Hybrid knowledge based system for automatic classification of B-scan images from ultrasonic rail inspection

J. Jarmulak[1,2], E.J.H. Kerckhoffs[1], P.P. van't Veen[2]

[1] Delft University of Technology, Faculty of Information Technology and Systems,
Department of Technical Informatics, P.O. Box 356, 2600 AJ Delft, The Netherlands.
[2] TNO - Institute of Applied Physics, P.O. Box 155, 2600 AD Delft, The Netherlands.
jacek@kgs.twi.tudelft.nl

Abstract

Dutch Railways use a special train for the ultrasonic inspection of rails. The output of the ultrasonic scanning system installed on the train consists of echo images – so-called B-scans. The B-scans contain images of rail constructions, noise artifacts, and/or defects. Originally, all the images were interpreted and classified by an operator. Later, a simple rule-based classifier was build which could classify some of the images automatically. Recently, a new version of the train has been built capable of faster and more detailed rail inspection. This necessitated improvement of the automatic classification software. A prototype system has been developed which uses both a rule-based expert system and case-based reasoning (CBR) for the image classification. A hybrid architecture has been chosen because it satisfies the requirements better than systems based on one technique only. The paper describes the overall system design and presents the results of tests on real data. The future work necessary for the deployment of the system on the inspection train is outlined.

Introduction

To guarantee safe operation of the railway net periodical inspection has to be carried out in order to detect damage before it leads to disruption of the normal operation. One type of damage that can occur are cracks usually found in the head of the rail, at the bolt holes, or at the welded joints. These cracks can be detected using ultrasonic inspection techniques. Special transducers are used to send ultrasonic pulses into the rails and the returning echoes are recorded. The echoes can come from the bottom of the rail, the bolt holes (or other construction elements), or the cracks.

Dutch Railways use a special train to perform ultrasonic inspection of the rail net in the Netherlands as well as abroad (Roos 1990). Inspection (data acquisition) is done at speeds up to 50km/h (75km/h using the new train) by a set of ultrasonic transducers (each placed at a different angle) gliding over the rails on a water film. On a clean rail only an echo from the bottom of the rail is present and it is ignored. On sections where other echoes are present they are assembled to form an image (so-called B-scan), see e.g. Figure 1. Any cracks present in the rails will be visible in these images. However, most of the images show the constructions present in the rails and/or noise (see Figure 2). There is no simple method of defect detection available and all images have to be analyzed to determine if they contain any echoes from defects or not.

When the ultrasonic rail-inspection system was first introduced the B-scan image analysis was performed entirely by an operator. Later, automatic classification was added to the system, first using SQL queries on Oracle database, and later using a rule-based expert system (Clips (Giarratano and Riley 1989)). With the help of the automatic classification it was possible to deliver a full inspection report at the end of each work day.

Recently, a new inspection train has been build with more transducers per rail (8 instead of 4) and a higher inspection speed. To keep up with the larger amount of data a better automatic classification system (achieving higher recognition ratio) had to be developed.

This paper describes the results of the research in better methods of automatic B-scan image interpretation. First, a description of the problem is given and the choice for the use of a hybrid rule-based/CBR system is motivated. Then, the design of the prototype system is described. The results of the tests on real data are presented. The chosen way of deployment of the system on the inspection train is discussed before the conclusions are presented.

Problem Description

The problem that has to be solved concerns classification of images, in which pixels correspond to echoes from various channels, into two main classes: defects or non-defects.

Types of Images

There are various types of images showing rail constructions, noise, defects, or combinations of these. The most common image type is the loss of the bottom echo constituting approx. 60% of all images. These are trivial to recognize. The research

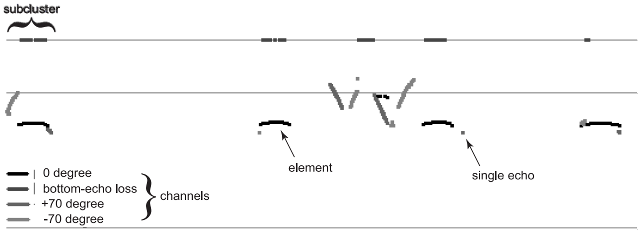

Figure 1. Ultrasonic B-scan of a fish-plated joint. (Vertical placement of the bottom-echo-loss indications is arbitrary.)

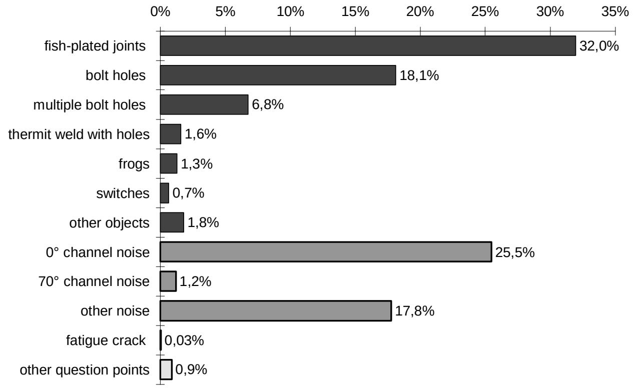

Figure 2. Objects present in the rails (percentages add up to more than 100% because some categories may overlap).

concentrated on the recognition of the remaining images and whenever recognition percentages are presented in this paper they do not include the bottom-echo-loss images.

The classes recognized among the non-bottom-echo-loss images are listed in Figure 2. As one can see the defects are rare. The so-called question points (images with possibly defects) constitute only about 1% of the non-bottom-echo-loss images. The low percentage of defects is common in NDT (non-destructive testing) problems.

Classification Goals

As already stated, the goal of the system is to automatically classify the B-scan images. First of all, we want to distinguish between defect and non-defect images. Additionally, in the process of recognizing non-defect images, the image type (e.g. bolt hole) is determined. The possible combinations of real and automatic classifications are shown in Table 1.

The two main parameters which describe the performance of the system are: the percentage of the images automatically classified (not shown to the operator) and the percentage of the defects classified as non-defects. Obviously, we want the first percentage to be as high as possible and the second one as low as possible. For an inspection company it is especially important that the percentage of missed defects is as very low. To keep it low one is ready to sacrifice the overall automatic recognition ratio.

A system destined for field use should be able to do the classification fast enough. In practice it means that it should be at least as fast as the human inspector. As far as the speed is concerned, one can distinguish between average processing speed (e.g. calculated over the hole data set) and a worst-case speed (e.g. for a very complex image) – important when the system has to work interactively with an operator.

Methodology Choice

There exist only a few rail inspection systems similar to the one owned by Dutch Railways. The articles published about these systems contain few details about the techniques used. For example system described in (Havira & Chen 95) uses syntactic pattern recognition for classification, but the article does not describe the recognition algorithm at all. Lesiak (92) also gives no details of the classification algorithms, but the low resolution of the system (128x24 cells) suggests that a technique like template matching could be used.

Table 1. Possible combinations of real and found classifications.

real	**classified as**	**shown to operator**	**remarks**
non-defect	non-defect	no	desired behavior
non-defect	defect	yes	acceptable, but reduces confidence in the system
non-defect	unknown	yes	acceptable if not too often
defect	defect	yes	desired behavior
defect	non-defect	no	**should not occur**
defect	unknown	yes	almost the desired behavior

A system for interpretation of ultrasonic B-scans, however not of rails but of welds, is described in (Hopgood et al. 93). It is build around a blackboard rule-based system and uses neural networks for local defect classification. Though this problem is in many aspects similar to ours the differences in the images and the difficulties in knowledge formalization (described further) did not let us use the same techniques.

A common method of data classification used in NDT problems is the use of statistical classifiers (or neural networks), e.g. (Chien-Ping et al. 93). This usually requires being able to describe the data with a fixed length feature vector and the availability of a training set.

In our problem the images are combinations of constructions, acquisition system disturbances, noise, and defects. As a result the images are very inhomogenous and impossible to describe with a fixed length feature vector. The number of all possible image types is very high thus obtaining an exhaustive set of data examples is practically impossible. Related to this is the fact that there is always a chance of occurrence of unexpected data (significantly different from the training set) and most statistical classifiers are not good at handling such data. Also, obtaining a good classifier is difficult when one class is only a small fraction of the possible classes (as are the defects). Moreover, statistical classifiers are difficult to maintain by the end users. Making changes to the system requires good insight into the working of the classifier otherwise the reliability may be compromised. Altogether, statistical classifiers were found unsuitable for the task.

As there was already a working rule-based classification system, one way of improving the classification ratio would be to add extra rules to the rule base. After initial study, it became clear that describing the images using rules was not feasible. Apart from the images already described by rules, the knowledge about the images was not well formalized, the record kept from past inspections was incomplete because not all images were fully classified, and the knowledge acquired from the operator in a few acquisition sessions was either limited to very typical (ideal) images or anecdotal of character. There was a large variety of images, thus apart from a few simple image types, any particular image description would apply to only a small fraction of all possible images. Additionally, because of the workload at the inspection company, it was difficult to arrange knowledge acquisition sessions.

Even if a rule-based system could be successfully constructed the problem of rule maintenance would still be un-

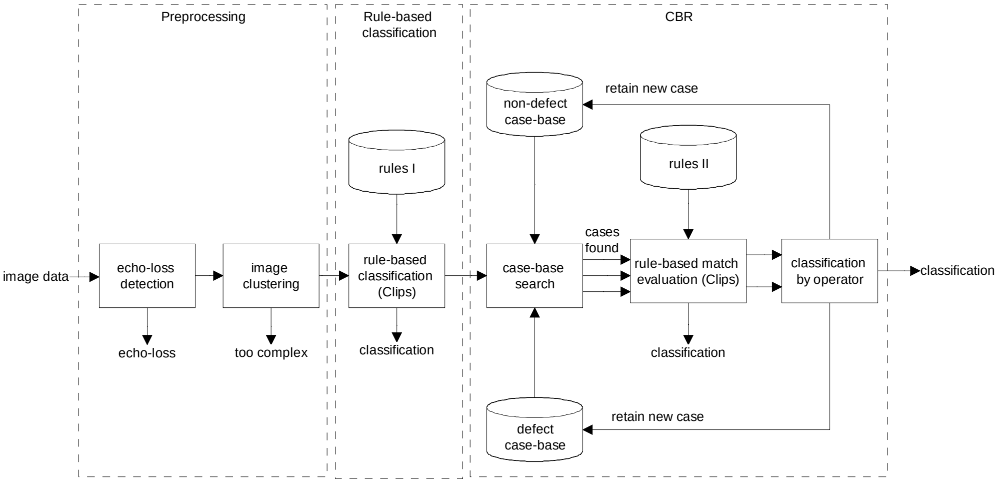

Figure 3. Schema of the prototype hybrid classification system.

solved. Already in the old system there was an example of reduced recognition ratio after changes were made to the acquisition system and the rules not updated accordingly. Besides, during the project the data from the new 8-channel system was not available, which meant that any rule-base designed using data from the 4-channel system would require serious adaptation to the new data.

Because the knowledge acquisition and maintenance were such a big problem, our attention turned to learning systems. Because statistical classifiers and neural networks were already dismissed, and the various types of decision trees suffered form similar problems as these two, a choice was made in favor of case-based reasoning, which let us develop an adaptable, reliable, and easy to maintain system.

However, because the existing rule-base performed quite well we decided to keep it. If one knows how to describe the images then they can be recognized quicker by the rule-base. Besides, large noise images are very difficult for the case-based reasoner because per definition they are not repeatable.

The Prototype

The overall design of the prototype is shown Figure 3. The system consists of three main stages: preprocessing, rule-based classification, and case-based classification.

Preprocessing

In the preprocessing stage the B-scan image consisting of pixels corresponding to echoes from various channels is segmented into meaningful elements like lines, parabolas and regions of noise. The segmentation algorithm begins with constructing a minimum-spanning tree through all the echo points in a channel. The tree is then split on the longer edges and lines and parabolas are fitted through the branches. More details of the segmenting algorithm are described in (Jarmulak 1996).

Apart from segmenting into elements each image can be divided into subclusters. The subclustering thresholds have been chosen so that, e.g. multiple-bolt-hole objects are divided into single bolt holes (see Figure 1).

Rule-Based Classification

Information about image elements is represented as fact strings and sent to the Clips shell. Twenty six rules process the facts in two stages: first the image subclusters are classified, and then the subcluster classifications are combined into the final classification.

After processing by the inference engine the resulting classification facts (both for the subclusters and the whole image) are retrieved from Clips. If the image was recognized the processing stops here.

Case-Based Reasoning

The case-based classifier maintains two case-bases: one with defect images and one with non-defect images. For each new image a search is done in both case-bases for similar images. The image to be classified and the retrieved images together with the match results are processed by a set of rules which either arrive at the final classification or decide that the image has to be classified by the operator. The images classified by the operator are stored in the appropriate case-base for future use.

Case. A case contains first of all the information about the image elements and the correct image classification. For each element its type, location, and equation of the fitted parabola or line are stored. Each case also contains pointers to the whole image data (all echoes) stored in a separate file. The subcluster classifications assigned by the rule-based classifier are also stored. For the purpose of case-base maintenance the date of entry, the number of all matches and the number of better matches are stored with each case.

Case matching. Matching is done in stages corresponding to the image hierarchy: whole image, subclusters, channels, elements.

For each corresponding pair of image elements their location, size, and orientation are compared. For each parameter an absolute and/or relative difference is calculated and then mapped via a fuzzy membership function into the [0..1] range. Because a channel in a subcluster may contain more than one element and the number of elements in the channels being matched may differ, the channels are matched using inexact graph matching. Within the subclusters the channel locations and within the whole image the subcluster locations are compared. Subresults are combined using a *min* operator. The matching procedure is broken off when it becomes clear that the current match would be worse than the best match so far.

Case-base. In the prototype the case-base has a two level organization. The top level has a simple hierarchical organization with divisions based on the number of subclusters, channels present and the length of an image. The leaves of the hierarchy are collections of clusters of similar cases. For each cluster of cases a representative case is found which has the smallest distance to the remaining cases in the cluster. When a new case is added to the case-base it is inserted into the cluster which contains the best matching case. A cluster has a certain size limit and if after adding a new case the limit is exceeded then the cluster is split into two new clusters.

Case retrieval. Looking for the best match in the case-base proceeds fastest if a good match is found at the very beginning of the search, this way matching with many remaining less similar cases does not have to be carried out in full. To increase the probability of finding a good match at the beginning of the search the following is done:

– the top level of the case-base guides the search to a collection of clusters of cases with characteristics similar to the current case,

– a collection of clusters of cases is sorted on the good match ratio for the cluster representatives, so that clusters usually giving high matches are searched first,

– within a collection the matching is first done with the cluster representatives and the further search is done beginning with the clusters with higher representative matches.

The system can retrieve several best matching cases – in the current system three.

Match evaluation. For each of the retrieved cases the match evaluation is done by Clips using a set of 25 rules. The simpler rules look at the values of the overall matches with defect and non-defect cases. The more complex rules look at the differences between the images and decide if for a given image type the differences are significant or not.

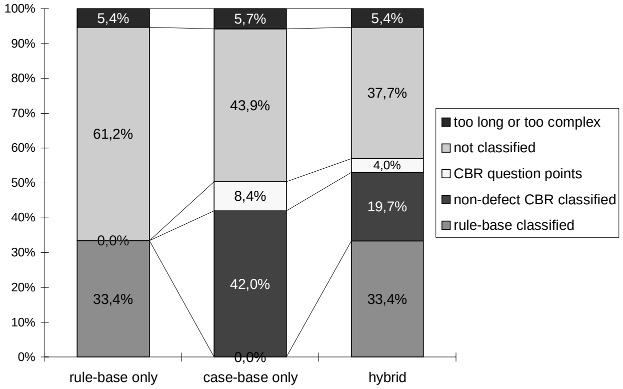

Figure 4. Comparison of classification results for rule-based, case-based, and hybrid classifiers.

Implementation

The software has been written entirely using Visual C++. As an expert system shell Clips from NASA was used. A possibility of using an existing CBR shell has been considered but they seem not well suited to processing cases consisting of images (Watson and Marir 1994).

The Clips code has been encapsulated in a separate program – a Clips server. The main application communicates with it by sending and receiving messages via the socket interface. Because rule processing is done twice by two different sets of rules there are two Clips servers running, each listening to a different port number.

Tests, Results, and Observations

Because the processing speed is an important factor in evaluating the system it is important to note here that all the tests described in this section have been done on a single CPU 300MHz Pentium II computer with 64MB RAM. Because the data files correspond to a certain length of track inspected the system performance is expressed in kilometers of track-data processed per hour (km/h).

Test Data Set

All the data ever acquired with the ultrasonic rail-inspection system is available on optical disks. It would seem that it would be easy to obtain a training set, however, it turns out that only the computer classified images have the detailed classification (e.g. 0° noise, bolt hole), all the remaining images were given classification only when a possible defect was present. This means that the available data had practically only defect/non-defect classification. Several test runs were done on this data, however, the results did not provide the required detail on information about what type of images are recognized and what not.

For the purpose of the test a set of data files had to be fully classified. After a period of working on the project we had enough experience to be able to do classification without the help of the inspector. It took more than a week to fully classify a set of 128 normal data files and 12 files containing defect images (extracted from a large number of normal data files). The test data sequence consisted of 6 randomly ordered defect files (to fill the defect case-base), 128 randomly ordered normal data files, and again 6 defect files (to test the system reliability). Altogether, the test set contained approx. 38000 non-bottom-echo-loss images from approx. 1000 km of track.

Rule-Based vs. Case-Based vs. Hybrid Classifier.

Three test runs were done to compare the performance of the rule-based, the case-based, and the hybrid classifiers. The overall classification results are shown in Figure 4. The rule based classifier has the lowest recognition ratio, however, it was approx. 14 times faster than the other two classifiers (the avg. processing speed was 401 km/h, 23 km/h and 28 km/h respectively). Important things that can be noticed are:

– rule-base is capable of recognizing some complex noise images which are unrecognized by the case-based classifier (5.7% vs. 5.4%),

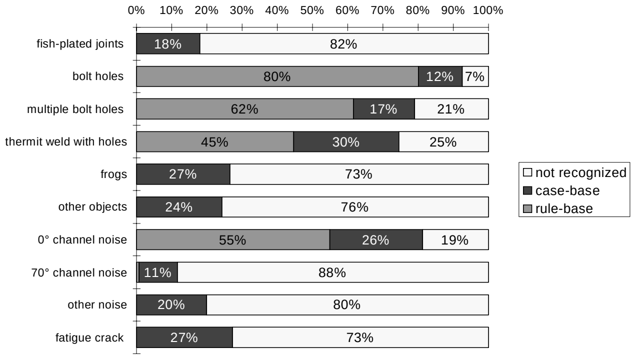

Figure 5. Average recognition ratios for various image types for the hybrid system.

- CBR question points are these images which had a close match to a defect case – rule base recognizes some of them correctly as noise, while the case-based classifier shows them to the operator.
- in the hybrid system the case-based classifier uses some information from the rule-based classification during matching and match evaluation – this information is not available in a case-based only system.

It is clear that the results of the hybrid classifier, as far as the recognition ratio is concerned, are the best. It has certainly advantages over the case-based-only classifier especially as the rule-based part is already being used and can easily be incorporated into the new system. In spite of a much lower processing speed than the rule-based classifier, the adaptability and maintainability of the CBR system is considered as a big enough advantage to outweigh the longer processing time.

Results for the Hybrid System

The recognition percentages for the various categories of images are shown in Figure 5. Only 27% recognized fatigue cracks does not mean that some of these defects were missed by the system – the remaining 73% were classified as belonging to the question point class and shown to the oper ator.

For a CBR system one can expect an increasing recognition ratio with an increasing size of a case base. This trend is visible especially at the beginning of the scan where the CBR system quickly (after approx. 30 data files) learns to recognize simpler images like bolt holes and 0° noise. It takes much longer to improve recognition ratio of the more complex images like fish-plated joints (see Figure 6).

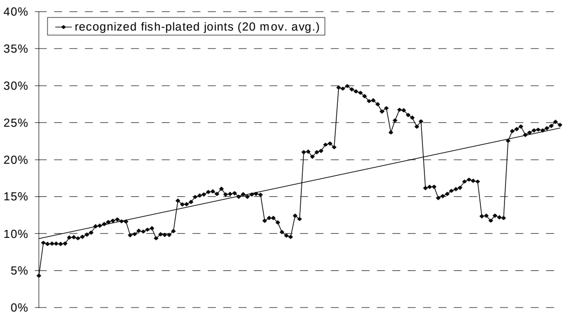

Figure 6. Increasing fish-plated-joint recognition ratio with an increasing size of the case-base (20-file moving average).

At the end of the scan the non-defect case-base contained approx. 13000 cases (15.5MB) and the defect case-base 1700 cases (1.7MB).

The system has turned out to be very reliable. The discrepancies between the classifications done by an expert and by the system were analyzed and it has turned out that, except for two cases, they concerned ambiguities in possible interpretation or inconsistent use of some classifications in the preclassified files.

Future Work

The results of the tests have been found satisfactory enough to continue the work and make the system suitable for deployment on the train. The work goes mainly towards improving the recognition ratio and the speed. Because larger and more case-bases will be used, methods for case-base maintenance will also have to be developed.

Improving Recognition Ratio

Obviously the recognition ratio will improve with the increased number of cases in the case-base. However, the larger the case-base the slower the whole system becomes.

Another way to increase the recognition ratio is to optimize the image segmentation and matching algorithms. An optimal segmentation algorithm should return as small number of elements as possible without compromising the ability to detect defects.

The matching algorithms can be further optimized, mainly the membership functions used. Because of the amount and the variety of data the most useful method is by iteratively changing the parameters and testing the system on a large data set. Also, making use of the subcluster classifications (from the rule-based classifiers) during the case matching can improve the recognition ratio.

Extra rules can be added in the match evaluation rule-base to ignore certain types of noise or to make better use of the subcluster classifications.

Increasing Processing Speed

Increasing the speed of the system is a crucial factor for its success. Currently, the system reaches an average processing speed of approx. 25 km/h with a case-base of approx. 13000 cases. The goal is to reach a processing speed of approx. 75 km/h before the system will be deployed on the train.

Increase in the speed can be achieved by improving the algorithms, optimizing the code, and the use of faster hardware. For example, the top level case-base organization will be optimized using decision tree building algorithms.

The system will probably be implemented on a 2 CPU Pentium II hardware. The case-base search algorithm will be implemented so that it can search several clusters of cases in parallel (in multiple program threads), this way an optimal use of the two CPUs will be made.

Maintenance

Currently, the only case-base maintenance possibility is the ability to prune the cases with only bad matches. For the use in practice two types of tools will have to be made. First one will be used for keeping track of available case-bases. As the inspections are done on various types of rail track in various countries, therefore, it makes sense to have a set of case-bases from which the most appropriate can be used for a given type of track. Because the current system already uses an Oracle database for administration purposes, the case-base administration will probably be added to the existing administration system.

Another type of tool is needed for investigating and manipulating individual case-bases. One needs to be able to navigate the case-base hierarchy and see various statistics for the underlying branches. This tool would enable browsing the individual cases, e.g. allowing for selective case deletion.

Phased Deployment

Figure 7a shows schematically the current classification system used on the train. The classification is done in two stages. First the data is processed by the rule-based classifier. It writes the classifications to a classification file, marking every not recognized image as still-to-be-classified. The classification file is later read by the interactive classification program which allows the operator to step through the not classified images and do the final classification.

Figure 7b shows the outline of the new system as it will first be implemented on the train. The existing rule-based classifier will be used unchanged. As the new system will require entry of all the classifications, which was only optional in the old system, the interactive classification software will be adapted to simplify the entry of image classifications by providing good default values based on the best case-base matches.

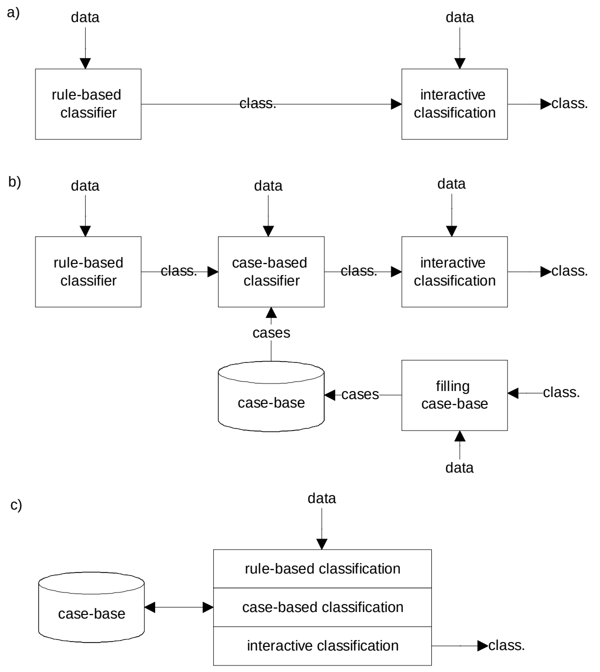

Figure 7. Phased deployment: a) current system, b) system extended with CBR capabilities (batch mode), c) goal system. (class. = classification file)

The CBR system will work in batch mode, which means that it will not reach it full potential because it will not be able to learn directly from the classifications made by the user. Only afterwards, the case-base will be filled with batches of classified images.

This system will enable us to gather experience with the use of the CBR system in real-life conditions. If the results are satisfactory an integrated system will be developed which combines the rule-based, the case-based, and the interactive classifiers, see Figure 7c.

Conclusions

The project has shown that the use of a combination of a CBR and a rule-based system for the classification of complex images has many advantages. The CBR component is capable of adapting to a large variety of data by learning from the classifications made by the operator. The rule-based component makes it possible to use the a-priori knowledge about the images. Rules can also recognize large noise images which are difficult for a CBR system. Subclassifications made by the rule-based classifier can also improve the speed and recognition ratio of the CBR system. The rule-base size is kept small which should make the system maintenance easy.

The tested prototype has achieved satisfactory recognition ratio combined with very high reliability. The main disadvantage of the system is the low processing speed. However, the latest high-end PC hardware is fast enough to enable development of a system that can be used in practice. The system deployment will be done in phases so that minimum risk is taken and maximum experience can be gathered.

We think that presented approach can be useful for other problems where complex inhomogeneous data has to be classified and where the domain knowledge is difficult to formalize.

References

Chien-Ping, Ch., Shmerr, L.W., and Thompson, R.B. 1993. Ultrasonic flaw detection using neural network models and statistical analysis: Simulation studies. In *Review of Progress in QNDE*, Vol. 12, pp. 789-795.

Giarratano, R. and Riley, G. 1989. *Expert systems: Principles and Programming*, PWS-KENT Publishing Company.

Havira, R.M. and Chen J. 1995. “High Speed Rail Flaw Pattern Recognition and Classification”, *Proc. SPIE* Vol. 2458, pp.64-73.

Hopgood, F.F., N. Woodcock, N.J. Hallam, and P.D. Picton, 1993. “Interpreting ultrasonic images using rules, algorithms and neural networks”, *European Journal of NDT*, Vol. 2:4, April 1993, pp. 135-149.

Jarmulak, J. 1996. B-scan Image Clustering and Interpretation in Ultrasonic Rail-Inspection System. In *Proceedings of the second annual conference of the Advanced School for Computing and Imaging, Lommel, Belgium, June 5-7*, 1996, pp. 190-195.

Lesiak, P. 1992. “System for Automatic Ultrasonic Quality Control of Railroad Rails”, *Russian Journal Of Nondestructive Testing*, Vol. 28:7, pp.383-388.

Roos, R. 1990. Het ultrasoon railinspectie systeem, *NAG Journal*, Nr. 105, november, pp. 21-34.

Watson, I. and Marir, F. 1994, Case-based reasoning: A review, *The Knowledge Engineering Review*, Vol. 9:4, pp. 327-354.