Chapter 7

Image analysis applications

Erkki Oja, Jorma Laaksonen, Jukka Iivarinen, Markus Peura, Markus Koskela, Sami Laakso, Jussi Pakkanen, Ville Viitaniemi

7.1 Content-based image retrieval by Self-Organizing Maps

Erkki Oja, Jorma Laaksonen, Markus Koskela, Sami Laakso, Ville Viitaniemi

Content-based image retrieval (CBIR) has been a subject of very intensive research effort for more than a decade. It differs from many of its neighboring research disciplines in computer vision due to one notable fact: human subjectivity cannot totally be isolated from the use and evaluation of CBIR systems. In addition, two more points make CBIR systems special. Opposed to such computer vision applications as production quality control systems, operational CBIR systems would be very intimately connected to the people using them. Also, effective CBIR systems call for means of interchanging information concerning images' content between local and remote databases, a characteristic very seldom present, e.g., in industrial computer vision.

PicSOM

We have developed a neural-network-based CBIR system named PicSOM [1, 2]. The name stems from "picture" and the Self-Organizing Map. The SOM is used for unsupervised and topology-preserving mapping from the image descriptor space to a two-dimensional lattice, or grid, of artificial neural units. The PicSOM system is built upon two fundamental principles of CBIR: query by pictorial example and relevance feedback [3].

The methodological novelty of PicSOM is to use several Self-Organizing Maps in parallel for retrieving relevant images from a database. These parallel SOMs have been trained with separate data sets obtained from the image data with different feature extraction techniques. The different SOMs and their underlying feature extraction schemes impose different similarity functions on the images. Every image query is unique and each user of a CBIR system has her own transient view of image similarity and relevance. Therefore, a system structure capable of holding many simultaneous similarity representations can adapt to different kinds of retrieval tasks. In the PicSOM approach, the system is able to discover those of the parallel Self-Organizing Maps that provide the most valuable information for each individual query instance.

Instead of the standard SOM version, PicSOM uses a special form of the algorithm, the Tree Structured Self-Organizing Map (TS-SOM) [4]. The hierarchical TS-SOM structure is useful for large SOMs in the training phase. In the standard SOM, each model vector has to be compared with the input vector in finding the best-matching unit (BMU). This makes the time complexity of the search $O(n)$, where n is the number of SOM units. With the TS-SOM one can, however, follow the hierarchical structure and reduce the complexity of the search to $O(\log n)$. This reduction can be achieved by first training a smaller SOM and then creating a larger one below it so that the search for the BMU on the larger map is always restricted to a fixed centered below the already-found BMU on the above map.

After training each TS-SOM hierarchical level, that level is fixed and each neural unit on it is given a visual label from the database image nearest to it. This is illustrated in Figure 7.1, where MPEG-7 *Edge Histogram* descriptor has been used as the feature. It can be seen that, e.g., there are many ships in the top-left corner of the map surface, standing people and dolls beside the ships, and buildings in the bottom-left corner. Visually – and also semantically – similar images have thus been mapped near each other on the map.

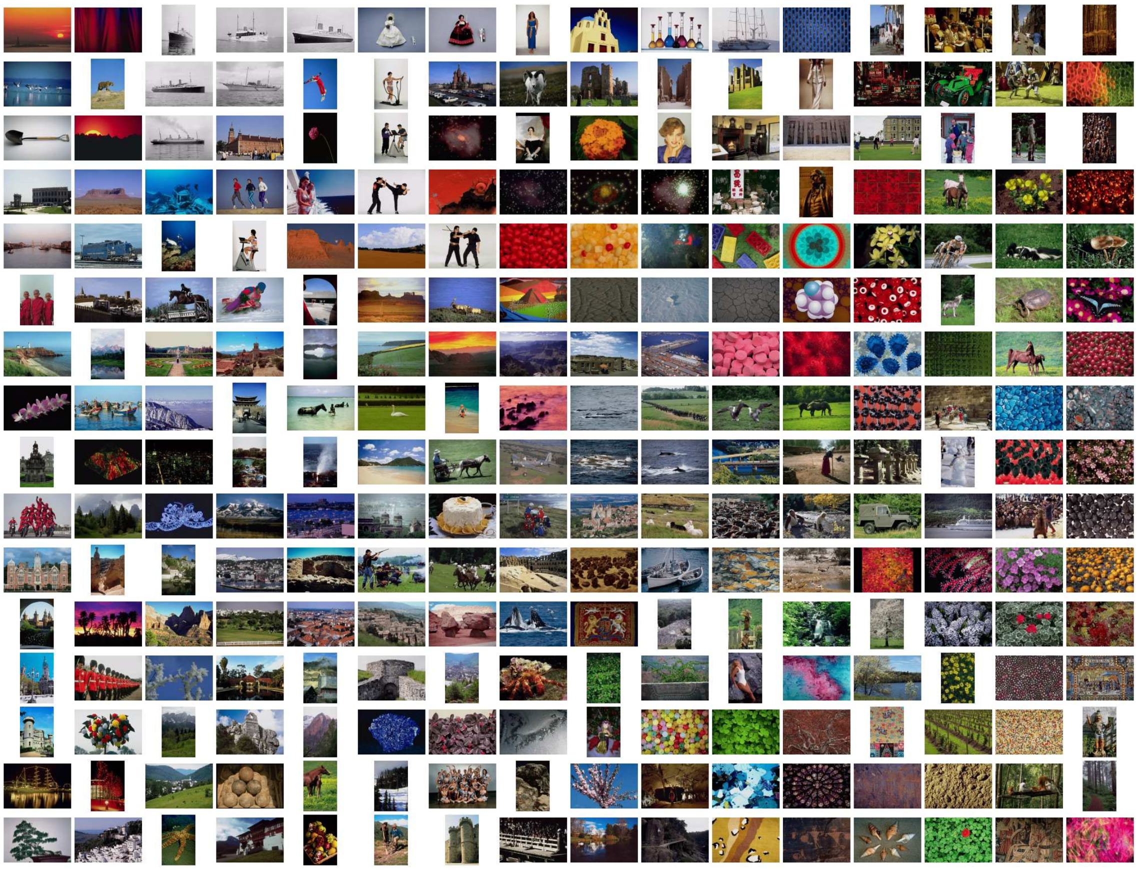

Figure 7.1: The surface of a 16×16-sized TS-SOM level trained with the MPEG-7 *Edge Histogram* descriptor.

Self-organizing relevance feedback

Each image seen by the user of the system is graded by her as either relevant or irrelevant. All these images and their associated relevance grades are then projected on all the SOM surfaces. This process forms on the maps areas where there are 1) many relevant images mapped in same or nearby SOM units, or 2) relevant and irrelevant images mixed, or 3) only irrelevant images, or 4) no graded images at all. Of the above cases, 1) and 3) indicate that the corresponding content descriptor agrees well with the user's conception on the relevance of the images. Whereas, case 2) is an indication that the content descriptor cannot distinguish between relevant and irrelevant images.

When we assume that similar images are located near each other on the SOM surfaces, we are motivated to spread the relevance information placed in the SOM units also to the neighboring units. This is implemented in PicSOM by low-pass filtering the map surfaces. All relevant images are first given equal positive weight inversely proportional to the number of relevant images. Likewise, irrelevant images receive negative weights that are inversely proportional to the number of irrelevant images. The overall sum of these relevance values is thus zero. The values are then summed in the BMUs of the images and the resulting sparse value fields are low-pass filtered. Figure 7.2 illustrates how the positive and negative responses, displayed with red and blue map units, respectively, are first mapped on a SOM surface and how the responses are expanded in the convolution. Content descriptors that fail to coincide with the user's conceptions produce

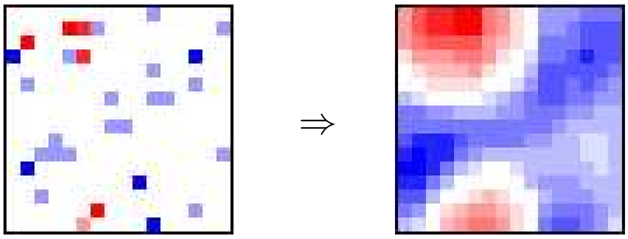

Figure 7.2: An example of how a SOM surface, on which the images selected and rejected by the user are shown with red and blue marks, is convolved with a low-pass filter.

lower qualification values than those descriptors that match the user's expectations. As a consequence, the different content descriptors do not need to be explicitly weighted as the system automatically takes care of weighting their opinions.

After removing duplicate images, the second stage of processing is carried out. Now, the qualification values of all images in this combined set are summed up on all used SOMs to obtain the final qualification values for these images. Then, a set of images with the highest qualification values are returned as the result of the query round.

MPEG-7 content descriptors

Development of content-based image retrieval techniques has suffered from the lack of standardized ways for describing visual image content. Luckily, MPEG-7 international standard is now emerging as both a general framework for content description and a collection of specific, agreed-upon content descriptors. MPEG-7 aims at standardizing the description of multimedia content data. It defines a standard set of descriptors that can be used to describe various types of multimedia information. In the scope of our work, the most relevant part of MPEG-7 is the implementation of a set of still image descriptors. Recently, we have integrated the standard MPEG-7 content descriptors into PicSOM [5] and shown that they can be successfully used with it.

References

[1] J. T. Laaksonen, J. M. Koskela, S. P. Laakso, and E. Oja. PicSOM - Content-based image retrieval with self-organizing maps. *Pattern Recognition Letters*, 21(13-14):1199–1207, November 2000.

[2] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja. Self-Organizing Maps as a relevance feedback technique in content-based image retrieval. *Pattern Analysis and Applications*, 4(2+3):140-152, June 2001.

[3] G. Salton and M. J. McGill Introduction to Modern Information Retrieval. McGraw-Hill, 1983.

[4] P. Koikkalainen and E. Oja. Self-organizing hierarchical feature maps. In *Proc. International Joint Conference on Neural Networks*, vol. II, pages 279-285, Piscataway, NJ, 1990.

[5] J. Laaksonen, M. Koskela, and E. Oja. PicSOM – Self-Organizing Image Retrieval with MPEG-7 Content Descriptors. Accepted to *IEEE Transactions on Neural Networks*, 2002.

7.2 Fault analysis of running paper web

Jukka Iivarinen

On-line inspection is an essential part of modern web or sheet manufacturing. The purpose of an inspection system is to detect and classify those defects which impair the quality of a product as compared to the requirements set by a user. Typical characteristics of web manufacturing processes are the large values of web width and production speed. The web width of a modern paper machine may exceed 9 metres and its speed may reach 30 m/s. Such a machine makes about $3{\cdot}10^8 \mathrm{mm}^2$ of paper each second, and all that production has to be inspected with a 100 % coverage and with a resolution even less than 1 mm^2.

Historically defect detection of web surfaces has been accomplished by hardware solutions for thresholding and matched filters. These techniques have made possible to detect only the most basic defect types. Detection of more complicated but critical defect types has remained unreliable. In this project more sophisticated methods have been developed. Use of texture and more complicated classifiers have become possible due to new sensor technology, increased calculation capability of computers and specialized hardware. Surface inspection has been studied in our laboratory since 1995 [1, 2]. The main interest has been the detection and classification of defects in a running paper web.

Overview of the method

The proposed system model for web inspection has two phases: a segmentation phase (defect detection) and a classification phase (defect classification) (Figure 7.3). In the segmentation phase feature extraction is done and potential defect areas are marked. In the classification phase features describing the shape and internal structure of defects are extracted and defects are classified to different defect classes.

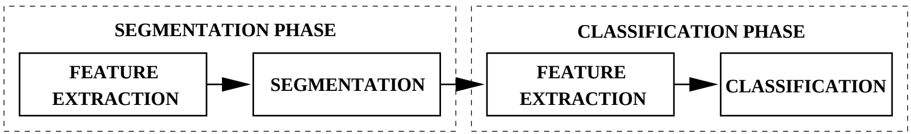

Figure 7.3: The proposed method for web inspection consists of two phases.

Segmentation phase A modified self-organizing map, called the statistical SOM, is used to separate defects from a fault-free web. A set of co-occurrence matrix features is used for the texture description. The feature distribution of fault-free samples is estimated with the statistical SOM. Now fault detection is based on the following idea: an unknown sample is classified as a defect if it differs enough from this estimated distribution. The segmentation scheme is depicted in Figure 7.4.

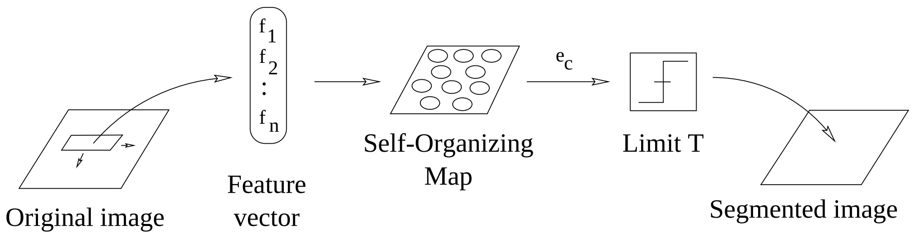

Figure 7.4: The segmentation scheme.

Classification phase In defect classification the SOM is used to cluster unknown defects. It finds the classes (clusters) that are inherent to defect samples. These classes are then given an explanation by hand. Classification is based on the visual appearance of defects. Five simple shape descriptors are used to characterize the shape, and a gray level histogram and some co-occurrence matrix features are used for the internal structure.

An example defect and its features are depicted in Figure 7.5(a). In Figure 7.5(b)-(d) are the classification results. Each column shows the best-matching defect when using different features. In the bottom row are the numbers and the weight vectors of the best-matching SOM units. The first and the second row show one of the typical defects that belongs to the best-matching unit. The labels of the best-matching units are *elongated, smooth shape* (SSD), *light spot* (GLH), and *light spot* (TEX). The final classification is then *elongated, smooth, light spot*.

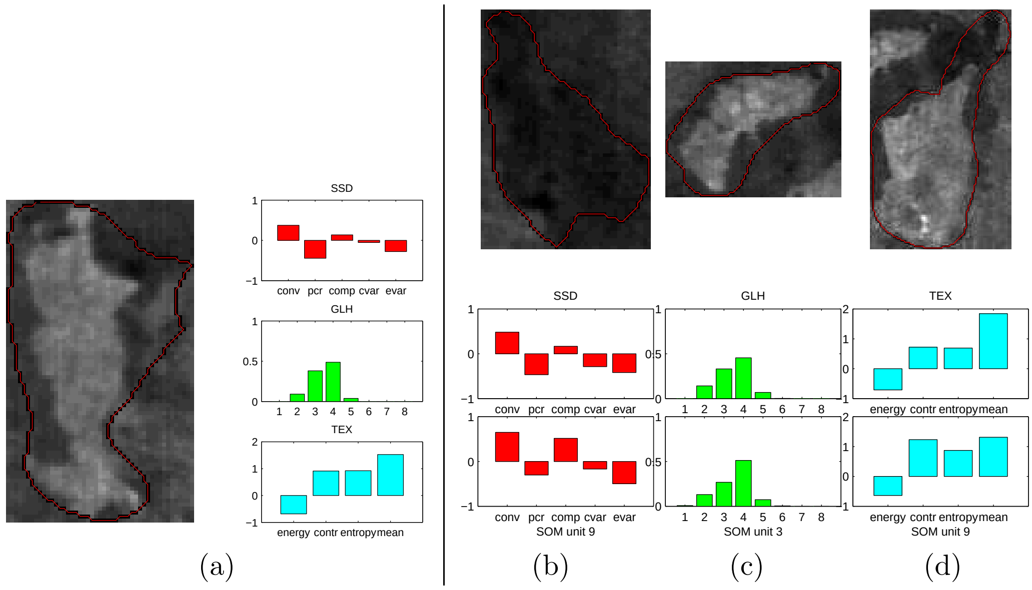

Figure 7.5: (a) A defect and its features. (b)-(d) The best matching defects to the defect in (a) with respect to different features.

Conclusions

The two-stage approach offers advantages when considering real-time processes. Speed requirements of real-time defect detection and classification can be satisfied by splitting the procedure into two stages. The defect detection is a suitable part to hardware implementation, whereas more time can be spend on classifying found defects. In defect detection the classifier is taught with examples of fault-free surface while in defect classification shape and internal structure characteristics of defects are learned from examples.

References

[1] J Iivarinen. Unsupervised Segmentation of Defect Images. In D. P. Casasent, Ernest L. Hall, editors, *Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision*, Proc. SPIE 4572, pp. 488–495, 2001.

[2] J. Iivarinen, K. Heikkinen, J. Rauhamaa, P. Vuorimaa, and A. Visa. A defect detection scheme for web surface inspection. *International Journal of Pattern Recognition and Artificial Intelligence*, 14(6):735–755, Sept. 2000.

7.3 Content-based retrieval of defect images

Jussi Pakkanen and Jukka Iivarinen

A need for efficient and fast methods for content-based image retrieval (CBIR) has increased rapidly during the last decade. The amount of image data that has to be stored, managed, browsed, searched, and retrieved grows continuously on many fields of industry and research.

In this project we have taken a noncommercial CBIR system called PicSOM, and applied it to a database of surface defect images. PicSOM has been developed in our laboratory at Helsinki University of Technology to be a generic CBIR system for large, unannotated databases. We have made some modifications to the original PicSOM system that affect mostly feature extraction and visualization parts of PicSOM. As an extra problem-specific knowledge we have segmentation masks for each defect image. This information is utilized in PicSOM so that feature extraction is only done for defect areas in each defect image. The project started on March 2001.

Overview of the method

Interpretation of defect images is a demanding task even to an expert. The defect images concerned in this work contain surface defects, and they were taken from a real, online process. There exist several defect classes (e.g. dark and light spots, holes, and wrinkles) that are fuzzy and overlapping, so it is not possible to label defects unambiguously.

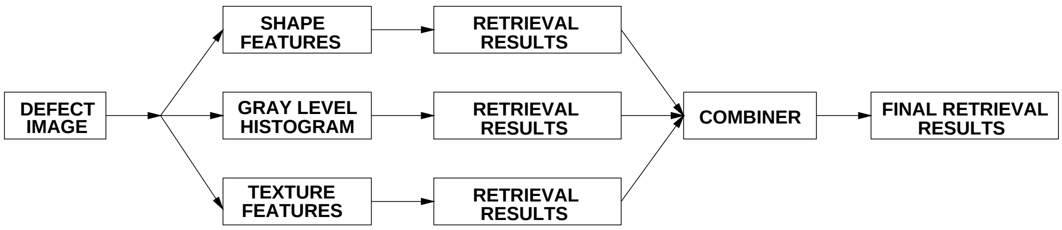

Figure 7.6: The structure of a CBIR system for defect images.

In the present work we have adopted the PicSOM system as our content-based image retrieval (CBIR) system and embedded a defect image database into PicSOM. PicSOM has several features that make it a good choice for our purposes. The basic structure of our defect PicSOM system is shown in Figure 7.6. Given a new defect image, three different feature sets are extracted, similar defects according to each feature set are retrieved from the database, and these retrieved images are combined to produce the final, combined set of similar defect images.

Features for defect characterization Several types of features can be used in PicSOM for image querying. These include features for color, shape, texture, and structure description of the image content. When considering defect images, there are two types of features that are of interest: shape features and internal structure features. Shape features are used to capture the essential shape information of defects in order to distinguish between differently shaped defects, e.g. spots and wrinkles. Internal structure features are used to characterize the gray level and textural structure of defects.

Six simple shape descriptors are used to characterize the shape of a defect. The descriptors are: convexity, principal axis ratio, compactness, circular variance, elliptic variance, and angle. The internal structure can be regarded as a distribution of gray levels in a defect. This distribution can be characterized as a simple gray-level histogram. In addition, four texture features, energy, contrast, entropy, and mean, calculated from the co-occurrence matrix are also used.

Experiments

The problem at hand is now the following one: Given a new defect or a set of defects, retrieve similar defects that might have appeared previously. The retrieval is based on shape and internal structure features, so there is no need for manual annotation or labeling. The defect database has almost 13000 defect images that were taken from a real, online process. The images have different kinds of defects, e.g. dark and light spots, holes, and wrinkles. They are automatically segmented beforehand so that each defect image has a gray level image and a binary segmentation mask that indicates defect areas in the image. The image database was provided by our industrial partner, ABB Industry Oy.

The example query images in Figure 7.7 show that the system works quite well. Under the TS-SOMs are the images selected by the user (or the query images), and on the bottom are the images returned by the PicSOM system. All returned images are visually similar to the query images. The system retains a similar level of success when queried with different types of defects. The true power comes from combining the maps. The PicSOM engine combines the various maps in a powerful manner, yielding good results.

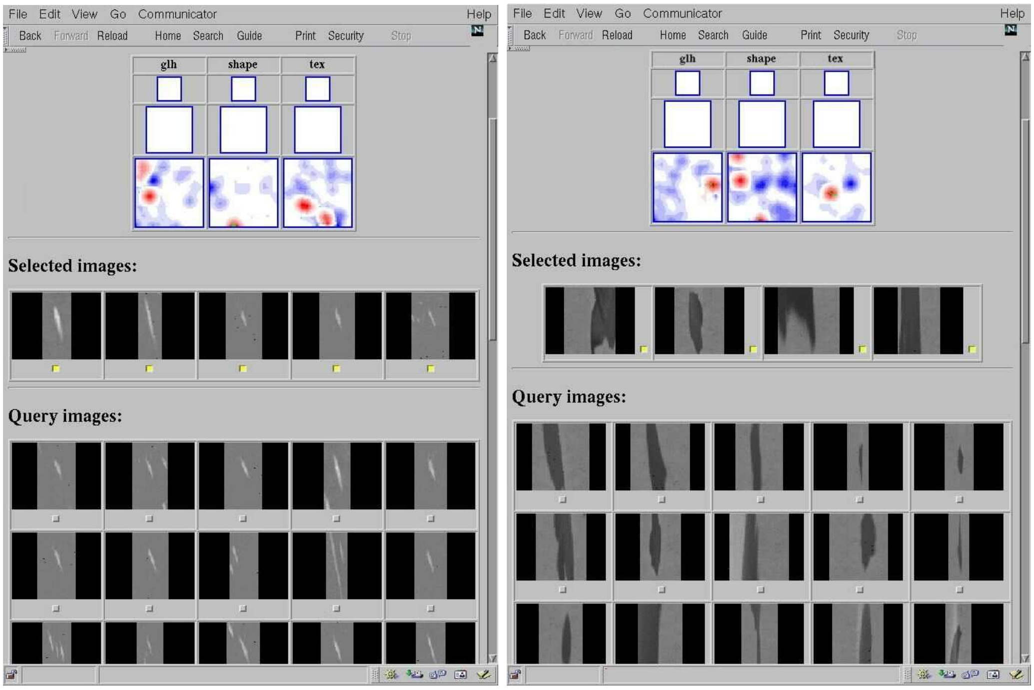

Figure 7.7: Some example PicSOM queries.

Conclusions

In this project a noncommercial content-based image retrieval (CBIR) system called PicSOM is applied to retrieval of defect images. New feature extraction algorithms for shape and internal structure descriptions are implemented in the PicSOM system. The results of experiments with 13000 surface defect images show that the system works fast with good retrieval results.

7.4 Attribute trees in image analysis

Markus Peura

Introduction

The appearance of many natural objects is irregular and indefinite. Spatial hierachy and dynamical behaviour are often additional challenges in the tasks of computer vision: automated recognition, classification, tracking, and prediction. Irregular objects are encountered frequently in biology, medical sciences, meteorology, and geosciences (Fig. 7.8).

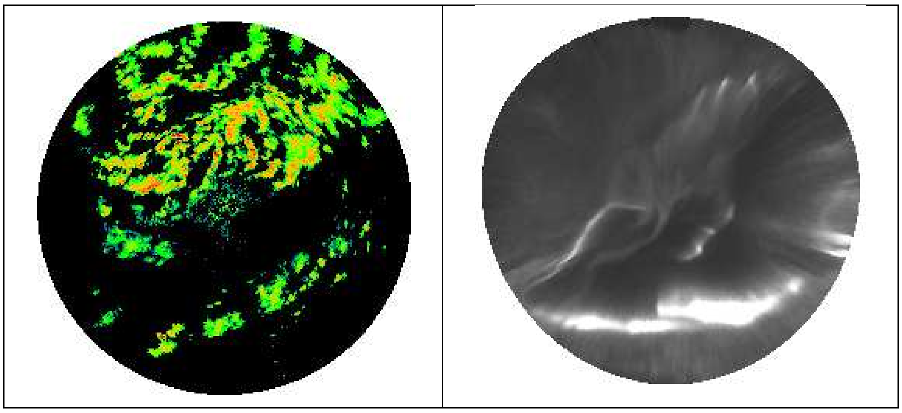

Figure 7.8: Examples of irregular natural images: precipitative clouds detected by a weather radar (left) and auroral lights captured by an all-sky camera (right).

Basic idea

The intensities of precipitation (rain) in a radar image can be interpreted analogously to terrain elevation on a topograhic map. The spatial hierarchy of the intensity segments can be extracted as a *tree*. The tree is a presentation of a topology and thus rotation, translation and scaling invariant. If nodes (or edges) are attached additional data - such as intensities, sizes, shape descriptions, coordinates, or types of objects - an *attribute tree* is obtained. An attribute tree is a compact presentation of an image as it preserves a significant part of the visual information and still has a storage size equal to a fraction of that required by the original image. Consequently, attribute trees can be processed faster than the original images.

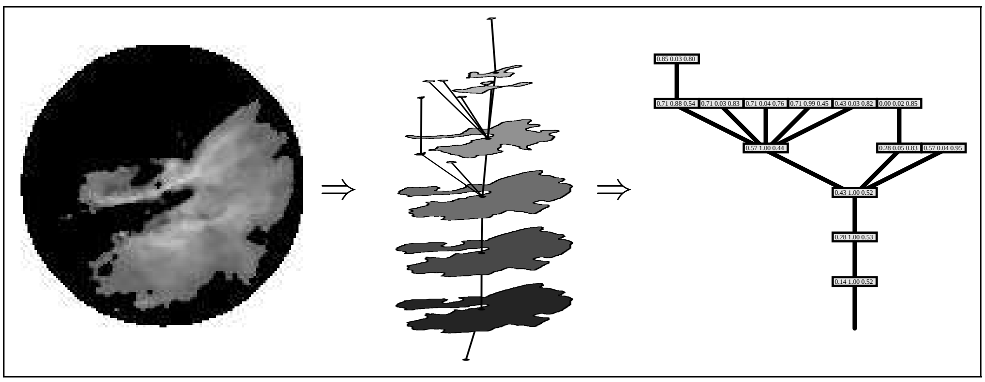

Figure 7.9: An image, its topology and attribute tree.

Fast tree matching

One of the central innovations in this study has been *the heuristic matching scheme for unordered attribute trees.* Denoting the size of a tree by N and the maximal out-degree by d, the proposed matching scheme has the complexity of $O(Nd \log d)$ while the exact algorithms are of $O(N^3)$. The obtained speed is essential in real time applications, like in *weather nowcasting* based on radar data.

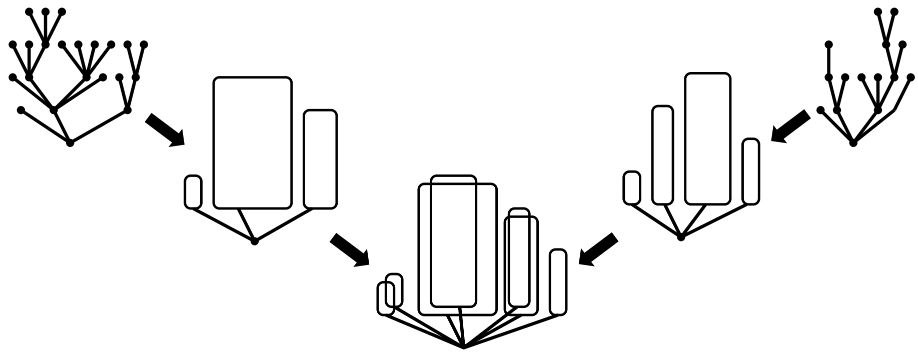

Figure 7.10: Illustration of index-based heuristic matching.

The scheme is based on dividing trees recursively into subtrees. The subtrees are matched according to indices, which have been calculated in advance using cumulative updating rules. In other words, exhaustive matching of subtrees is replaced by matching points in space, yielding an approximate matching result. The descriptors used in this study are height, child count (out-degree), node count (tree size), vertical centroid, and branching variance. If the node count is regarded as the mass of a tree, the centroid reflects the vertical distribution of the mass. The branching variance measures structural irregularity. A step of the matching scheme is illustrated in Fig. 7.10, where height and node count have been depicted as the height and width of the rectangles, respectively. When matching attribute trees, indices are calculated for attributes in similar fashion.

In comparison with the exact algorithms, the proposed linear-time approximation performs reliably with trees of up to a hundred nodes (Fig. 7.11). The proposed heuristic approach seems to outperform the exact algorithms even further when the quality criterion is the absolute number of matched nodes per computation time.

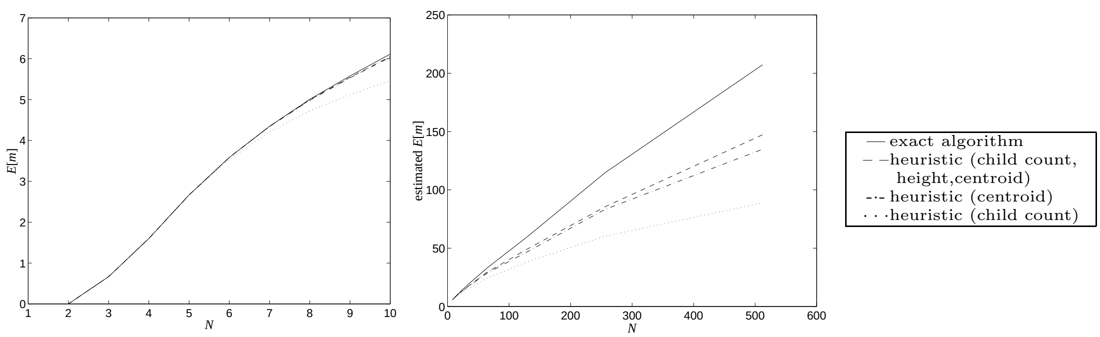

Figure 7.11: Matches found for non-isomorphic trees of size 1, 2, ..., 10 (complete sets) and 8, 16, ..., 512 (randomly sampled sets).

The Self-Organizing Map of attribute trees

After matching two trees, the trees can be merged in a *continuous fashion*. Such mixtures of matched trees are essentially interpolations and have applicability in learning systems that typically involve template generation through averaging. The *self-organizing map of attribute trees* [1, 2, 3] is an extension of the standard self-organizing map (applying vectors); the key issues is in the revised definitions of distance functions and averaging operations. An example of a self-organizing map of attribute trees is shown in Fig. 7.12, indicating how the structure and the attributes of trees can be ordered in continuous fashion.

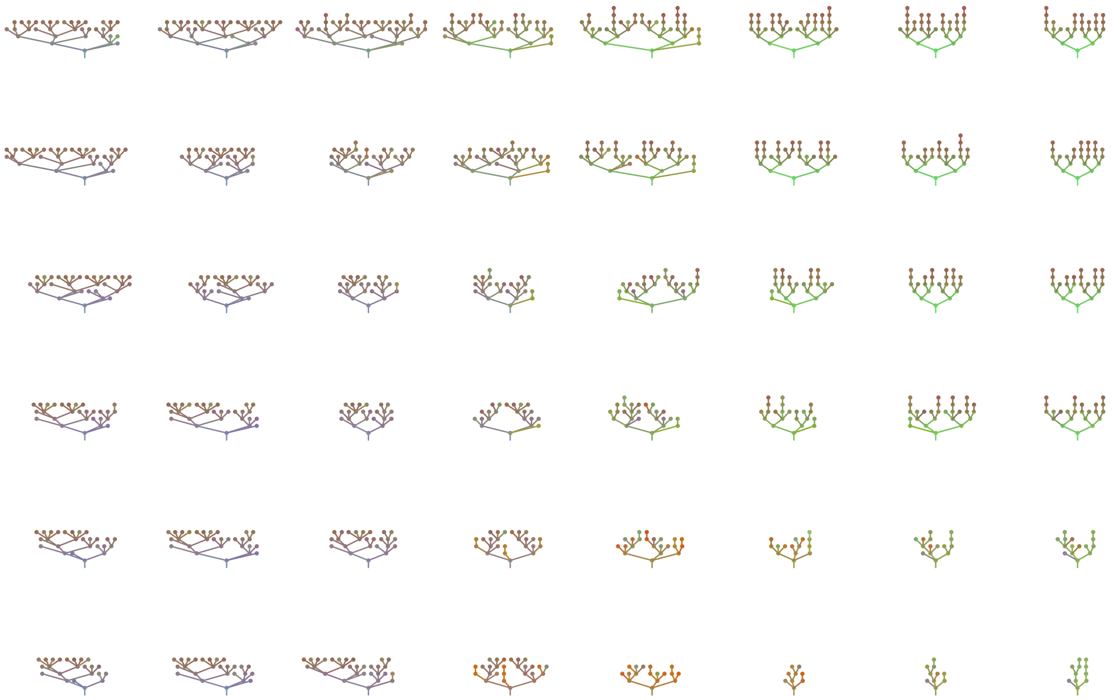

Figure 7.12: A self-organizing map of attribute trees.

References

[1] M. Peura. *Attribute trees as adaptive object models in image analysis.* D. Sc. thesis. Helsinki University of Technology, Laboratory of Computer and Information Science, February 2001.

[2] M. Peura and M. Syrjäsuo. Flexible Heuristic Matching of Attribute Trees. In *12th Scandinavian Conference on Image Analysis (SCIA'01)*, pages 207–213 Bergen, Norway, June 2001.

[3] M. Peura. Attribute trees in image analysis — heuristic matching and learning techniques. In *10th International Conference on Image Analysis and Processing, (ICIAP'99)*, Venice, Italy, September 1999. IEEE Computer Society.