Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.

# Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation

Petra Perner

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

**Abstract.** In our previous work, we introduced the basic structure of a case-based reasoning system for image interpretation, a structural similarity measure, and some fundamental learning techniques. In this paper, we describe more sophisticated learning techniques that are different in abstraction level. We evaluate our method on a set of images from the non-destructive testing domain and show the feasibility of the approach. As result, we can show that conventional image processing methods combined with machine learning techniques form a powerful tool for image interpretation.

**Keywords:** CBR Learning, Incremental Learning, Image Interpretation

## 1 Introduction

We introduced in [1] the basic structure of a case-based reasoning system for image interpretation and a structural similarity measure. The fundamental incremental learning method for such a CBR system was introduced in [2]. In this paper, we describe some improved incremental learning capabilities of a CBR system.

In Section 2, we give a brief introduction to CBR and review some fundamental topics introduced in [2][3]. The cases are represented as attributed image graphs where the nodes are the objects in the image, the node attributes are the attributes of the objects, and the arcs are the spatial relations between the object.
Such kinds of representations are widely used in image interpretation systems like e.g. scene analysis and interpretation of technical drawings. The main problem with such kind of applications is that it is hard to build a vision model because of the complexity of the domain [5]. However, often a large number of cases are available, which favors the use of a CBR image interpretation system equipped with proper learning techniques.

Section 3 describes the learning strategies in the CBR system, which were already introduced in [2][4]. Different learning strategies are necessary to get automatically a more compact and error-tolerant representation of the case base. The used learning strategies are different in abstraction ability. The simplest one is just learning of new

Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.

cases, which can be useful for the construction of the complete model. Once a new case has been observed and confirmed by more similar cases the first more generalizing learning strategy starts to learn more compact representatives for this class of cases. While learning new case classes and more compact case classes, the relation of these case classes to each other is observed by a third learning strategy so that a more compact representation of the whole case base is achieved.

In this paper, we employ for the learning algorithm some basic concepts of a concept formation algorithm described by Fisher [6], see Section 4. Thus, we use a case base evaluation function. This prevents us from the problem of defining threshold for similarity like in numerical clustering [7]. Following the concept of similarity, our evaluation function is based on within case class variance and between case class variance.

Finally, in Section 5 we give some results on the behavior of our learning algorithm based on the images from non-destructive testing.

## 2 The Fundamental Structure of a Case-Based Reasoning System

A case consists of an image, the high-level representation of the image and the interpretation result. Sometimes, also non-image information like image acquisition parameter can be associated to a case [1]. The image needs to be transformed into a high-level representation for the reasoning process. Therefore, image processing and image analysis techniques are necessary for the system [8].

In our system, the cases are attributed graphs, where the nodes are the objects in the image and the attributes associated with the nodes are the features describing the objects. The attributed arcs represent the spatial relation between the objects like left_behind, infront and so one. The spatial relation between two objects is determinable between two objects at any time. Thus does mean the edges between the nodes of the graph do exist. The assumed symmetry of the set of edges is redundant according to the 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=infront”. The attributes can be numerical or categorical in nature. The system is able to handle both types of attributes for similarity determination [2][3].

For our investigations, we used ultra sonic images from non-destructive testing showing defects inside a metallic component like cracks, holes and so on. Figure 1 shows a crack inside metallic components. In a preprocessing phase, the images are binarized (see Fig. 2), filtered, objects are labeled (see Fig. 3) and features for the objects and the spatial relation are extracted [9]. The resulting graph representation shows Fig. 4.

The attributed graph is given as query to the core of the reasoning system. A set of close cases is retrieved from the case base by determining similarity between the attributed graph and the cases in the case base, which are also represented as attributed graphs. Similarity determination is based on subgraph isomorphism detection [2][3]. We adapted an algorithm of Schlesinger et al. [10] to our problem. The algorithm takes into account missing attributes and uncertainty in attribute values by calculating average differences over all attributes of two graphs.

The case base is organized in a hierarchical fashion. This allows finding quickly close cases in a large case base. The present case is classified by the matching function to descend the hierarchy along a path of best matching nodes. The nodes of this hierarchy are labeled with the first observed case graph or after two similar cases have been observed for this node with the prototype calculated from these cases. The associated attributed graph is used for the partial matching.

The close cases together with the interpretation results are shown to the user on display ranked according to the similarity value. The user gives the system a feedback by evaluating the result. This can also be done based on domain knowledge but is not the scope of this paper.

Learning takes place if a new case has been presented to the system. Then, the present case is classified according to the case base. Depending on the classification results the system takes the new case and updates its case base.

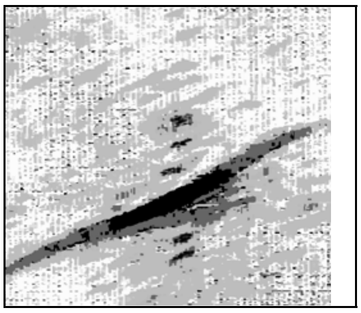


Fig. 1 Original Ultra Sonic Image

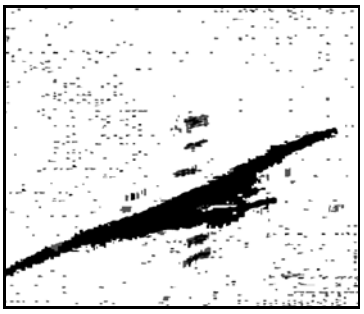


Fig. 2 Binarized Image

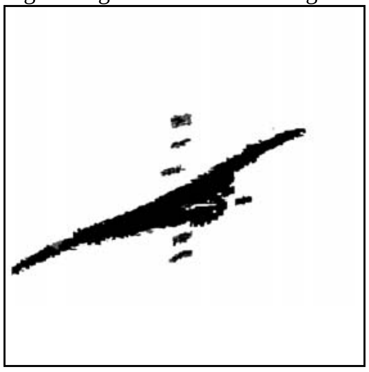


Fig. 3 Labeled Image

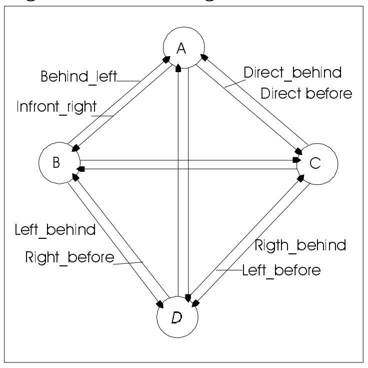


Fig. 4 High Level Representation of an Image

## 3 Incremental Learning

A CBR system acts in an incremental fashion. If the system detects a new case or case class, the learning process will be performed, see Fig. 5.

There are several different learning strategies possible. The simplest one is just learning of new cases, which can be useful for the construction of the complete model. Once a new case has been observed and confirmed by more similar cases the first more generalizing learning strategy start to learn a more compact representative for this class of cases. Closely related to this learning strategy is learning by forgetting. Obviously, bed cases should be detected and dropped from a class of given cases. While learning new case classes and more compact case classes, the relation of these case classes to each other is observed by a third learning strategy so that a more compact representation of the whole case base is achieved [11].

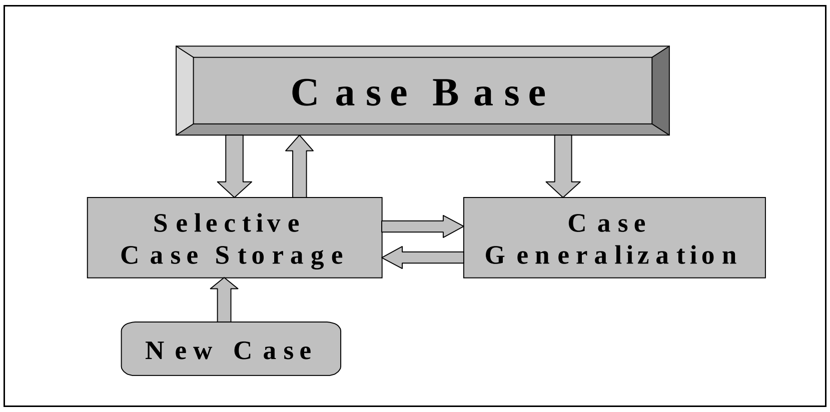


Fig. 5 Case Base Learning Process

## 3.1 Evaluation Function

When several distinct partitions are generated over case base, a heuristic is used to evaluate the partitions. This function evaluates the global quality of single partition and favors partitions that maximize potential for inferring information. In doing this, it attempts to minimize within case class variances and to maximize between case class variances. The employment of an evaluation function prevents us for the problem of defining a threshold for class similarity and inters class similarity from which the proper behavior of the learning algorithm depends. The threshold is usually domain dependent and is not easy to define.

Given a partition *{$C_1$, $C_2$, ... , $C_m$}*. The partition which maximizes the difference between the between case class variance $s_B$ and the within case class variance $s_W$ is chosen as the right partition:

$$SCORE = \frac{1}{m}\left|s_B^{*2} - s_W^{*2}\right| \Rightarrow MAX! \qquad (1)$$

The normalization to *m* (m-the number of partitions) is necessary to compare different partitions.

If $G_{pj}$ is the prototype of the *j-th* case class in the hierarchy at level *k*, $\bar{G}$ the mean graph of all cases in level *k*, and $G^2_{vj}$ is the variance of the graphs in the partition *j*, then follows for SCORE:

$$SCORE = \frac{1}{m}\left|\sum_{j=1}^{m} p_j \left(G_{pj} - \overline{G}\right)^2 - \sum_{j=1}^{m} p_j G^2{}_{vj}\right| \qquad (2)$$

with $p_j$ the relative frequency of cases in the partition $j$.

## 3.2 Learning Strategies

*Learning of New Cases and Case Classes*

It might happen that the reasoning process results in the answer: “There is no similar case in case base”. This indicates that such a case has not seen before. The case needs to be incorporated into the case base in order to complete the present understanding of the domain. The case is classified according to the case base and a new node is opened in the hierarchy of the case base at that position where the part isomorphism relation holds. The new node represents a new case class and the present case is taken as case representative.
The evidence of new case classes increases when new cases where the evaluation measure holds are incorporated into the node. This is considered as learning of case classes.

*Prototype Learning*

When learning a class of cases, cases where the evaluation measure holds are grouped together. The first appeared case would be the representative of the class of these cases and each new case is compared to this case. Obviously, the first appeared case might not always be a good case. Therefore, it is better to compute a prototype for the class of cases.

*Definition 1*

A Graph $G_p = (N_p, p_p, q_p)$ is a prototype of a Class of Cases $C_i = \{G_1, G_2, ..., G_t(N_t, p_t, q_t)\}$ iff $G_p = C_i$ and if there is a one-to-one mapping f: $N_p \rightarrow N_i$ with

(1) $p_p(x_i) = \frac{1}{t}\sum_{n=1}^{t} p_n(f(x_i))$ for all $x_i \in N$ and

(2) $q_p(x_i, y_i) = \frac{1}{n}\sum_{n=1}^{t} q_n(f(x_i), f(y_i))$ for all $x_i, y_i \in N$ .

In the same manner, we can calculate the variance of the graphs in one case class. The resulting prototype is not a case that happened in reality. Therefore, another strategy for calculating a prototype can be to calculate the median of the case in a case class.

*Refinement and Abstraction of Case Classes*

The constructed case classes and the hierarchy are sensitive to the order of case presentation, creating different hierarchies from different order of the same cases. We included already one operation to avoid this problem by learning of prototypes. Two additional operations [12] should help it recover from such nonoptimal hierarchies. At each level of the classification process, the system considers merging the two nodes that best classify the new instance. If the resulting case class is better according to the evaluation function described in Sect. 3.1 than the original, the operation combines the two nodes into a single case class more abstract case class. This transforms a hierarchy of *N* nodes into one having *N-1* nodes, see Fig. 6. This process is equivalent to the application of a "climb-generalization tree" operator [13], with the property that the generalization tree is itself built and maintained by the system. Other merge strategies are described in [14].

The inverse operation is splitting of one node into two nodes, see Fig. 7. This is also known as refinement. At each level, the learning algorithm decides to classify an instance as a member of an existing case class, it also considers removing this case class and elevating its children.

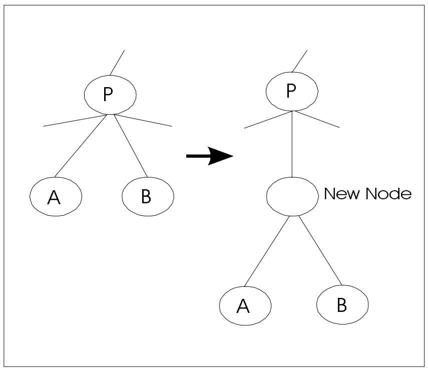


Fig. 6 Node Merging

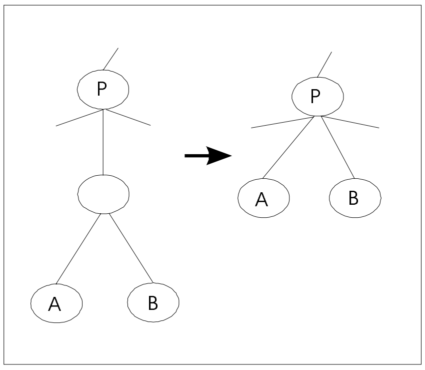


Fig. 7 Node Splitting

If this leads to an improved hierarchy, the algorithm changes the structure of the case base hierarchy accordingly.

## 4 Algorithm

After we have defined our evaluation function and the different learning levels, we can describe our learning algorithm. We adapt the notion of Fisher et. al [6] for concept learning to our problem. If a new case has to be entered into the case base the case is tentatively placed into the hierarchy by applying all the different learning operators described before. The operation, which gives us the highest evaluation score, is chosen (see Figure 9). The new case is entered into the case base and the case base hierarchy is reorganized according to the selected learning operation. The following describes the CBR learning algorithm in detail:

Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.

```
Input:      Case Base Supergraph *CB*
            An unclassified image graph *G* (the case)
Output:     Modified Case Base *CB´*

Top-level call:   Casebase (top-node, G)
Variables:        A, B, C, and D are nodes in the hierarchy
                  K, L, M, and N are partition scores

Casebase (N, G)

If N is a terminal node,
          Then Create-new-terminals (N, G)
               Incorporate (N, G)
          Else For each child A of node N,
                 Compute the score for placing G in A.
                 Compute the scores for all other action with G
                 Let B be the node with the highest score Y.
                 Let D be the node with the second highest score.
                 Let N be the score for placing I in a new node C.
                 Let S be the score for merging B and D into one node.
                 Let I be the score for splitting D into it s children.

                 If  Y is the best score,
                       Then Casebase (P, G) (place G in case class B).
                       Else If N is the best score,
                            Input a new node C
                       Else if S is the best score,
                            Then let  O be merged (B, D, N)
                                 Casebase (N, G)
                       Else if Z is the best score,
                            Then Split (B, N)
                            Casebase (N, G)


Operations over Casebase

Variables: X, O, B, and D are nodes in the hierarchy.
           G is the new case

Incorporate (N, G)
     Update the prototype and the variance of case class N

Create new terminals (N, G)
     Create a new child W of node N.
     Initialize prototype and variance
```

Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.

```
Merge (B, D, N)
    Make O a new child of N
    Remove B and D as children of N
    Add the instances of P and R and all children of B and D to the node O
    Compute prototype and variance from the instances of  B and D

Split (B, N)
    Divide Instances of Node B into two subsets according to evaluation criteria
    Add children D and E to node N
    Insert the two subsets of instances to the corresponding nodes D and E
    Compute new prototype and variance for node D and E
    Add children to node D if subgraph of children is similar to subgraph of node D
    Add children to node E if subgraph of children is similar to subgraph of node E
```

## 5 Results

The system is implemented in C++ and runs on a PC.

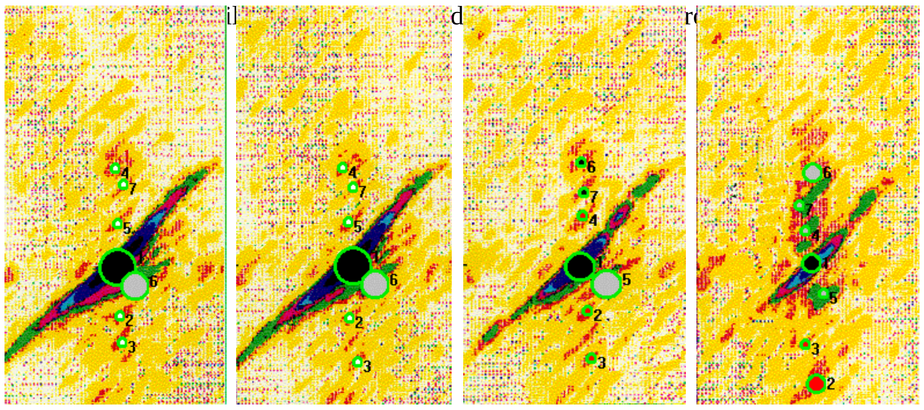
The experiment was made on images from a non-destructive testing domain. The system was given 40 images of one defect type having different subclasses, like e.g. crack-like-defect-pressure-100 or crack-like-defect-presure-50. The images were analyzed by the image analysis procedure described in Section 2 and the resulting attributed graph was given to the system for learning.

The learning process is demonstrated in Fig. 9 on four cases shown in Fig. 8. Case 1 is already contained in the case base and case two should be included in the case base. The learning algorithm tests first, where in the hierarchy the case should be incorporated. The scores of all three options in question show the same results. In such a case, the algorithm favors inserting to an existing node. This prevents us from building a hierarchy with many nodes including only one case. Later the algorithm can split this node if necessary.

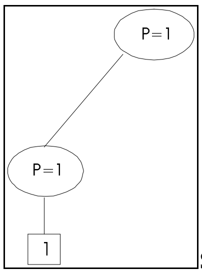
Giving case three to the new case base the algorithm favor to open a new node instead of splitting two nodes.

The new case four is right inserted to the case class having the closest similarity value, see Tab. 1.

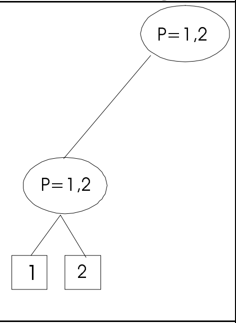
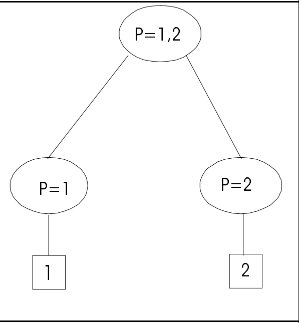
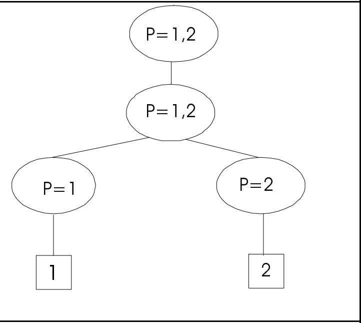
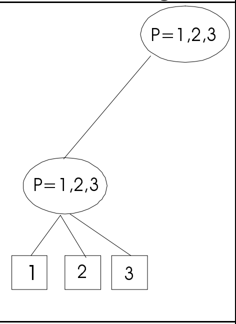
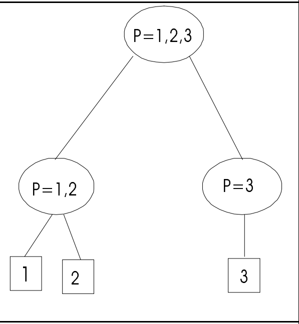
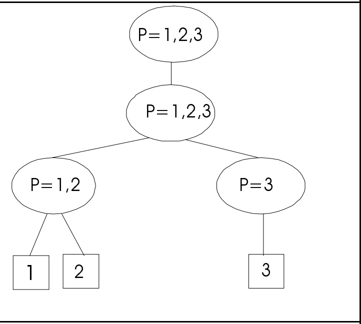
The system was given 40 cases for learning the case base. For evaluating the system, we considered the grouping of the cases into the hierarchy and into the case classes. Based on our domain knowledge we could recognize that meaningful groupings were achieved by the learning algorithm. Further evaluation will be done based on a larger set of cases in future. In addition to that, we will consider the problem of detecting outliers of cases.



Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.



Starting Point

| Insert Case 2 | | |
|---|---|---|
| Insert to existing node | New Node | Refinement |
|  |  |  |
| SB = 0 | SB = 0,00018172 | SB = 0,00018172 |
| SW = 0,00018172 | SW = 0 | SW = 0 |
| SCORE= 0,00018172 | SCORE = 0,00018172 | SCORE = 0,00018172 |
| **Resulting Case Base** | | |
| Insert Case 3 | | |
| Insert to existing node | New Node | Refinement |
|  |  |  |
| SB = 0 | SB = 0,0255671 | SB = 0,0255671 |
| SW = 0,0156513 | SW = 0,0001211 | SW = 0 |
| SCORE = 0,0156513 | SCORE = 0,0254459 | SCORE = 0,0254459 |
| | **** Resulting Case Base **** | |

| Insert Case 4 | | | |
|---|---|---|---|
| Insert to existing node_1 | Insert to existing node_2 | New Node | Refinement |

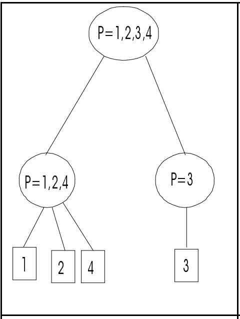
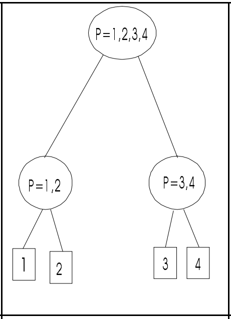
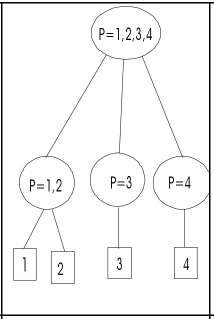
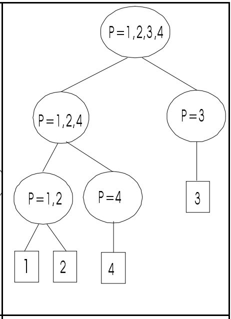
|  |  |  |  |
|---|---|---|---|
| SB = 0,0159367 | SB = 0,0250232 | SB = 0,0218856 | SB = 0,0204 |
| SW = 0,0120498 | SW = 0,0008960 | SW = 0,0000795 | SW = 0 |
| SCORE = 0,0038869 | SCORE = 0,024127 | SCORE= 0,021805 | SCORE = 0,0204 |
| | ***Resulting Case Base*** | | |

Fig. 9 Demonstration of the Learning Process

| | 1 | 2 | 3 | 4 |
|---|---|---|---|---|
| 1 | - | 0.0253968 | 0.1120370 | 0.1977513 |
| 2 | | - | 0.10674603 | 0.2041005 |
| 3 | | | - | 0.1804232 |
| 4 | | | | - |

Tab. 1 Dissimilarity between Image Graphs

## 6 Conclusion

In the paper, we presented different learning strategies for a case-based reasoning system for image interpretation. The CBR system works on structural case representation. These learning strategies fulfill different learning processes necessary in incremental learning of a complete and more compact and robust representation of the domain. It can be observed new cases and case class which helps to get a more complete representation of the domain. Refinement and abstraction processes make the structure of the case base more compact and robust. Experimental results illustrate the performance of the system. Further work will be done on detecting outliers such as distorted cases.

Although we presented our results on ultra sonic image interpretation from non-destructive testing, the described method can be used for other domain where structural representations are used like e.g. interpretation of technical drawings or interpretation of industrial parts.

Proc. in Springer Verlag P. Perner, Different Learning Strategies in a Case-Based Reasoning System for Image Interpretation, Advances in Case-Based Reasoning, B. Smith and P. Cunningham (Eds.), LNAI 1488, Springer Verlag 1998, S. 251-261.

## References

1. P. Perner, Case-Based Reasoning for Image Interpretation in Non-Destructive Testing, In Proceedings EWCBR-93, volume 2, pages 403-409, 1993
2. P. Perner and W. Paetzold, An Incremental Learning System for Interpretation of Images, SSPR´94, Oct. 1994 Naharya, In: Shape, Structure, and Pattern Recognition, D. Dori and A. Bruckstein (Eds.), World Scientific Publishing Co., 1995, pp. 311-323.
3. P. Perner and W. Paetzold, An Incremental Learning System for Image Interpretation, HTWK Report 5/93
4. P. Perner, P. Anderson, D. Summner, J. Kyle, An Application of Case-based Reasoning in Test Process Diagnosis , IBM International Conference on Expert Systems, Proc. IBM ITL Conference on Expert Systems, Yorktown Heights USA, pp 73-85, Oct. 1992, Plenary Talk
5. K. Tombre, "Structural and Syntactic Methods in Line Drawing Analysis: To Which Extent Do They Work?," In: P. Perner, P. Wang, and A. Rosenfeld, Advances in Structral and Syntactic Pattern Recognition, Springer Verlag 1996, Lncs 1121
6. D.H. Fisher, "Knowledge Acquisition via Incremental Clustering," Machine Learning, 2: 139-172, 1987.
7. A.K. Jain and R.C. Dubes, Algorithms for Clustering Data, Prentice Hall, New Jersey1988.
8. H. Niemann, Pattern Analysis and Understanding, Second Edition, Springer Verlag, 1990.
9. P. Perner," Ultra Sonic Image Interpretation for Non-Destructive Testing," IAPR Workshop on Machine Vision Applications, Tokyo, Japan, Nov. 1996, p. 552-554.
10. M.I. Schlesinger, Mathematical Tools of Picture Processing, Naukowa Duma, Kiew 1989.
11. P. Perner, Case-Based Reasoning for Image Interpretation, In: Computer Analysis of Images and Patterns, V. Hlava_ and R. Šárka, Springer Verlag 1995, pp. 532-537
12. J.H. Gennari, P. Langley, and D. Fisher, "Models of Incremental Concept Formation," Artificial Intelligence 40 (1989) 11-61.
13. R.S. Michalski. A theory and methodology of inductive learning. In R. S. Michalski, J.G. Carbonell, and T.M. Mitchell, (Eds.), Machine Learning: Artificial Intelligence Approach. Morgan Kaufmann, 1983.
14. Manuela M. Veloso, „Merge Strategies for Multiple Case Plan Replay," Proceedings ICCBR-97.