P. Perner, Why Case-Based Reasoning is Attractive for Image Interpretation, D. Aha and I. Watson (Eds.), Case-Bases Reasoning Research and Developments, Springer Verlag 2001, lnai 2080, p. 27-44. (invited paper)

Why Case-Based Reasoning is Attractive for Image Interpretation

Petra Perner

Institute of Computer Vision and Applied Computer Sciences
Arno-Nitzsche-Str. 45, 04277 Leipzig
ibaiperner@aol.com http://www.ibai-research.de

Abstract. The development of image interpretation systems is concerned with tricky problems such as a limited number of observations, environmental influence, and noise. Recent systems lack robustness, accuracy, and flexibility. The introduction of case-based reasoning (CBR) strategies can help to overcome these drawbacks. The special type of information (i.e., images) and the problems mentioned above provide special requirements for CBR strategies. In this paper we review what has been achieved so far and research topics concerned with case-based image interpretation. We introduce a new approach for an image interpretation system and review its components.

1 Introduction

Image interpretation systems are becoming increasingly popular in medical and industrial applications. The existing statistical and knowledge-based techniques lack robustness, accuracy, and flexibility. New strategies are necessary that can adapt to changing environmental conditions, user needs and process requirements. Introducing case-based reasoning (CBR) strategies into image interpretation systems can satisfy these requirements. CBR provides a flexible and powerful method for controlling the image processing process in all phases of an image interpretation system to derive information of the highest possible quality. Beyond this CBR offers different learning capabilities, for all phases of an image interpretation system, that satisfy different needs during the development process of an image interpretation system. Therefore, they are especially appropriate for image interpretation.

Although all this has been demonstrated in various applications [1]-[6][35], case-based image interpretation systems are still not well established in the computer vision community. One reason might be that CBR is not very well known within this community. Also, some relevant activities have been shied away from developing large complex systems in favor of developing special algorithms for well-constrained tasks (e.g., texture, motion, or shape recognition). In this paper, we will show that a CBR framework can be used to overcome the modeling burden usually associated with the development of image interpretation systems.

We seek to increase attention for this area and the special needs that image processing tasks require. We will review current activities on image interpretation and describe our work on a comprehensive case-based image interpretation system.

In Section 2, we will introduce the tasks involved when interpreting an image, showing that they require knowledge sources ranging from numerical representations to sub-symbolic and symbolic representations. Different kinds of knowledge sources need different kinds of processing operators and representations, and their integration places special challenges on the system developer.

In Section 3, we will describe the special needs of an image interpretation system and how they are related to CBR topics. Then, we will describe in Section 4 the case representations possible for image information. Similarity measures strongly depend on the chosen image representation. We will overview what kinds of similarity measures are useful and what are the open research topics in Section 5. In Section 6, we will describe our approach for a comprehensive CBR system for image interpretation and what has been achieved so far. Finally, we offer conclusions based on our CBR systems working in real-world environments.

2 Tasks an Image Interpretation System Must Solve

Image interpretation is the process of mapping the numerical representation of an image into a logical representation such as suitable for scene description. An image interpretation system must be able to extract symbolic features from the pixels of an image (e.g., irregular structure inside the nodule, area of calcification, and sharp margin). This is a complex process; the image passes through several general processing steps until the final symbolic description is obtained. These include image preprocessing, image segmentation, image analysis, and image interpretation (see Figure 1). Interdisciplinary knowledge from image processing, syntactical and statistical pattern recognition, and artificial intelligence is required to build such systems. The primitive (low-level) image features will be extracted at the lowest level of an image interpretation system. Therefore, the image matrix acquired by the image acquisition component must first undergo image pre-processing to remove noise, restore distortions, undergo smoothing, and sharpen object contours. In the next step, objects of interest are distinguished from background and uninteresting objects, which are removed from the image matrix.

In the x-ray computed tomography (CT) image shown in Figure 1, the skull and the head shell is removed from the image in a preprocessing step. Afterwards, the resulting image is partitioned into objects such as brain and liquor. After having found the objects of interest in an image, we can then describe the objects using primitive image features. Depending on the particular objects and focus of interest, these features can be lines, edges, ribbon, etc. A geometric object such as a block will be described, for example, by lines and edges. The objects in the ultrasonic image shown in Figure 1 are described by regions and their spatial relation to each other. The region’s features could include size, shape, or the gray level. Typically, these low-level features have to be mapped to high-level features. A symbolic feature such as *fuzzy margin* will be a function of several low-level features. Lines and edges will be grouped together by perceptual criteria such as collinearity and continuity in order to describe a block.

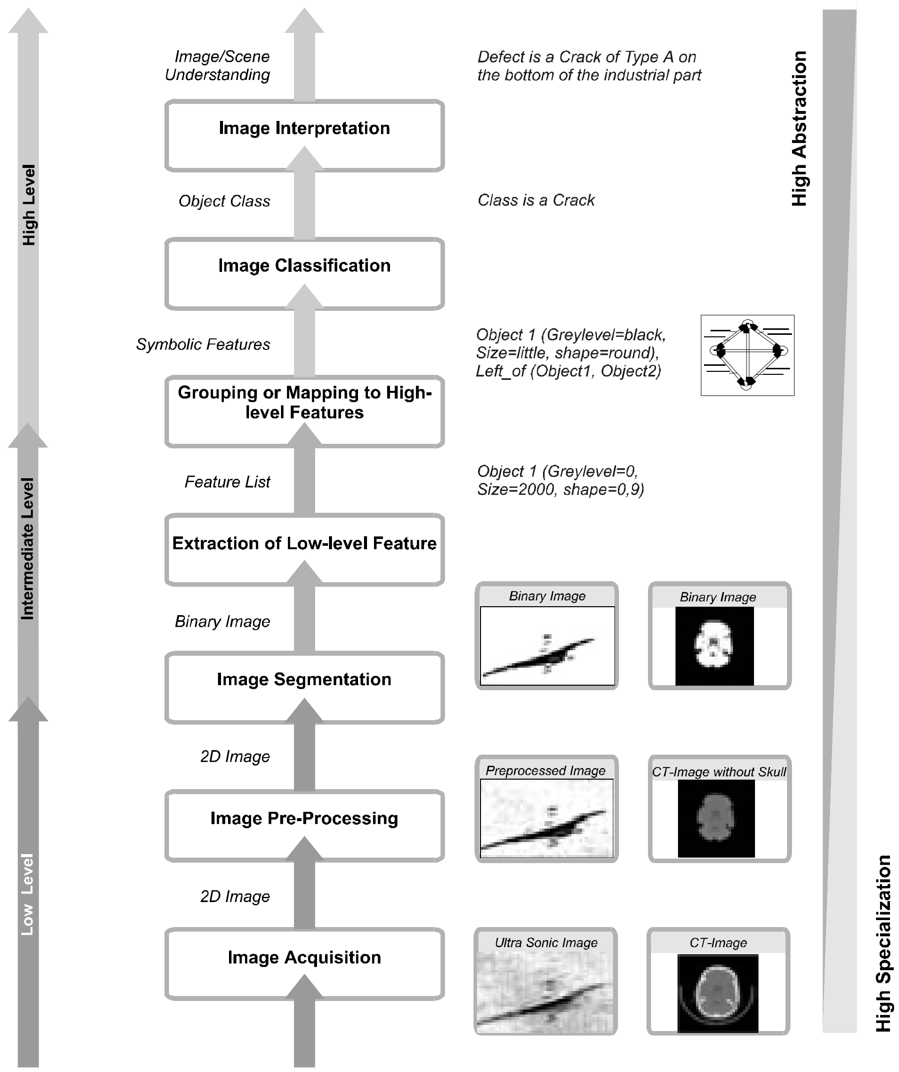

Fig. 1. Architecture of an Image Interpretation System

Image classification is usually referred to as the mapping of numeric features to predefined classes. Sometimes image interpretation requires only image classification. However, image classification is frequently only a *first* step of image interpretation. Low-level features or part of the object description are used to classify the object into different object classes in order to reduce the complexity of the search space. The image interpretation component identifies an object by finding the object that it belongs to (among the models of the object class). This is done by matching the symbolic description of the object in the scene to the model of the object stored in the knowledge base. When processing an image using an image interpretation system, an image's content is transformed into multiple representations that reflect different abstraction levels. This incrementally removes unnecessary detail from the image. The highest abstraction level will be reached after grouping the image's features. It is a product of mapping the image pixels contained in the image matrix into a logical structure. This higher level representation ensures that the image interpretation process will not be affected by noise appearing during image acquisition, and it also provides an understanding of the image's content. A bottom-up control structure is shown for the generic system in Figure 1. This control structure allows no feedback to preceding processing components if the result of the outcome of the current component is unsatisfactory. A mixture of bottom-up and top-down control would allow the outcome of a component to be refined by returning to previous component.

3 Development Concerns

Several factors influence the quality of the final result of an image interpretation system, including environmental conditions, the selected imaging device, noise, the number of observations from the task domain, and the chosen part of the task domain. These cannot often all be accounted for during system development, and many of them will only be discovered during system execution. Furthermore, the task domain cannot even be guaranteed to be limited. For example, in defect classification for industrial tasks, new defects may occur because the manufacturing tool that had been used for a long period suddenly causes scratches on the surface of the manufactured part. In optical character recognition, imaging defects (e.g., heavy print, light print, or stray marks) can occur and influence the recognition results. Rice *et al.* [7] attempted to systematically overview the factors that influence the result of an optical character recognition system, and how different systems respond to them. However, it is not yet possible to observe all real-world influences, nor provide a sufficiently large enough sample set for system development and testing.

A robust image interpretation system must be able to deal with such influences. It must have intelligent strategies on all levels of an image interpretation system that can adapt the processing components to these new requirements. A strategy that seems to satisfy these requirements could be case-based reasoning. CBR does not rely on a well-formulated domain theory, which is, as we have seen, often difficult to acquire.

This suggests that we must consider different aspects during system development that are frequently studied CBR issues. Because we expect users will discover new aspects of the environment and the objects during system usage, an automatic image interpretation system should be able to incrementally update the system's model, as illustrated in Figure 2. This requires knowledge maintenance and learning. The designated lifetime of a case also plays an important role. Other aspects are concerned with system competence. The range of target problems that a given system or algorithm can solve are often not quite clear to the developer of the image interpretation system. Often researchers present to the community a new algorithm that can, for example, recognize the shape of an object in a particular image and then claim that they have developed a model. Unfortunately, all too often another researcher inputs a different image to the same algorithm and finds that it fails. Did the first researcher develop a model or did they instead develop a function? Testing and evaluation of algorithms and systems is an important problem in computer vision [8], as is designing the algorithm's control structure so that it fits best to the current problem. CBR strategies can help to solve this problem in computer vision.

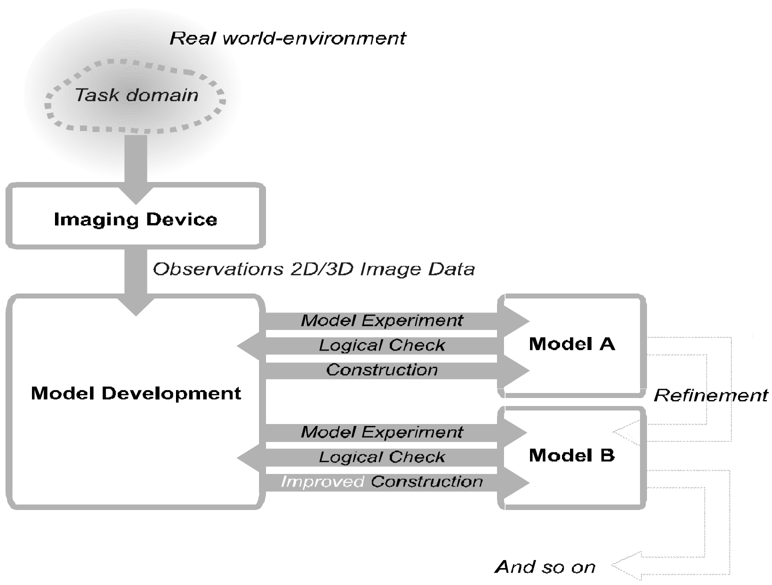

Fig. 2. Model Development Process

4 Case Representations for Images

Usually the main types of information concerned with image interpretation are image-related and non-image-related information. Image-related information can be the 1D, 2D, or 3D images of the desired application, while non-image-related information can include information about image acquisition (e.g., the type and parameters of the sensor, information about the objects, or the illumination of the scene). The type of application determines what type of information should be considered for image interpretation. For medical CT image segmentation [3], we used patient-specific parameters such as age, sex, slice thickness, and number of slices. Jarmulak [1] considered the type of sensor for a railway inspection application and his system used it to control the type of case base that the system used during reasoning.

How the 2D or 3D image matrix is represented depends on the application the developer's point of view. In principle it is possible to represent an image using one of the abstraction levels described in Section 2. An image may be described by the pixel matrix itself or by parts of this matrix (a pixel representation). It may be described by the objects contained in the image and their features (a feature-based representation). Furthermore, it can be described by a more complex model of the image scene comprising objects and their features as well as the object's spatial relationships (an attributed graph representation or semantic network).

As mentioned earlier, processing the image through multiple components and describing it by higher-level representations can reduce the number unnecessary details in its representation. This allows more noise tolerance and may speed up the retrieval process but may require additional modeling of the image content, which is difficult and time-consuming. Also, it requires processing steps that are often computationally intensive. Thus, the necessary abstraction level of the image information should be carefully chosen.

Jarmulak [1] solved this problem by using a four-level case hierarchy and different case bases for different sensor types. Stored at the lowest level of the hierarchy are the objects described by features such as their location, orientation, and type (line, parabola, or noise) parameters. The next level consists of objects of the same channel within the same subcluster. In the following level the subcluster is stored and the highest level stores the entire image scene. This representation allows cases to be matched on different granularity levels. Because the entire scene may have noise distortions and imprecise measurements, the influence of noise can be reduced by retrieving cases on these different levels.

Grimnes and Aamodt [2] developed a model-based system for the interpretation of abdominal CT images. The image's content was represented by a semantic network where concepts can be a general concept, a case, or a heuristic rule. Poorly understood parts of the model are expressed by cases and can be revised during system usage by the learning component. The combination of a partial well-understood model with cases helps to overcome the usual burden of modeling. The learning component is based on failure-driven learning and case integration. Non-image information is also stored such as sex, age, earlier diagnosis, and social condition.

In both of these systems, CBR is used only for the high-level component. We have studied different approaches for the different processing stages of an image interpretation system. For image segmentation [1], we studied a pixel-based approach and also a feature-based approach that described an image's statistical properties. Our results show that the pixel-based approach can yield better image segmentation. For the high-level approach in an ultra sonic image interpretation system, we used a graph representation [9].

Micarelli *et al.* [4] have also calculated image properties from images and stored them into a case base. They used the Wavelet transform because it is scale-independent, but this limits their similarity measure to consider only object rotation.

Representing images at multiple levels of abstraction presents some technical challenges. When representing an image with a high-level abstraction rather than the image matrix itself, some information will be lost. Abstraction requires deciding which details of an image are necessary. If only some objects are seen at one time, then we might think that one detail is not of interest since our decision is based on a limited number of objects. This can cause problems. Therefore, storing the images themselves is always preferable but requires high storage capacity. Also, the different representations at each abstraction level require different similarity measures.

5 Similarity Measures for Image Interpretation

Images can be rotated, translated, different in scale, or may have different contrast and energy yet still considered to be similar. In contrast, two images may be dissimilar because the object in one image is rotated by 180 degrees. The concept of *invariance* in image interpretation is closely related to that of similarity. A good similarity measure should take this into consideration.

Classical similarity measures do not consider invariance. Usually, the images or the features have to be pre-processed in order to be adapted to the scale, orientation, or shift. This process is an additional and expensive processing step that needs some a priori information, which is not always given. Matched, linear, Fourier, and Wavelet filters are especially useful for invariance under translation and rotation [4]. There has been a lot of work done to develop such filters for image interpretation. The best way to achieve scale invariance from an image is by means of invariant moments, which can also be invariant under rotation and other distortions. Some additional invariance can be obtained by normalization (to reduce the influence of energy).

Depending on the image representation (see Figure 3) we can divide similarity measures into:

1. pixel (Iconic)-matrix similarity measures;
2. similarity measures for comparing strings;
3. feature-based similarity measures (numeric, symbolic, or mixed type); and,
4. structural similarity measures.

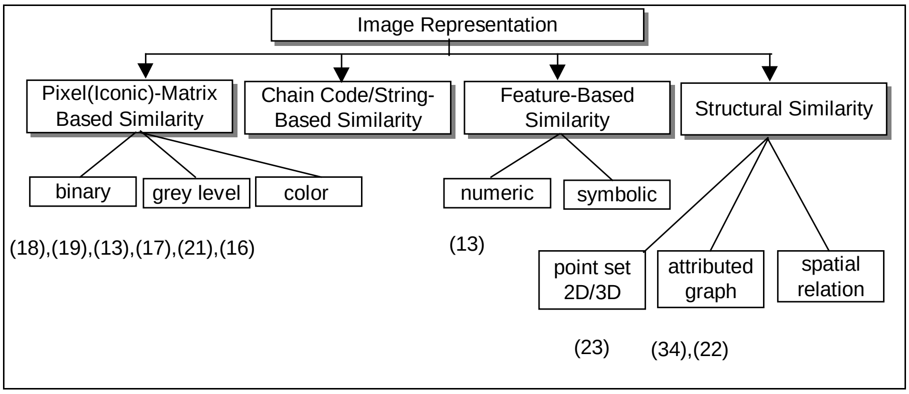

Fig. 3. Image Representation and Similarity Measures

Because a CBR image interpretation system must also account for non-image information (e.g., about the environment or the objects), similarity measures are needed that can combine non-image with image information. In [10], we described a first approach for doing this.

Systematic studies on image similarity have been conducted by Zamperoni and Starovoitov [11]. They studied how pixel-matrix similarity measures behave under different real-world influences such as translation, noise (spikes, salt and pepper noise), and different contrast. Image feature-based similarity measures have been studied from a broader perspective by Santini and Jain [12]. To our knowledge, these are the only comprehensive studies on image similarity. Otherwise, every new conference on pattern recognition contains proposals for new similarity measures for specific purposes and different kinds of image representation [13]-[23]. While there was some simultaneous research on image similarity in the CBR community (e.g., [24]), this work has also not achieved new insight. In our view, images are a special type of information source that require special similarity measures, and these measures require more rigorous analysis.

6 A Case-Based Image Interpretation System

We proposed an architecture (Figure 5) that uses CBR on all levels of an image interpretation system in [9]. The system subdivides into a run-time part and a maintenance and learning part. During run-time, the system uses CBR strategies to reason over images while the maintenance and learning part attempt to improve system performance off-line. We are further developing this system based on an application that is called HEp-2 cell image analysis [25] (Figure 4). This kind of cell is used to identify antinuclear autoantibodies (ANA) . HEp-2 cells can recognize over 30 different nuclear and cytoplasmatic patterns, which are given by upwards of 100 different autoantibodies. This exemplifies the difficulty with this application. We have to recognize a large number of different patterns that are neither well described nor fixed in number. Furthermore, we cannot exclude the possibility of new patterns occurring.

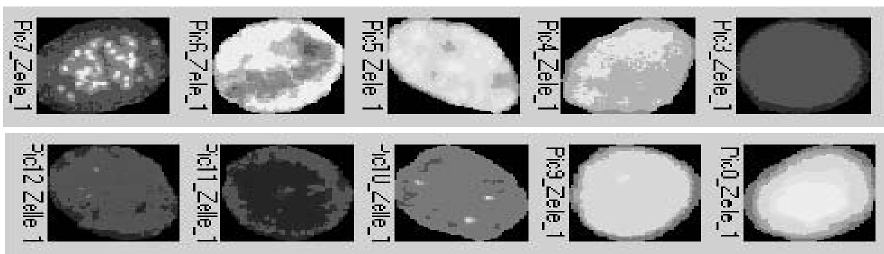

Fig. 4. Some Example Images of HEp-2 Cells

6.1 Image Segmentation

Most CBR image interpretation systems (e.g., [2][6]) select among different image processing chains but they do not control the algorithm itself. This in accordance with most knowledge-based image interpretation systems described in the computer vision literature, which select a processing chain that best fits the current image analysis problem. This approach requires a large enough library of image processing procedures and special image processing knowledge.

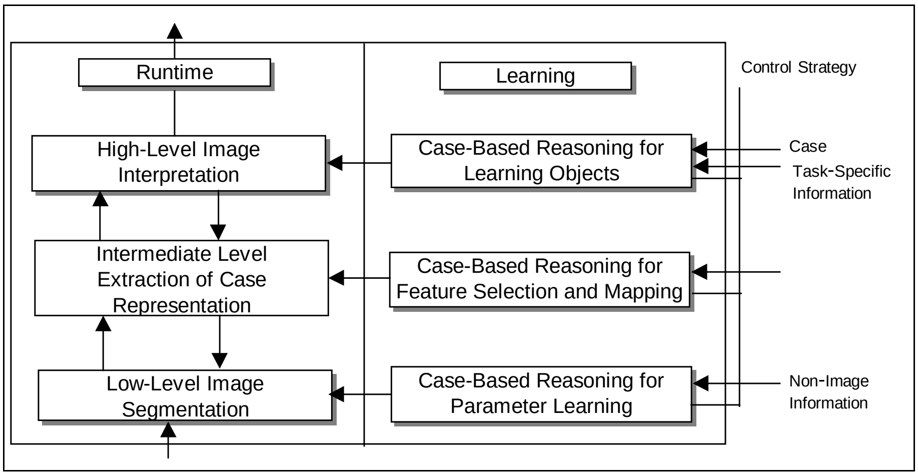

Fig. 5. Architecture of a Case-Based Image Interpretation System

However, modern segmentation techniques contain numerous control parameters, which can be adjusted to obtain optimal performance. Parameter selection should be done using a sufficiently large test data set that represents the entire domain well enough to support a general segmentation model. However, obtaining a suitable test set is often impossible, which means that the segmentation model does not fit the data well and must be adjusted to new data. Also, a general model does not guarantee the best segmentation for each image, but instead it guarantees an average best fit over the entire set of images. Finally, differing image quality (e.g., caused by variations in environmental conditions, image devices) requires adapting the segmentation process accordingly. This necessitates equipping the segmentation component with learning capabilities, which can incrementally acquire segmentation model knowledge.
We use a case-based approach for parameter learning, in which formerly processed cases contain their original images, their non-image information (e.g., image acquisition parameters, object characteristics), and their image segmentation parameters. Finding the best segmentation for the current image is done by retrieving similar cases from the case base. Similarity is computed using non-image and image information. The evaluation component will use the most similar case for further processing. If two or more cases have the same highest similarity score then the first of these cases is used. The image segmentation parameter associated with the selected case will then be given to the image segmentation component, which will segment the current image (see Figure 6). Images with similar image characteristics are assumed to yield similar good segmentation results when the same segmentation parameters were applied to these images. Superior performance for this approach has been demonstrated for CT image segmentation [3]. This approach is sufficiently flexible to be used for other applications and will therefore be used for Hep-2 cell image analysis.

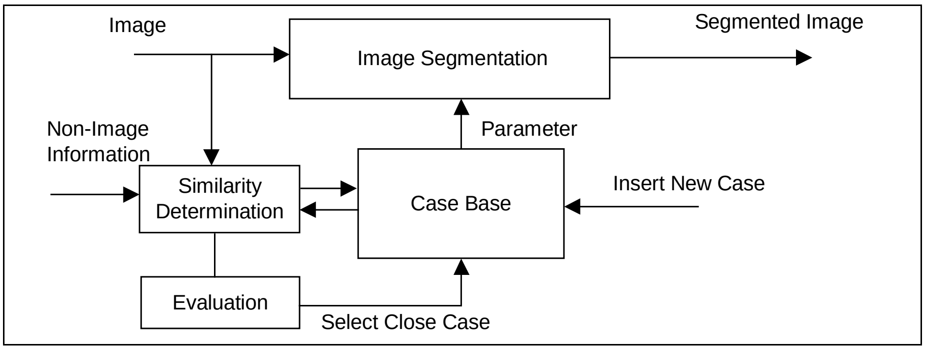

Fig. 6. Case-Based Image Segmentation Component

6.2 Feature Selection

Feature selection is concerned with learning the most important (symbolic) features, while feature extraction is responsible for locating those features in the image and finding their values. From the preprocessed, segmented, and labeled 1-D, 2-D, or 3-D image matrix we can extract low-level or primitive image features that are corners, extended edges, textured regions, ribbons, the 2 1/2-D sketch, and semantic clusters of edges and regions. The number of primitive features that can be extracted from the image content is limited (e.g., color, gray level, spatial relations, motion). Understanding the image's content requires mapping those primitives to the desired symbolic features. In current approaches to image interpretation, performance degrades when new objects are encountered that may require the extraction of "shape primitives" not known to the system. To overcome the bottleneck of predetermined and static object features, automatic acquisition of new features using a learning approach is necessary, particularly for flexible image interpretation processes.

Therefore, we introduced for our system a library of feature extractors that can calculate all possible features. In the next step, the system selects from these features the necessary features describing the desired symbolic feature.

6.3 Signal-to-Symbol Mapping and Feature Selection

It is seldom the case that one low-level feature describes the right meaning of one quality of an object. Often a combination of a few low-level features is necessary to express a symbolic feature like *fine speckled*, which is a combination of low-level features such as number of small objects, object sizes, and their gray-level. In these situations, a mapping of (n) low-level features to the symbolic feature is needed. This problem is concerned with the selection of the right features (feature selection), their parameters, and the creation of a mapping function (classification function).

The problem here is to select this subset of features from a large/complex feature set that represent best the symbolic feature by means of classification accuracy or intra/inter class distance, see Fig. 7. To solve this problem, we use an induced decision tree [26]. This approach acts as feature filter for the image interpretation process. Once a new feature is discovered the low-level features are calculated from the image and labeled by the symbolic feature. The prototypes of the other features are taken and applied together with data from the new feature to the induction algorithm. The resulting set of rules are used as a feature selector.

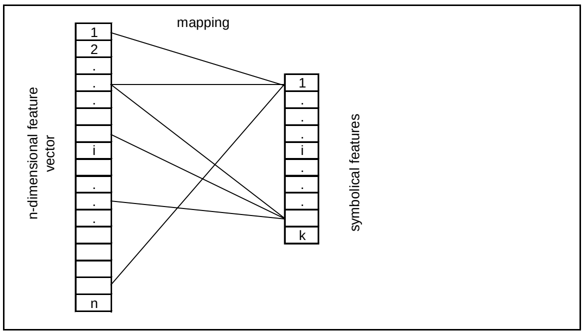

Fig. 7. Low-Level Feature Selection and Mapping to Symbolic Features

6.4 Image Interpretation

The case representation of an image's high-level information can differ among images. This ranges among semantic networks, graphs, and decision trees. Image interpretation problems always have some hidden taxonomy that, if discovered, can be used to help model the problem. An ultrasonic image showing a defect type *crack* might show a crack of a specific subclass such as *crack_under_pressure_x*. To classify this type of crack as a specific subtype might prevent the class *crack* from having large variations, which can help to improve classification results.

To discover these concepts we have found decision tree induction [26] and incremental conceptual clustering [27][36] very suitable. Based on the available cases we used C4.5 to induce a tree for indexing the case base. Our approach differs from 's [1], who also induced a tree for case indexing, in that we will incrementally update the tree structure based on newly discovered cases. Leaves in the tree where no class overlap occurs will remain as terminal leaves, while a leaf with class overlap will be pruned back until a predefined number of samples remain in the group covered by this leaf.

The query case may be clustered through the tree until it reaches a leaf node. If the leaf node is labeled with its class, then that class is assigned to the query. If it not a final node then similarity will be calculated between all cases belonging to this node. We do not divide these cases into clusters but instead incrementally update the index structure when entering a new case.

7 Maintenance & Learning

An important focus of recent CBR research is on how to develop strategies for obtaining compact, competent case-bases, as a way to improve the performance of CBR systems [28]. Although maintenance approaches have not yet been extensively studied for image interpretation systems, they will play an important. Grimnes and Aamodt [2] mention that maintaining the case base in ImageCreek is complex, and that knowledge base maintenance methods are crucial for making the architecture scale. The main problem is handling the different types of knowledge. Jarmulak [1] takes into account case addition, splits the clusters into groups of fixed size, and represents them using a prototype to speed up the matching process. Perner [3][9] takes into account case addition, learning of case classes and prototypes, and higher order constructs. We focus here on topics that, until now, have only been addressed as more specific problems.

7.1 Case Addition and Case Deletion

Case deletion in a pre-determined time window based on failure results [29] might not be appropriate for image interpretation because a failure might mean that, instead of the retrieved case being erroneous, there is some relevant knowledge that we could not describe using features. Also, cases that occur infrequently (i.e., that have not been used recently) should be recognized by the system.

The causes for case deletion or addition might differ from other CBR applications:

1. Since images may be distorted and very noisy it might not be useful to store distorted representations. Determining which representation is distorted is sometimes not easy even if you have seen only a few images, and it is usually necessary to have domain knowledge that must also be built up over time.
2. Imprecise or noisy measurements can be caused by some defects of illumination, the image acquisition device, or the object itself. If the image analysis cannot adapt to these measurements, or the reasoning process cannot handle it, then this might cause failure results. However, if this is a systematic event then it might be worthwhile to store the recent case in the case base.
3. The last fact comes from the real world environment. It is not possible to determine all real world influences a priori.

Thus, developers prefer to incorporate cases into a case base instead of forgetting them. Although case bases can grow very large, instead of forgetting cases, we would rather subdivide the case base into frequently vs. rarely used cases. This requires addressing the issue of how should the addition of cases into one of these two case bases be controlled, as well as their respective reasoning processes.

7.2 Case Acquisition Tool

A CBR system for image classification needs to have some particular features with respect to images. These features result from:

- special requirements of visual knowledge acquisition (image-language problem) and
- the need to transform an image's numerical data into a symbolic description.

The main problem with images and their translation into a language is that the knowledge about an image is usually tacit. To make this knowledge explicit is often hard. Sometimes the meaning of a word does not correspond to the correct meaning of the image. Therefore, it is necessary to support the operator in an efficient way.

Most case-based image interpretation systems do not pay attention to this problem. The only functionality these systems provide is visualization of the image or the processed image. Usually, new case knowledge is obtained via manual acquisition with the expert. This is a time-consuming and sometimes boring process for both the system developer and the expert.

A CBR system for image interpretation should have a special case acquisition tool, such as the one detailed in [30]. By using a questioning strategy and evaluating the answers given by the expert, the expert or operator is forced to specify the right knowledge for image interpretation. The questioning strategy is designed to force an expert to explain what distinguishes one object from another and to specify the right property for the object.

Recently, this problem has received attention for e-commerce applications. Automatic questioning strategies are important for acquiring customer requirements in e-commerce applications [31] because the customer acts alone on the net.

A special case acquisition tool for image segmentation was described in [3]. With the help of this tool the user can control the parameters of the image segmentation algorithm. Simultaneously, he can view the segmented image and, if he is satisfied with the segmentation quality, he can store the parameters of the image segmentation algorithm together with the case description in the case base.

7.3 Competence of Case Bases

An important problem in image interpretation concerns system competence. We follow the definition in [32] and define the competence of a system as the range of target problems the system can solve. As we have already pointed out in Section 3 it is often not clear to the computer vision community what problems the desired algorithm can solve. So we have to find a way to describe the competence of a system. This differs from what is usually understood about this problem in CBR. Competence is described based on statistical properties such as case-base size, density and distribution, or group coverage and group density. But what if some groups overlap? Smyth and McKenna [32] argue that these groups have shared competence and can be linked together in some way. However, we can also view it as having a poor description of the target problem. Based on this description we may retrieve a similar case but its solution application to the query image may be low in quality. By investigating the failure we may learn that we did not consider a property of the environment or maybe we could not specify it because it was not contained in the description of the target problem. Therefore, the system performance decreases. The measures described in [32] and [33] only view competence based on the coverage of the problem space. How do we know that cases in group 1 and group 2 belong to the same target problem group? Proximity in problem space does not imply that they belong to the same problem group; misclassifications can occur because the patterns overlap. We argue that system competence must also account for the misclassification of the target problem based on the problem description.

7.4 Control Strategies and Monitoring System Performance

An important issue in maintaining an image interpretation system involves the controlling and monitoring of system performance. The system is a complex system comprising different processing components (e.g., image analysis, feature extraction and high-level image interpretation). The quality of the results of one component strongly depends on the quality of a preceding component. Several possible strategies exist for improving system performance.

Control without Feedback (Local Optimization)

The simplest approach is to adjust the performance of each component without considering the others. Each component - segmentation, feature extraction and selection, and interpretation - acts alone. No interaction between them is allowed. Image segmentation performance may be determined by subjective evaluation of the segmentation result as done by an expert, by calculating the similarity between the original and segmented images, by interclass distances for feature extraction, or by classification error. This strategy has the advantage that the control of the system is simple. Nevertheless, it cannot optimize system performance because only local optimums can be achieved for each single component.

Control with Feedback (Global Optimization)

If after local optimization the performance of a component could not be improved or is not satisfactory, the control algorithm will lead the learning process to the preceding processing component in an attempt to further improve its performance. This process stops if the first processing component is reached and if no improvement could be established after local optimization.

The logical scheme in Table 1 shows us how control is guided. If the performance of all components is good, no action has to be taken. If the interpretation component's performance is poor, then its performance needs to be optimized. We assume that it is impossible for a preceding component to perform poorly while its successor components perform well.

8. Conclusion

We surveyed special topics associated with a case-based image interpretation system. From our point of view case-based image interpretation differs in many aspects from other CBR applications that require further investigation. First, more systematic work on special image similarity measures is needed that investigates the measures under different influences that may occur in an image. Next, case representations are required for all the different abstraction levels of an image. Finally, the maintenance and learning strategies must be defined so that they can help to improve the system performance and discover the range of target problems that the system can solve.

We have recently deployed two CBR image interpretation systems. One is installed at the university hospital in Halle; it is used for image segmentation to determine the brain/liquor ratio of the head in a CT image. The second system is used to interpret ultra-sonic SAFT images. In both applications the CBR strategies we used achieved good system performance that satisfied the users and outperformed other systems. The learning and maintenance facilities installed to date have been particularly well-received.

In summary, we believe that investigations of case-based image interpretation systems can reveal special challenges to both the CBR and computer vision communities, and encourage more people to work on this topic.

Table 1 Logical Scheme of Performance Control

Segmentation (S)	Feature Extraction (FE)	Interpretation (I)	Action
Good	Good	Good	No Action
Good	Good	Poor	Optimize I
Good	Poor	Good	Impossible
Good	Poor	Poor	Optimize FE and examine effects on I
Poor	Good	Good	Impossible
Poor	Good	Poor	Impossible
Poor	Poor	Good	Impossible
Poor	Poor	Poor	Optimize S, then re-examine the performance of the other components

Acknowledgements

We would like to thank M. Richter for helpful discussions on this topic. Special thanks to David Aha for his kind advice on this topic and for his helpful comments that significantly improved earlier versions of this paper. The implementation of the case-based image segmentation on a parallel computer has been funded by the EU project (HP-ISIS). The CBR system for Hep-2 cells is funded within in the project LernBildZell by the German Ministry of Economy .

References

1. Jarmulak, J. (1998). Case-based classification of ultrasonic B-Scans: Case-base organisation and case retrieval. In B. Smyth & P. Cunningham (Eds.) *Advances in Case-Based Reasoning* (pp. 100-111). Berlin: Springer Verlag.
2. Grimnes, M. & Aamodt, A.(1996). A two layer case-based reasoning architecture for medical image understanding, In I. Smith & B. Faltings (Eds.) *Advances in Case-Based Reasoning* (pp. 164-178). Berlin: Springer Verlag.
3. Perner, P. (1999). An architecture for a CBR image segmentation system. *Journal of Engineering Application in Artificial Intelligence*, *12*(6), 749-759.
4. Micarelli, A. Neri, A., & Sansonetti, G. (2000). A case-based approach to image recognition, In E. Blanzieri & L. Portinale (Eds.) *Advances in Case-Based Reasoning* (pp. 443-454). Berlin: Springer Verlag.
5. Venkataraman, S., Krishnan, R., & Rao, K.K. (1993). A rule-rule-case based system for image analysis. In M.M. Richter, S. Wess, K.D. Althoff, & F. Maurer (Eds.) *First European Workshop on Case-Based Reasoning* (Technical Report SFB 314) (pp. 410-415). Kaiserslautern, Germany: University of Kaiserslautern.
6. Ficet-Cauchard, V., Porquet, C., & Revenu, M. (1999). CBR for the reuse of image processing knowledge: A recursive retrieval/adaption strategy. In K.-D. Althoff, R. Bergmann, & L.K. Branting (Eds.) *Case-Based Reasoning Research and Development* (pp. 438-453). Berlin: Springer.
7. Rice, S.V., Nagy, G., & Nartker, T.H. (1999). *Optical character recognition: An illustrated guide to the frontier*. London: Kluwer.
8. Klette, R., Stiehl, H.S., Viergever, M.A., & Vincken, K.L. (2000). *Performance characterization in computer vision.* London: Kluwer
9. Perner, P. (1998). Using CBR learning for the low-level and high-level unit of a image interpretation system. In S. Singh (Ed.) *Advances in Pattern Recognition* (pp. 45-54). Berlin: Springer Verlag.
10. Perner, P. (1999). An architecture for a CBR image segmentation system. In K.-D. Althoff, R. Bergmann, & L. K. Branting (Eds.) *Case-Based Reasoning Research and Development* (pp. 525-535). Berlin: Springer Verlag.
11. Zamperoni, P. & Starovoitov, V. (1995). How dissimilar are two gray-scale images? In *Proceedings of the Seventeenth DAGM Symposium* (pp.448-455). Berlin: Springer Verlag.
12. Santini, S. & Jain, R. (1999). Similarity measures *IEEE Transactions on Pattern Analysis and Machine Intelligence,. 21*(9), 871-883.
13. Horikowa, Y. (1996). Pattern recognition with invariance to similarity transformations based on third-order correlations. In *Proceedings of Internatinal Confernce on Pattern Recognition´96*, IEEE Computer Society Press (pp 200-204).
14. Leitao, F. (1999). A study of string dissimilarity measures in structural clustering. In: S. Singh (Ed.) *Advances in Pattern Recognition* (pp. 385-394). Berlin: Springer Verlag.
15. Mehrotra, G. (1993). Similar shape retrieval using a structural feature index. *Information Systems*, *18* (5), 525-537.
16. Cortelazzo, C., Deretta, G., Mian, G.A., & Zamperoni, P. (1996). Normalized weighted Levensthein distance and triangle inequality in the context of similarity discrimination of bilevel images. *Pattern Recognition Letters*, *17*(5), 431-437.
17. Crouzil, A., Massipo-Pail, L., & Castan, S. (1996). A new correlation criterion based on gradient fields similarity. In *Proceedings of Internatinal Confernce on Pattern Recognition´96,* IEEE Computer Society Press, (pp. 632-636).
18. Moghadda, Nastar, & Pentland (1996). A Bayesian similarity measure for direct image matching. In *Proceedings of Internatinal Confernce on Pattern Recognition´96,* IEEE Computer Society Press, (pp. 350-358).

19. Moghadda, Jebra, & Pentland (1998). Efficient MAP/ML similarity matching for visual recognition, In *Proceedings of Internatinal Confernce on Pattern Recognition´98,* IEEE Computer Society Press, (pp. 876-881).
20. Wilson, D.L., Baddely, A.J., & Owens R.A. (1997). A new metric for gray-scale image comparison, *International Journal of Computer Vision*, *24*(1), 5-19.
21. Messmer, B., & Bunke, H. (2000). Efficient subgraph isomorphism detection: A decomposition approach. *IEEE Trans. on Knowledge and Data Engineering*, *12*(2), 307-323.
22. van der Heiden, A., & Vossepoel, A. A landmark-based approach of shape dissimilarity. In *Proceedings of the International Conference on Pattern Recognition'99,* IEEE Computer Society Press, (pp. 120-124).
23. Perner, P. (1998) Content-based image indexing and retrieval in a image database for technical domains, In H.H.S. Ip & A. Smuelder (Eds.) *Multimedia Information Analysis and Retrieval* (pp. 207-224). Berlin: Springer Verlag.
24. Voss, A. (Ed.) *Similarity Concepts and Retrieval Methods*. Fabel Report No. 13, GMD-1993, ISSN 0942-413.
25. Perner, P. (1998). Image analysis and classification of HEp-2 cells in fluorescent images, *Fourteenth International Conference on Pattern Recognition* (pp. 1677-1679). Brisbane Australia: IEEE Computer Society Press.
26. Perner, P., Zscherpel, U., & Jacobsen, C. (2000). Comparison between neural networks and decision trees application of machine learning in industrial radiographic testing. *Pattern Recognition Letters, 22/1*, 47-54.
27. Perner, P. (1998). Different learning strategies in a case-based reasoning system for image interpretation. In B.Smyth & P. Cunningham (Eds.) *Advances in Case-Based Reasoning* (pp. 251-261). Berlin: Springer Verlag.
28. Leake, D.B., & Wilson, D.C. (2000). Remembering why to remember: performance-guided case-base maintenance, In E. Blanzieri & L. Portinale (Eds.) *Advances in Case-Based Reasoning* (pp. 161-172). Berlin: Springer Verlag.
29. Portinale, L., Torasso, P., & Tavano, P. (1999). Speed-up, quality, and competence in multi-modal reasoning In K.-D. Althoff, R. Bergmann, & L.K. Branting (Eds.) *Case-Based Reasoning Research and Development* (pp. 303-317). Berlin: Springer.
30. Perner**,** P. (1994), How to use Repertory Grid for Knowledge Acquisition in Image Interpretation. HTWK Leipzig Report Nr. 2.
31. Schmitt, S., Jerusalem, D., & Nett, T. (2000) Representation and execution of questioning strategies for the acquisition of customer requirements, In M.H. Göker (Ed.) *Eighth German Workshop on Case-Based Reasoning*, p.23-37, DaimlerChrysler Research and Technology FT3/KL.
32. Smyth, B., & McKenna, E. (1998). Modeling the competence of case-bases. In B. Smyth & P. Cunningham (Eds.) *Advances in Case-Based Reasoning* (pp. 208-220). Berlin: Springer Verlag.
33. Leake, D.B., & Wilson, D.C. (2000). Remembering why to remember: performance-guided case-base maintenance. In E. Blanzieri & L. Portinale (Eds.) *Advances in Case-Based Reasoning* (pp. 161-172). Berlin: Springer Verlag.
34. Cordella, Goggia, Sansone, Tortorella, & Vento (1999). Graph matching: A fast algorithm and its evaluation. In *Proceedings of the International Conference on Pattern Recognition'99,* IEEE Computer Society Press (pp. 1582-1584).
35. Cheetham, W. & Graf, J. (1997), Case-Based Reasoning in Color Matching, In: Leake, D.B. & Plaza, E. (Eds*.) Case-Based Reasoning Research and Development,* (pp. 1-12).Berlin, Springer Verlag.
36. Bergmann, R. & Wilke, W.(1996), On the Role of Abstraction in Case Based Reasoning, In: Smith, I. & Faltings, B. (Eds.) Advances in Case-Based Reasoning, (pp. 28-43). Berlin, Springer Verlag.