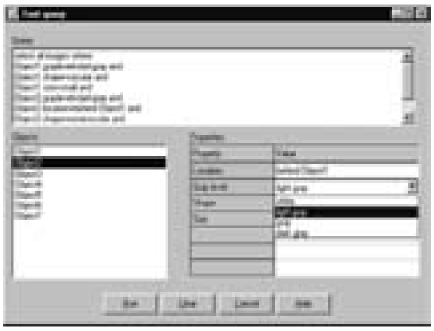
Fig. 13 Text Query

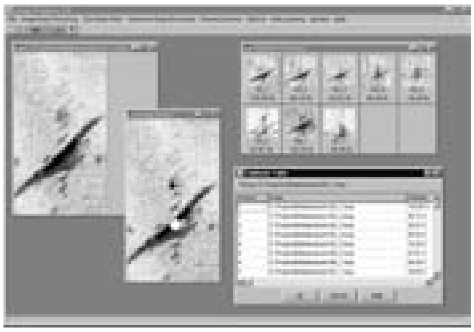
Fig. 14 Hardcopy of the User Interface

10 Conclusion

In our paper, we proposed 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 an image content based query. For the later, 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 an easy way.
Finally, the performance of the system is presented based on ultra sonic images from non-destructive testing.

References

1. S.-F. Chang,"Content-Based Indexing and Retrieval of Visual Informations," IEEE Signal Processing Magazine (July 1997), vol. 14, no.4, p. 45-48.
2. C., Meghini, "An image retrieval model based on classical logic", SIGIR Forum (1995) spec. Issue, p. 300-308.
3. Th. Whalen, E.S. Lee, and F. Safayeni, "The retrieval of images for image databases: trademarks," Behaviour and information technology, 1995, vol. 14, No. 1, 3-13.
4. A.R. Rao, A Taxonomy for Texture Description and Identification, Springer Verlag 1990.
5. R. Mehrotra, "Integrated Image Information Managment: Research Issues," Proc. of the SPIE 1995, vol. 2488, 168-177.
6. J. Hildebrandt, C. Irving, and K. Tang,"An Implementation of Image Database Systems using differing indexing methods," Proc. of the SPIE (1995), vol. 2606, p. 246-257.
7. R. Samadani, C. Han, and L.K. Katragadda, "Content based event selection from satellite images of the aurora," Proc. of SPIE (1993), vol. 1908, p. 50-59.
8. Q.-L. Zhang, S.-K. Chang, and S. S.-T. Yau," A unified approach to iconic indexing, retrieval, and maintenance of spatial relationships in image databases," Journal of Visual Communiction and Image Representation (Dec. 1996), vol. 7, no.4, p. 307-24.
9. D. Lee, R. Barber, and Wayne Niblack, "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.
10. D. Papadias and T. Sellis,"A Pictorial Query-By-Example Language," Journal of Visual Languages and Computing (1995) 6, p. 53-72.
11. M.S. Lew, D.P. Huijsmans, and D. Denteneer, "Optimal keys for image database," In Proc. of the 9th ICIAP`97, A. Del Bimbo (Eds.) , Springer Verlag 1997, p. 148-55.
12. F.R. Johannesen, S. Raaschou, O.V. Larsen, and P. Jürgensen,"Using Weighted Minutiae for Fingerprint Identification," In Proc: P. Perner, P. Wang, and A. Rosenfeld (Eds.), Advances in Structural and Syntactical Pattern Recognition, Springer Verlag 1996, pp. 289-299
13. H. Bunke, K. Wakimoto, S. Tanaka, and A. Maeda, "A similarity retrieval method of drawings based graph representation," Systems and Computer in Japan, Oct. 1995, vol. 26, no. 11, p. 100-109.
14. G. Vosselman, Relational Matching, Lecture Notes in Computer Sciences, Springer Verlag 1992.
15. D. Lee, R. Barber, W. Niblack, M. Flickner, J. Hafner, and D. Petkovic, "Query by Image Content using multiple objects and multiple features: User Interface Issues. In ICIP, Austin, TX, 1994

16. P. Zamperoni,"Feature Extraction," In: H. Maitre and J. Zinn-Justin, Progress in Picture Processing, Elsevier Science, 1996, pp.123-184.
17. B. B. Mandelbrot, The Fractal Geometry of Nature, W.H. Freeman and Company 1977
18. D. Hernandez,"Relative Representation of Spatial Knowledge: The 2-D Case," Report FKI-135-90, Aug. 1990, TU Muenchen.
19. J.R. Schirra, "A Contribution to Reference Semantics of Spatial Prepositions: the Visualization and its Solution in VITRA," SFB 314, KI-Wissensbasierte Systeme, Dec. 1990.
20. M.I. Schlesinger, "Mathematical Tools of Picture Processing", (in Russian), Naukowa Dumka, Kiew 1989.
21. F. Harary, "Graph Theory," Addison-Wesley 1972.
22. P. Perner and W. Pätzold,"A Incremental Lerning System for Interpretation of Images," In Proc. of SSPR: D. Dori and A. Bruckstein (Eds.) Shape, Structure and Pattern Recognition, World Scientific Publishing Co., pp. 311-323
23. P. Perner,"Ultra Sonic Image Interpretation for Non-Destructive Testing," In Proc. MVA`96, pp. 552-554.