Content-based retrieval of surface defect images with MPEG-7 descriptors

Jussi Pakkanen and Jukka Iivarinen

Helsinki University of Technology, Laboratory of Computer and Information Science
P.O. Box 5400, FIN-02015 HUT, Finland

ABSTRACT

In this paper a prototype system is described for the management and content-based retrieval of defect images in huge image databases. This is a real problem in surface inspection applications, since modern inspection systems may produce up to thousands of defect images in a day. We are using a noncommercial, generic content-based image retrieval (CBIR) system called PicSOM that is modified to fit to the special requirements of our application. The system is tested with a small pre-classified database of surface defect images using the MPEG-7 features. The scalability of the system is also examined using a larger database. Results indicate that the system works with a high level of success.

Keywords: self-organizing map, content-based image retrieval, surface inspection, surface defects, MPEG-7 descriptors

1. INTRODUCTION

The need for efficient and fast methods for content-based image retrieval (CBIR) has increased rapidly during the last decade in many fields of industry and research. For example, modern surface inspection systems produce huge amounts of image data that has to be stored, managed, browsed, searched, and retrieved efficiently. The CBIR systems seem to be ideal tools for management of these huge image databases. The basic idea in CBIR is to find similar images without first labeling or annotating the images. Instead some relevant features are extracted from the images, e.g. color histograms, texture features and shape features. These features are then stored in a database and indexed to speed up the retrieval process. What this means is that the search criterion is the image content, not some prior information that has been extracted by a human.

Surface inspection of web materials is a challenging problem that consists of several subproblems, e.g. image acquisition, defect detection, and defect classification. In this paper we are dealing with management and content-based retrieval of surface defect images that are obtained from a real, online paper inspection system. We have designed and realized a prototype CBIR system that is based on a noncommercial, generic content-based image retrieval system called PicSOM. The visual descriptors of the MPEG-7 standard are used as image features. The test database consists of images of preclassified surface defects. Some earlier work on the CBIR and classification of these kinds of images can be found e.g. in references.

The surface defects we use in this paper are paper defect images taken from a running paper web. These images are extremely hard to classify due to their several unique characteristics. When running at full speed, a paper machine produces paper at speeds of over 30 m/s. The kind of cameras capable of taking sharp pictures of web defects have only existed for a couple of years. This means that there is no “decades old” knowledge pool we could turn to for additional guidance. Another thing is that even an expert cannot sometimes classify a defect just by looking at the image. He would need to do laboratory experiments on the the actual defect area to get an unambiguous classification. These features make the classification problem even more difficult.

Further author information: E-mail: jussi.pakkanen@hut.fi, jukka.iivarinen@hut.fi
2. CONTENT-BASED IMAGE RETRIEVAL

The most common way to do content-based retrieval of images is to calculate various features of the images. These may be simple, such as color histograms and distributions, or very complex, such as hierarchical descriptions of the items in the image. The similarity search can then be done using these feature vectors. The calculated features form a feature space. To proceed, we also need a way to measure distances between these feature vectors, in other words we need a metric for the data space. Usually the features consist of \( n \) real numbers, and the distances are measured with the Euclidian distance.

When manipulating huge databases, a good index is a necessity. Processing every single item in a database when doing queries is extremely inefficient and slow. Raw image data is non-indexable as such, so the feature vectors must be used as the basis of the index. The problem we now face is that indexing data points in a multidimensional vector space is a non-trivial task. We propose to solve this image indexing problem by using the self-organizing map (SOM)\(^{10}\) as an index to the images’ feature space data. The SOM is trained to match the shape of the data in the feature space. After the training, the closest node in the SOM is calculated for every image in the database. This information about the closest nodes is stored. When a query is done, the first thing to be done is to calculate the closest SOM node, also know as the best matching unit (BMU), to the query image’s feature vector. When this is done, we know which images in the database are the closest to the query image: the ones that map to the same node as the query image. This cuts down processing time significantly. An added bonus is the hierarchy-preserving structure of the SOM. If we want to find more similar images, we can just examine the neighboring nodes of the BMU.

2.1. PicSOM Retrieval System

PicSOM has been developed in Laboratory of Computer and Information Science at Helsinki University of Technology to be a generic content-based image retrieval (CBIR) system for large, unannotated databases.\(^3,4\) It builds on the concepts of unsupervised clustering, self-organizing maps, and relevance feedback.

**PicSOM Training**  PicSOM works by first calculating a chosen number of feature sets for each image in the database. Several types of features can be used e.g. color, shape, texture, and structural features. Then a tree-structured self-organizing map (TS-SOM)\(^{11}\) is trained with each feature set. TS-SOM is a basic building block of the PicSOM system. It is a tree-structured vector-quantizer that has self-organizing maps (SOMs) at each of its hierarchical levels. The SOM sizes grow from top to bottom in a hierarchical way (see Figure 1). This kind of tree-structure makes training and searching much faster when compared to a normal, nonhierarchical SOM of the same size — the amount of computations is reduced from \( O(n) \) to approximately \( O(\log(n)) \). After training each image is associated to the closest map unit in each TS-SOM. Then the database can be searched.

![Figure 1](image.png)

*Figure 1*. A diagram of the TS-SOM. The dashed lines show the links between parent and children nodes.
Relevance Feedback Every CBIR system needs some way of determining what a user wants to find. PicSOM uses the idea of relevance feedback to adapt to the user’s search criteria. First the user is shown a group of different kinds of images. The user then selects the ones that look like the image he is searching for. The system then looks what SOM nodes the images map to. Then it tries to find images that map to the same or neighboring nodes as the selected images and as far as possible from the discarded images. The best images are then shown to a user, who again selects the best ones and discards the rest. When this process is repeated the selected images should form clusters in the SOM grid. These can be seen in Figure 2 where the bottom levels of TS-SOMs are plotted. These so-called response maps show how important the particular TS-SOM is in the query. If the positive weights (red areas, marked with arrows) are spread then the map’s effect on the query is small, and if they are well clustered then the map’s effect is high. This way the system learns within few query iterations which maps (or feature sets) are important in each query.

3. EXPERIMENTS

3.1. Feature Extraction

In this research we have used the features of the MPEG-7 standard. The MPEG-7 standard ISO/IEC 15938, formally named “Multimedia Content Description Interface”, defines a comprehensive, standardized set of audiovisual description tools for still images as well as movies. The aim of the standard is to facilitate quality access to content, which implies efficient storage, identification, filtering, searching and retrieval of media. This standard defines several general-purpose features for still images as well as movies. We have used the following still image features.
Color layout specifies a spatial distribution of colors. The image is divided into \(8 \times 8\) blocks and the dominant colors are solved for each block in the YCbCr color system. Discrete Cosine Transform is applied to the dominant colors in each channel and the DCT coefficients are used as a descriptor.

Color structure slides a structuring element over the image, the numbers of positions where the element contains each particular color is recorded and used as a descriptor.

Scalable color is a 256-bin color histogram in HSV color space, which is encoded by a Haar transform.

Edge histogram calculates the amount of vertical, horizontal, 45 degree, 135 degree and non-directional edges in 16 sub-images of the picture.

Homogeneous Texture descriptor filters the image with a bank of orientation and scale tuned filters that are modeled using Gabor functions. The first and second moments of the energy in the frequency domain in the corresponding sub-bands are then used as the components of the texture descriptor.

For a more thorough explanation of the features, see the references cited above.

The reason we have chosen to use these descriptors is that they are standardized. We wanted to test how the actual retrieval system works. Therefore it is better to use these features than it is to develop our own, very problem-specific features. Using MPEG-7 features also makes it easier to compare our work with other CBIR systems.

3.2. Defect Database

The defect database has approximately 1300 surface defect images that were obtained from a real, online paper web inspection system. There are several different kinds of defects, such as light spots, dark streaks, wrinkles, holes, oil stains, and so on. The database had been pre-classified into 14 different classes. All the images were in gray-scale with 256 colors. The size of the images varied considerably. Some were only 200 by 200 pixels in size while others were several thousand pixels high. Examples of the images can be seen in Figure 3.

3.3. The test procedures

PicSOM has a built-in method of evaluating the retrieval results. You tell it which defect class you want to test and which features to use. It then seeks a single example image belonging to that class to start the query. Then it seeks the best matches to the query and selects all returned images that belong to the desired class. All these selected images are then used as a search criterion for another search. Again the found correct images are added to the query criterion and a new query is performed. This process continues until a pre-specified number of iterations is reached.

The results are data vectors, containing the recall and precision values for each iteration. Recall means the percentage of correct images retrieved so far. When recall reaches 1, all desired images have been found. Precision tells how many of all returned images belong to the correct class. A minimum requirement of any CBIR system is that precision should be higher than the a priori probability of the queried class. Otherwise the system is worse than just selecting images in the database at random.

The tests were run separately for every class. The test results were then loaded into Matlab for analysis. We ran tests both using a single feature and a combination of several features in the query.

3.4. Results

The results we obtained are very good. Most of the classes are found with high precision values using relatively few queries. There are, however, two classes that are very difficult. These classes are very similar to each other and they also contain fewer images (30 and 70 images) than the other classes (100 images per class).

Figure 4 gives an example of how the system works. It has two precision/recall figures. Each one contains two graphs, one for an easy class and one for a difficult class. The graph on the top left uses only one feature, color layout, whereas the one on the top right uses all of the features and the bottom one uses a combination of the three best descriptors, color structure, homogeneous texture, and edge histogram. One can immediately see
that PicSOM can combine different features effectively. Precision is clearly higher in the graphs that combine several features.

Another result can be seen in Figure 5. It shows the amount of found images as the number of iterations increases with two different feature sets. The curves rise very steeply at the beginning, meaning that the query results are very good. The system only starts to slow down after 80%–90% of the desired images have been found. It should be noted that PicSOM returns 20 images per round. Therefore the minimum number of rounds required to obtain all desired images is five.

The general performance of the system is quite good. Even when using just one feature, the obtained results are clearly above the a priori probability. The easy class has especially good average precision. The difficult one has clearly worse results, but they are still quite acceptable. In the easy class, all desired images are obtained with just seven iterations. This is very close to the optimal value of five. The difficult class has again noticeably worse performance. On average 80% of the images are found with just 10 queries.

All of the graphs have one thing in common. Their precision increases noticeably after the first couple of iterations. This is a very desired feature of the PicSOM system. This increase indicates that PicSOM is able to learn what kind of images the user was searching. It should be noted that this learning is done based only on the feedback information gathered during the queries.

Another thing we examined was the optimal feature selection, meaning how different combinations of features work. We found that using only the three best features, color structure, edge histogram and homogeneous texture, yielded almost as good results as using all of the features. This can be seen in Figure 5. The graphs on the right is not as steep and go slightly slower than the ones on the left. Figure 4 shows how especially the difficult class is harder to find with only the three best features, as the precision is noticeably lower.

3.5. Generalization to larger databases

Content-based image retrieval is usually applied to very large image databases that have thousands or tens of thousands of images. To test the suitability and scalability of our system to this task, we have tested it with
another database that has 13 000 images. This testing has its own set of problems.

As was noted earlier, pre-classifying the image data is a very hard task that requires an expert. The amount of time and money required to pre-classify an image database of this size is beyond our reach. Therefore these tests are run on a non-classified database. This means that the results are qualitative and fuzzy rather than exact, measurable and repeatable numbers.

The system scales very well. The queries are completed almost in real time, even though the database size is ten-fold. The query results also seem very sensible when examined by visual inspection. The query results are very similar to the desired images and the system adapts if the query target is modified during the search. These results suggest that the system retains a high level of success when used with large databases. However, it should be noted that we can not prove this conclusively, due to the lack of a proper, classified database.

4. CONCLUSIONS

In this paper we have proposed and evaluated a content-based image retrieval system. The task of this system is to find similar images in a large, unannotated image database. For each image a set of MPEG-7 descriptors is calculated and a TS-SOM is trained for each feature. Using these and relevance feedback obtained from the user, images of the desired type can be found quickly and efficiently.

The PicSOM system is very efficient in combining different features. This can be easily seen in Figure 4. When using all features, the precision levels are clearly higher in both classes. The graphs go significantly higher.
Figure 5. Recall values as a function of the number of query iterations. The left one uses all features, the right one has the three best ones.

all the time. The decrease in precision with increasing number of queries is not as strong when using several features.

Another result of our research is showing that MPEG-7 descriptors can be used as descriptors in machine vision problems. Their performance on these rather difficult paper defect images is surprisingly good. Of course, these tests do not guarantee the usability of MPEG-7 descriptors in the general case. It does, however, imply that MPEG-7 descriptors are worth experimenting with.

Acknowledgments

The authors wish to thank the PicSOM group (J. Laaksonen, M. Koskela, S. Laakso, E. Oja) at Helsinki University of Technology and our industrial partner ABB Oy (J. Rauhamaa). The financial support of the Technology Development Centre of Finland (TEKES’s grant 40397/01) and the Academy of Finland (project New Information Processing Principles, 44886) is gratefully acknowledged.

REFERENCES