Focusing Keywords to Automatically Extracted Image Segments Using Self-Organising Maps

Ville Viitaniemi and Jorma Laaksonen

Adaptive Informatics Research Centre
Helsinki University of Technology
P.O.Box 5400, FIN-02015 TKK, Finland
ville.viitaniemi@tkk.fi, jorma.laaksonen@tkk.fi

1 Introduction

In this chapter we consider the problem of keyword focusing. In keyword focusing the input data is a collection of images that are annotated with a given keyword, such as "car". The problem is to attribute the annotation to specific parts of the images. There exists plenty of suitable input data readily available for this data mining type of problem. For instance, parts of the pictorial content of the World Wide Web could be considered together with the associated text. We propose an unsupervised approach to the problem. Our technique is based on automatic hierarchical segmentation of the images, followed by statistical correlation of the segments' visual features, represented using multiple Self-Organising Maps. The performed feasibility study experiments demonstrate the potential usefulness of the presented method. In most cases, the results from this data-driven approach agree with the manually defined ground truth for the keyword focusing task. In particular, the algorithm succeeds in selecting the appropriate level of hierarchy among the alternatives available in the segmentation results.

The rest of the text is organised as follows. Section 2 reviews related fields of image analysis and conceptually introduces the proposed keyword focusing technique. In Section 3 we present an overview of some of the relevant principles and techniques in image content analysis. Section 4 discusses the role of the image segmentation subsystem in image content analysis, along with reviewing some image segmentation principles. In Section 5 we define and discuss the keyword focusing problem. We also propose and conceptually describe the statistical correlation method for keyword focusing on the conceptual level. Section 6 describes in detail the proposed technical implementation of the statistical correlation method. Section 7 reports the results of feasibility studies we performed with our implementation in two databases. In Section 8 we present our conclusions and future views.

2 Background

In recent years the world around us has become increasingly visual. The technological evolution has made it possible to produce and store huge amounts of image data. The images can easily be indexed and searched according to some simple attributes or metadata, such as the date of archival. However, usually it is the content of the image that is more important for the future use of the image.

When we imagine us humans describing an image, quite often we would list what different parts the image contains, and then possibly describe the major parts in more detail. In this light, partitioning the image to disparate parts and describing it in terms of the content and relationship of these parts appears to be a promising approach. Indeed, image segmentation is often a crucial part in computerised image understanding systems, e.g. [42, 34, 1]. Part-based object representations also provide the basis for many theories of object recognition in biological vision systems [43, 5, 31].

The traditional approach to image indexing has been to annotate images textually according to their contents. Then the database of annotations can be managed with conventional methods developed for textual databases. Unfortunately, the manual annotation method has severe shortcomings. In many cases the data volume is so huge that manual annotation is plainly impossible. In addition, the manual annotations can capture only a small subset of descriptions and interpretations of the images. The subjectivity and language specificity of manual annotations makes their use nontrivial.

Against this background it is easy to see that automatic methods for content-based image characterisation would be highly desirable. The aim of content-based image retrieval (CBIR) research is to produce systems that automatically analyse the contents of given images, after which image databases can be queried by visual content. An important subproblem of CBIR is the automatic content analysis. Here one of the major challenges is the bridging of the large *semantic gap* between low-level image descriptors traditionally used in computer vision and the user's desire to query the systems with high-level semantic concepts.

The natural language, i.e. words, readily offers a symbolic representation of semantic concepts. Recently, the problem of matching words and images has attracted considerable research interest. Using textual annotations as proxy might offer a helpful approach to the semantic similarity assessment problem in CBIR. In the usual setting the word–image correspondence is regarded as a machine learning problem. The correspondence is learned from a set of annotated training images, after which the learned correlations can be used to automatically annotate any images.

The correspondence between entities on different semantic levels can be seen as an instance of *emergence* [44]. Emergence is a process where a new, higher-level phenomenon results from co-operation of a large number of elementary processes. Thus, understanding and modelling emergence provides an approach to overcoming the semantic gap. One essential element of emergence appears to be the involvement of large amounts data or processes. As there are lots of example data available where images are described with words, the image–keyword correspondence is a good example of a problem that can be naturally studied from the viewpoint of emergence.

In this chapter, we show how soft computing techniques, more specifically the Self-Organising Map (SOM) [26], can be used in modelling the phenomenon of emergence. In the following we propose a mechanism for the emergence of semantic concepts from low-level features of image parts. We consider a learning problem that differs somewhat from the typical automatic annotation setting. We set the goal of the learning system to be *keyword focusing*, i.e. we consider the problem of identifying the areas in an image that correspond to the keyword when we already know that the keyword must correspond to some parts of the image.

In this chapter we approach the keyword focusing problem using an unsupervised, data-driven method. We propose a two-stage mechanism that consists of a feedforward image representation front end, followed by an inference stage. The front end consists of image segmentation and feature extraction components, and a neural vector space quantizer utilising the SOM. The inference stage is based on finding statistical correlations between the annotating keywords and the representation formed by the front end. In addition to this baseline approach, we also consider interlinking the image segmentation step with the inference algorithm. In this case the unsupervised segmentation component produces a hierarchy of alternative tentative segmentations. The final selection among these segmentations is made during the keyword focusing process.

3 Image Content Analysis

Large part of computer vision research can be regarded to revolve around the analysis of image content. In this presentation we consider some aspects of the content analysis problem starting from the needs of content-based image retrieval (CBIR), e.g. [41, 20]. In content-based image retrieval image databases are indexed with descriptors derived from the visual content of the images. Image content analysis is thus an essential subproblem of CBIR, even though it only seldom is addressed explicitly.

Most CBIR systems are concerned with approximate queries where the goal is to find images visually similar to a specified target image. CBIR thus requires methods for both *characterisation* of image content and meaningful *similarity assessment* of the characterisations. A popular method of combining these two aspects is to employ the *vector space model* depicted in Figure 1. The idea behind the vector space model is to represent each image as a point or more generally, a collection of points, in a vector space. When a distance metric is defined for the vector space, the similarity assessment problem reduces

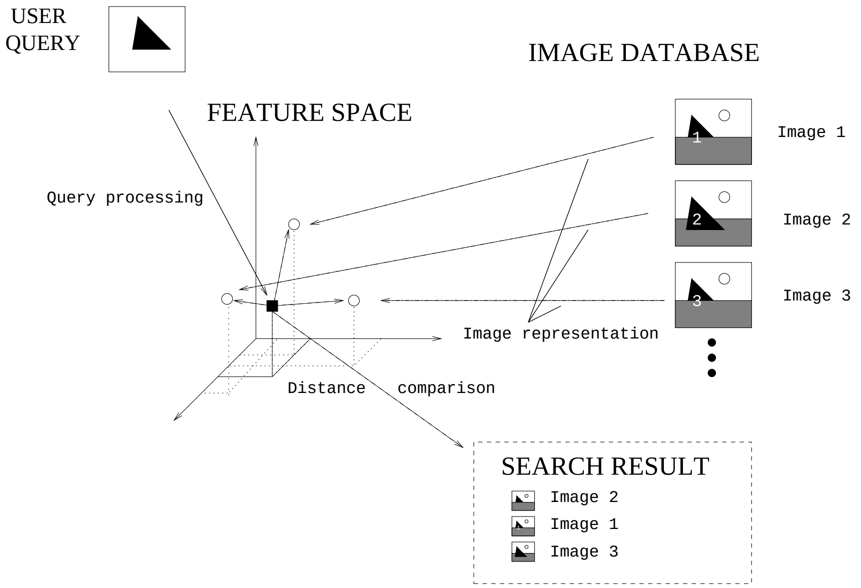


Fig. 1. In the vector space model of image retrieval both the database images and user queries are mapped into points in the feature (vector) space. The proximity of points in the feature space is taken as an indication of image similarity.

to a geometric problem, e.g. finding points closest to the point that is the representation of a given example image or otherwise specified query.

3.1 Semantic Levels of Image Content Characterisation

A central problem in content-based image retrieval is the *semantic gap* between the high-level semantic concepts we humans use in our reasoning and the low-level visual features directly available to computers for their information processing. This question can be elaborated by a categorisation of the depth of image content characterisation into three semantic levels, as proposed in [12]:

Level 1 contains characterisations by primitive features, such as colour, shape or spatial location of image regions. The features used on this level are both objective and directly obtainable from the image itself, without the need to resort to external knowledge bases. Most of the existing systems for general-purpose CBIR operate on this level.

Level 2 uses derived (sometimes called logical) attributes, involving some sort of logical inference about the identity of objects in the image. A useful distinction can be made between two sublevels: a) identification of objects of a given type, and b) identification of individual objects or persons. Usually one needs to use some kind of an outside knowledge store to answer queries on this level.

Level 3 uses abstract attributes for image content characterisation. This means that significant high-level reasoning about the meaning and purpose of the objects in the image must be carried out. Also this level can usefully be divided into two: a) identification of named events or types of activity (e.g. finding pictures of Finnish folk dancing), and b) identification of pictures with emotional or religious significance (e.g. finding pictures of suffering). A need for image similarity queries on this level is often encountered in practice.

Often (e.g.[21]) image retrieval based on the similarity on the semantic levels 2 and 3 is collectively termed as *semantic image retrieval*. In the practical systems of the present, the most significant semantic gap exists between the levels 1 and 2.

3.2 Feature Types by Spatial Extent

In the vector space model the visual content of images is represented by feature vectors. The feature calculation component of such an image analysis system thus considers the pixel representation of an image and forms a corresponding fixed-length feature vector according to some rule. In the keyword focusing application considered here we are interested in differentiating between the parts of the images. We therefore extend the model and allow an image to be represented with a set of feature vectors, each one corresponding to a part of the image.

Approaches to feature calculation can be classified according to the way the image is segmented, i.e. how its pixels are divided among the set of feature vectors. The following characterisations are an augmented version of [41]:

1. *Strong segmentation* denotes the segmentation of an image according to real world objects.
2. *Weak segmentation* is the term used for data-driven grouping of an image into homogeneous regions.
3. *Interest point detection* identifies visually salient locations in an image.
4. *Sign spotting* reveals whether objects of (almost) fixed shape and a known semantic interpretation are present in an image and where they are located.
5. *Partitioning* divides images into geometrical areas, regardless of the image data.

Grouping of pixels leads to various types of feature representation:

1. *local features* (correspond to sign spotting and interest point detection)
2. *object features* (correspond to strong segmentation)
3. *region features* (correspond to weak segmentation or partitioning)
4. *global features* (a special case of partitioning where all pixels belong to the same region)

These categories vary in the extent they combine feature values from different parts of an image. In one extreme is the local feature approach which describes each found object and its feature values separately. The other extreme is the global feature approach, which combines feature values from the whole image.

In the current context we are interested in correlating the semantic interpretations with locations in the images. The most relevant feature types to this end are local, object and region features. We limit ourselves into the automatic analysis of the image content. This rules out object features, for the bare reason that in a general setting, strong segmentation is still an unsolved problem (cf. also Section 4). On the other hand, by definition the connection between the features resulting from sign spotting and the semantics is already known by definition and is therefore uninteresting.

The local features resulting from interest point detection are the first of the two remaining feature types. Such local descriptors are calculated for neighbourhoods of the interest points and are usually made invariant to some image transformations, such as rotation and scaling. Examples of local features include steerable filters [18], SIFT descriptor [32] and shape context [4]. For a review on the subject see [33]. Local features have proven to be successful in object recognition tasks, for example in the recent PASCAL Visual Object Classes challenge [13] many of the best performing approaches used the SIFT representation.

When correlating the image locations with semantic interpretation of the content, the local feature based approach must overcome two challenges. Firstly, one interest point is usually not enough to robustly define an object, a configuration of multiple interest points must be considered simultaneously. The typical solutions to that problem do not conform easily to the vector space model. The second challenge is that it is desirable to attach semantic interpretations to whole image regions, not only to some isolated points such as corners. When using local features, a separate processing stage must be devised to perform this inference.

Region features are the other remaining feature type that can be used for keyword focusing. In the subsequent sections we explore the application of region features, calculated for automatically obtained image segments, to the keyword focusing task. For the experiments we have used a large variety of different low-level features. A detailed description of them is given in Section 6.2.

4 Image Segmentation for Image Content Analysis

In this section we will briefly review some of the key issues related to the use of image segmentation as a processing stage in an image content analysis system. We first discuss the requirements the segmentation subsystem should fulfill in order to be useful in this context. Then we review the principles behind the existing segmentation methods. Finally, we shortly discuss the use of the segmentation results for image content similarity assessment.

4.1 Requirements for Segmentation Subsystem

Generic, complete and to-the-pixel accurate unsupervised segmentation is virtually impossible [11]. Fortunately, less-accurate segmentations are useful enough for applications of image content analysis. Often it is enough to be able to locate salient regions of the image and calculate robust descriptions for them. In contrast, some types of visual descriptors, e.g. shape features, are more dependent on the segmentation accurately identifying the outline of objects or regions.

Usually the desired result of image segmentation is a classification of each pixel as having exactly one segment label, i.e. a partitioning of the image. There exist some problems concerning such a disjoint labelling as it is quite unnatural and not in accordance with human perception. At the first glance more attention is certainly paid to larger regions, but subsequently also the smaller, salient image details are registered equally well. The “mental segmentation map”, that is, the decomposition of the image into its parts in the mind of the observer, is not a flat two-dimensional map, but some sort of hierarchy of image regions results. In that map, an image region may be a part of different objects on different levels of the hierarchy.

As an example of hierarchically related regions we could think of a scene from the Orient in Figure 2. On the root level the image can be divided to three segments: sky, the fortress and the ground level (with the road and the people). The “fortress” segment is marked with a white outline in the leftmost figure. The fortress is further divided into wall and tower parts, the tower consists of two levels, and if we look closer at the upper level we can distinguish columns and two horisontal fascias. These steps are depicted in the rightmost column of Figure 2.

There is no question whether the use of this kind of hierarchy information could potentially be beneficial. The issue is more like whether we are able to find a practical way to incorporate this kind of information into an image content analysis system. In Section 6 we will present our implementation of a keyword focusing system that utilises results of hierarchical segmentation.

4.2 Types of Segmentation Methods

Local Methods

Local segmentation methods try to detect local discontinuities in image attributes. These local indications of segment boundaries can further be grouped together to form contours of objects. The problem with local methods is that information occurring on a larger scale is not used. The local

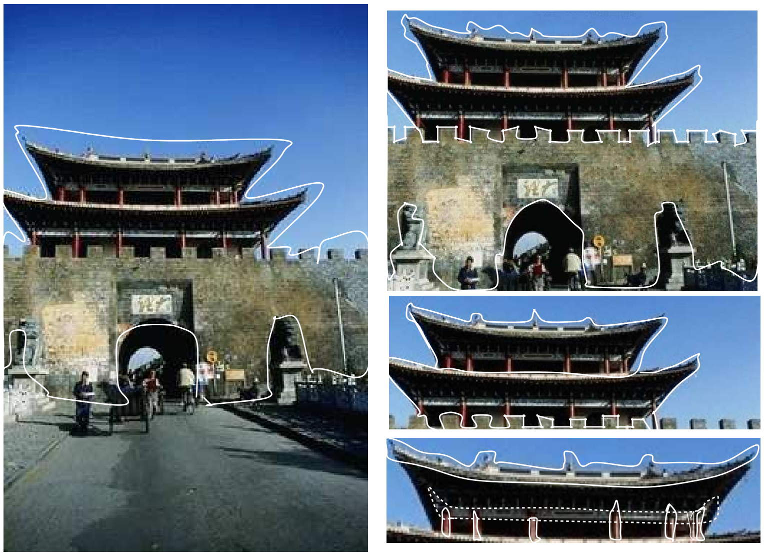


Fig. 2. Segment hierarchy of an image from the Orient. Subfigures display nested segments with increasing detail.

neighbourhoods—often just a single pixel or a couple of pixels—are considered independently of each other and the presence of boundaries and their attributes are determined separately for each neighbourhood. After edge detection, contours—possibly closed—are identified by linking the found edge segments with each other.

Often the information contained in the local neighbourhoods is not enough to reliably locate edges. In addition, the edge information from different local neighbourhoods may be contradictory or ambiguous.

Area-Based Methods

Area-based segmentation methods try to locate homogeneous areas in the images. One of the problems of this approach is to determine what is homogeneous enough to be considered as a single region. Another problem is the complete negligence of local edge information, resulting in difficulties in locating reasonable region boundaries even in the case of unambiguous edge information.

Region growing, region merging and region splitting are widely-used area-based segmentation methods. In region growing new image pixels are appended to a seed region as long as the newly-created region still fulfills a homogeneity criterion. Region merging tends to merge adjacent regions if the resulting region stays homogeneous enough. Region splitting starts from in-homogeneous regions. They are split until the homogeneity region is fulfilled. Often a combination of these area-based methods is used together.

Global Methods

In traditional area-based methods the regions are considered separately or pair-wise at time. In contrast the global optimisation methods used in image segmentation aim at partitioning the image into disjoint homogeneous regions in an optimal way. When determining the optimality measure all the regions in the image are taken into account simultaneously. There are, however, three major difficulties in the global optimisation approach:

1. localisation of boundaries,
2. the tradeoff between the size and the number of regions,
3. the hierarchy of regions.

Boundary localisation problems are similar to those with area-based methods. The tradeoff between the size and number of regions means that if the number of regions is chosen to be small, the average size of regions must be large. Then it is likely that the segmentation misses small but salient or homogeneous regions due to *undersegmentation*. On the other hand, if the number of regions is large, the image becomes *oversegmented* into many small regions so that large regions and structures are completely fragmented.

As mentioned earlier, it would be desirable to have the segmentation algorithm to output the complete region relation hierarchy instead of only one level of it. In principle, there seems to be no reason why this could not be combined with the idea of global optimisation. However, it may not be straightforward to find a formulation that would both be rigorous and produce reasonable results. In methods with simple local decision rules the hierarchy would be easier to incorporate in some form. An issue also arises concerning whether the segmentation hierarchy should be optimised as a whole or whether single levels of the hierarchy could be optimised separately with some constraints forcing the adjacent levels to be compatible.

4.3 Use of Segmentation in Image Content Analysis

Let us now assume that our images have been segmented and the contents of the individual image segments described. How should the content of the images then be characterised, and how should the characterisations be compared? The latter problem arises particularly in CBIR, where the similarity of the characterisations needs to be assessed. As long as the interest lies in individual segments of images, no special techniques are needed.

Often, however, the goal of the system is not to assess similarity of individual segments but of images in whole. Then a method must be devised to propagate the segment similarity assessments to the image level. A straightforward alternative is to treat the segments as independent images. In that way the information about relationships between the segments—even about co-occurrence in the image—is lost. Furthermore, the user might be interested in whole images instead of segments. Another approach is to let the user define her interest in terms of example segments, e.g [7]. This, however, could be laborious, as example images contain several segments. A solution to this is to form an image similarity measure automatically from the segment similarities without segment-level user intervention [46, 8, 24]. A more rigorous alternative is to use generative models such as in [3, 6] to describe the joint distribution of semantic concepts and a blob representation of the image segments and then relegate the query to the concept level.

In the present application of keyword focusing, half of the issue becomes redundant as the objects of interest are indeed image segments. Still, we are left with the issue of converting image level examples to the realm of image segments.

5 Focusing Keywords in Image Segments

The main topic of our present work, focusing of keywords to image segments, is addressed in this section. In Section 5.1 the keyword focusing task is defined and its potential application areas discussed. Section 5.2 reviews the use of related techniques in some of the literature. In Section 5.3 we propose a specific technique, statistical correlation method, for accomplishing the task of keyword focusing. Finally, some characteristics of the statistical correlation method are discussed in Section 5.4.

5.1 Task of Keyword Focusing

In the keyword focusing problem the input is a set of images, all of which are annotated with a single keyword. The goal is to find the areas of the images that correspond to the keyword. This can be considered as an unsupervised machine learning problem: no labeled data is given that directly pinpoints the appropriate image locations. In our solution to the problem we additionally allow the learning system to use unlabeled auxiliary image data that is non-specific to any given keyword. The auxiliary data can be considered as part of the system in the sense that it remains the same regardless of the particular keyword at hand.

Related to the *intra-image keyword focusing*, where the task is to compare different parts of the same image, is the problem concerning the database-wide identification of regions corresponding to a keyword. We denote this database-wide ranking of locations according their likelihood to correspond to a specific keyword with the term *database-level keyword focusing*. Intra-image keyword focusing is a subproblem of the database-level keyword focusing in the sense that a solution to the latter problem gives also an answer to the first problem. One may also argue that solving the intra-image keyword focusing problem is a prerequisite to solving the database-level counterpart. This, in turn, explains why keyword focusing can be regarded as a potential tool in solving a part of the CBIR problem.

There are numerous collections of images available that can potentially be used as training data for the focusing problem with minimal preparation. One prototypical example is formed by the commercial illustration image databases that are annotated with image-level keywords, for example the Corel image gallery [10]. Many museums have annotated collections of digital images (e.g. [14]). Also any image collection that is partitioned into classes can be used by considering each class to represent a keyword. The images in the World Wide Web along with the text they appear next to form a more speculative and an overwhelmingly large instance of the keyword focusing problem.

The focusing problem is defined for all keywords, but of course a sensible solution can be expected only for keywords that in reality can be localised into some part in an image. For example, focusing the keyword “evening” to a specific part of an image is usually senseless.

5.2 Keyword Focusing in Image Content Analysis

Learning the *image–word correspondence* has attracted considerable research interest recently. Often the motivation has been to learn the correspondence from a set of training images and then apply it to the automatic annotation of a new set of unlabeled images. For the automatic annotation research the keyword focusing is mostly a by-product, whose results are not explicitly stated or analysed beyond its effect on the annotation performance. This is reasonable since pairing images with keywords is somewhat different problem than focusing the keyword further down to a certain location inside the image. On the image level the prediction is often easier as the various objects in the images are correlated. For instance, airplanes often appear together with sky. Yet the location of sky in the images should not be called “airplane”.

The *automatic annotation* of images is more directed to database-level relevance keyword problem than to intra-image keyword focusing. Some of the models for automatic image annotation consider the image as a whole (e.g. [38]) or use a rough geometrical division of the image (e.g. [35]). In [30] a geometrically formed multi-resolution two-dimensional hidden Markov model is used for modelling the images, but concepts are correlated with the model as whole, not with its individual constituent parts. A more popular approach, however, is to formulate the models more directly in terms of image segment–keyword correspondence (e.g. [36, 3, 37, 16, 15, 6, 23, 19, 45]).

In a broader sense, any image analysis or classification task can be seen as involving kind of keyword focusing if the problem solution includes the identification of relevant image locations. In such settings, the existence of a keyword is not explicit but implicit.

A straightforward approach to the focusing problem is a *bottom-up* or *feedforward* process where the input images are first segmented and the subsequent focusing is then reduced to selecting the one or ones among these segments that correspond to the keyword. An attempt to interlink the segmentation and focusing phases is given in [6] where a Markov random field (MRF) model combines the subtasks of segmentation on a coarse grid and assigning the keywords to segments. On the other hand, even this approach can be considered feedforward by regarding the coarse grid as a fixed segmentation result and the MRF algorithm as the focusing stage. In Sections 6.1 and 6.5 we propose an approach that also takes a step into this direction. In the proposed method we consider producing a hierarchical segmentation, among whose levels the focusing stage selects the most appropriate one of the alternative segmentations.

5.3 Statistical Correlation Method

Our proposed approach to the keyword focusing problem is based on statistically correlating the keywords and image segments. For this we need a set of training image data that consists of example images of the particular keyword class and an auxiliary image collection. The outline of the approach is the following:

1. Automatically segment the studied images.
2. Form feature space representations for the image segments.
3. Identify feature space regions that are more densely populated by the example image segments than by the auxiliary image segments.
4. Find the example image segments with feature representations in the relatively dense regions of the feature space and associate them with the keyword.

As we see, two mechanisms are responsible for the working of the focusing strategy: (1) the effect of concentration of the example image segments in certain regions of the feature space, and (2) the negative influence of the auxiliary data in regions where it concentrates. The approach is thus qualitatively similar to the *term frequency – inverse document frequency* (TF-IDF) formula [39] successfully used in natural language (text and speech) retrieval. Also the TF-IDF formula awards high correlation scores to terms that appear often in the relevant documents (example images), but such terms are punished that appear often also in the reference corpus (the auxiliary images).

Different from the natural language processing, in image processing the representation of documents is not readily available in the form of words, but must be constructed. In this sense the segment boundaries produced by the segmentation algorithm correspond to the set of words in a document on the syntactical level, and the feature representation of the segments corresponds to the actual identity of the words, i.e. the semantic level. The segments and their feature representations form together a *blob representation* [7] of the image. Blob representations are often used in image analysis, for example in automatic image captioning [3, 6].

The purpose of the auxiliary data in the focusing algorithm is to give a basis of comparison for determining whether a certain region of feature space is more commonly populated by the actual example data than by images in general. The discussion in [13] gives some perspectives to the issue of selecting appropriate auxiliary data. In principle, a collection of general domain images could be used as auxiliary data regardless of the specific focusing task. However, if the example images are known to come from a restricted domain, using a narrower domain data set as auxiliary data as well will probably result in better focusing performance.

5.4 Discussion

In the preceding we have approached the keyword focusing problem as a data mining problem. We are interested in discovering new, previously unknown relationships in the studied data. In this case, the novel discoveries are the correlations of specific parts of the studied images with the keywords. This approach can be contrasted with a pattern recognition approach, where the goal is to learn a model from a *labeled* training data set and generalise it to an unlabeled test set. This distinction corresponds to the one between unsupervised and supervised learning.

A shortcoming of the statistical correlation principle is that inadequate example data is likely to lead to spurious correlations. For instance, if the example images are such that the sun and buildings always coincide, the focusing algorithm can not be expected to tell these apart. When the correlations between objects are artefacts of the example set, increasing the number of example images can remove the ambiguous situations. Still, for genuine correlations such as “sky” and “airplane”, the required amount of example data could get impractically high before enough separate occurrences of the keywords would be observed.

Especially serious this problem can be in object–part hierarchies. For instance, motorbike wheels without a whole motorbike are rare in images, although also such images exist. It would seem that the statistical correlation learning should be augmented in these cases with some other learning principle that could take such object hierarchies more directly into account.

6 System Implementation

This section describes in detail the implementation of the system we propose for the unsupervised keyword focusing task. The system is implemented inside the PicSOM[1] CBIR software framework [27, 28, 29]. As an input, the system takes two sets of images: a set of images annotated with a certain keyword

[1] http://www.cis.hut.fi/projects/cbir/

(*positive examples*), and a set of auxiliary background images (*negative examples*) As the result of the processing the system produces a segmentation of the positive example images and ranks the segments according to their *relevance* to the keyword, i.e. the likelihood of the segments to correspond to the given keyword.

The proposed system consists of a feedforward pre-processing stage, followed by an inference stage. In the preprocessing stage, both sets of images are first hierarchically segmented and statistical visual features extracted from the segments. The features are grouped into multiple feature spaces that are finally quantized using a variant of the Self-Organising Map (SOM) [26]. The inference stage implements the statistical correlation method for keyword focusing simultaneously for parallel quantized feature spaces. As a post-processing step the produced ranking of the segments is re-ordered in an additional relevance propagation step so that the hierarchy information in the segmentation results is explicitly taken into account. As a result, the system is able to automatically select the most appropriate level of hierarchy in the hierarchical segmentations.

The rest of the section is organised as follows. Sections 6.1 and 6.2 discuss the hierarchical image segmentation and feature extraction methods, respectively. Section 6.3 describes the use of SOM in quantizing the feature spaces. In Section 6.4 we delineate the implementation of the statistical correlation principle. Section 6.5 describes the algorithm propagating the relevance within the segment hierarchy.

6.1 Automatic Image Segmentation

For the current experiments we have used a generic image segmentation method which is simple and somewhat rudimentary. Referring to the taxonomy of Section 4.2, the method is in essence a hybrid of area-based region merging combined with a local edge heuristics. The method partitions the images to a fixed number of segments that are homogeneous in terms of average colour in the CIE L*a*b* colour space [9].

The images in the database are segmented in two steps. In the first step ISODATA variant of K-means algorithm [40] with a K value 15 is used to compute an oversegmentation based on the colour coordinates of the pixels. This step typically results in a few thousand separate segments.

In the second step the segments are merged. The difference $d_{Lab}(r_1, r_2)$ in the average CIE L*a*b* colour of regions r_1 and r_2 is used as the basis for the merging criterion. In addition, the multi-scale edge strength $\mathrm{e}(r_1, r_2)$ between the regions is also taken into account. The final merging criterion C is weighted with a function s of the sizes $|r_i|$ of the to-be-merged regions r_i:

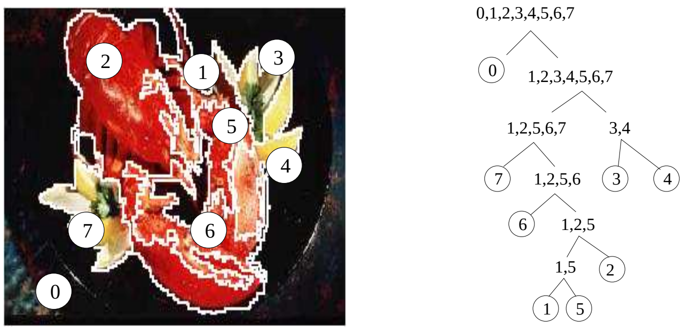
$$\mathrm{C}(r_1, r_2) = \mathrm{s}\,(r_1, r_2)\,\big(d_{Lab}\,(r_1, r_2) + \gamma \mathrm{e}\,(r_1, r_2)\,\big), \tag{1}$$

where

$$\mathrm{s}(r_1, r_2) = \min(|r_1|/|I|, |r_2|/|I|, a) + b \tag{2}$$

is the size-weighting function, $|I|$ is the number of pixels in the image and γ, a and b are parameters of the method. The values for the parameters have been selected to give visually feasible results for photographs and other images in earlier applications. The same values ($\gamma = 40$, $a = 0.02$, $b = 0.002$) have been used also in the current experiments.

The merging is continued until the desired number of regions are left. In addition to these *leaf segments*, we also record the hierarchical segmentation that results from running the region-merging algorithm on the leaf segments until only one region remains. Such *composite segments* are considered in our last experiments alongside with the leaf segments. Figure 3a shows an example of a segmented image and Figure 3b the corresponding segmentation hierarchy.



(a) (b)

Fig. 3. Example of a segmented image. Subfigure (a) displays the eight leaf segments found by the segmentation algorithm. Subfigure (b) shows the segmentation hierarchy resulting from the continued region merging. Leaf segments are circled in the tree.

6.2 Statistical Image Features

The PicSOM system implements a number of methods for extracting different statistical visual features from images and image segments. These features include a set of MPEG-7 content descriptors [22, 29] and additionally some non-standard descriptors for colour, shape and texture.

Colour

Of the used MPEG-7 descriptors Color Layout, Dominant Color and Scalable Color describe the colour content in image segments. In addition to the MPEG-7 colour descriptors, both the average colour in the CIE L*a*b* colour space [9] and three first central moments of the colour distribution are used as colour features.

Shape

Besides the MPEG-7 Region Shape, the shape features include two non-standard descriptors. The first consists of the set of the Fourier descriptors for the region contour [2]. Fourier descriptors are derived from the following expansion of the region contour:

$$z(s) = \sum_{n=-\infty}^{\infty} z_n e^{\frac{2\pi ins}{L}}. \tag{3}$$

Here the Cartesian coordinates of the contour are represented by the real and the imaginary parts of the complex function $z(s)$, parametrized by the arc length s. The resulting feature vector includes a fixed number of low-order expansion coefficients z_n. The coefficients are then normalised against affine image transformations. In addition, the high-order coefficients are quadratically emphasized.

The second non-standard shape descriptor is formed from the Zernike moments [25] of the region shape. The Zernike polynomials are a set of polar polynomials that are orthogonal in the unit disk. The Zernike moments A_{nm} are given by the expansion coefficients when the polar representation of the region shape is represented in the basis of Zernike polynomials:

$$A_{nm} = \frac{n+1}{\pi} \sum_x \sum_y I(x,y) V_{nm}\left(\rho(x,y), \theta(x,y)\right), \quad n - |m| \text{ even}. \tag{4}$$

Here n is the order of the moment, m the index of repetition, i, j are the rectangular image coordinates, and ρ, θ the corresponding polar coordinates. $I(x,y)$ is the binary representation of the region shape and V_{nm} is the Zernike polynomial:

$$V_{nm}(\rho,\theta) = R_{nm}(\rho) e^{im\theta} \tag{5}$$

$$R_{nm}(\rho) = \sum_{s=0}^{\frac{n-|m|}{2}} \frac{(-1)^s (n-s)!}{s!(\frac{n+|m|}{2} - s)!(\frac{n-|m|}{2} - s)!} \rho^{n-2s}. \tag{6}$$

The feature vector includes coefficients A_{nm} up to order a selected order. The feature is normalised against translation and scaling by fitting the region inside the unit disk. Rotation invariance is achieved by taking the absolute values of the coefficients.

Texture

We have used MPEG-7's Edge Histogram descriptor to describe the statistical texture in image segments. For non-standard description of a region's texture the YIQ colour space Y-values of the region pixels are compared with the values of their 8-neighbours. The feature vector describes the statistics of the resulting distribution.

6.3 Quantizing the Features with Self-Organising Maps

The visual features are disjointly partitioned into feature spaces, each feature space corresponding to the components of one visual descriptor. For instance, the components of MPEG-7 Color Layout descriptor form one feature space. The feature spaces are quantized using a variant of Self-Organising Map algorithm. The SOM is an unsupervised, self-organising neural algorithm widely used to visualise and interpret large high-dimensional data sets. The SOM defines an elastic net of points that are fitted to the distribution of the data in the input space.

The SOM consists of a two-dimensional lattice of neurons or map units. A model vector $\mathbf{m}_i \in \mathbb{R}^d$ is associated with each map unit i. The map attempts to represent all the available observations $\mathbf{x} \in \mathbb{R}^d$ with optimal accuracy by using the map units as a restricted set of models. During the training phase, the models become ordered on the grid so that similar models are close to and dissimilar models far from each other.

When training a SOM, the fitting of the model vectors is carried out by a sequential regression process, where $t = 0, 1, 2, \ldots, t_{max} - 1$ is the step index: For each input sample $\mathbf{x}(t)$, first the index $c(\mathbf{x})$ of the best-matching unit (BMU) or the "winner" model $\mathbf{m}_{c(\mathbf{x})}(t)$ is identified by the condition

$$\forall i: \quad \|\mathbf{x}(t) - \mathbf{m}_{c(\mathbf{x})}(t)\| \leq \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \ . \tag{7}$$

The distance metric used here is usually the Euclidean one. After finding the BMU, the vectors of map units constituting a *neighbourhood* centered around the node $c(\mathbf{x})$ are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h(t; c(\mathbf{x}), i)(\mathbf{x}(t) - \mathbf{m}_i(t)) \ . \tag{8}$$

Here $h(t; c(\mathbf{x}), i)$ is the neighbourhood function, a decreasing function of the distance between the ith and $c(\mathbf{x})$th nodes on the map grid. This regression is reiterated over the available samples and the value of $h(t; c(\mathbf{x}), i)$ is let decrease in time to guarantee convergence of the model vectors $\mathbf{m}_i$. Large values of the neighbourhood function $h(t; c(\mathbf{x}), i)$ are used in the beginning of the training for initialising the network, and small values on later iterations are needed for fine-tuning. After the training, any vector in the feature space can be quantized to a two-dimensional index by its BMU on the SOM.

6.4 Implementation of Statistical Correlation Inference

The implemented system performs statistical correlation separately in each of the feature spaces. For an image segment, this results in a *relevance score* for each of the feature spaces, as will be described below. A combined relevance score is then formed by summing the scores of all the feature spaces.

For the computational implementation of the statistical correlation principle, all the positive image segments are projected to all the feature SOMs. For each unit the number of segments projected to that particular unit is counted. The counts form a sparse value field on the SOM surfaces. Due to the SOM's property of mapping similar objects in nearby map units, we are motivated to spatially spread these sparse values by a low-pass filter, i.e. to convolve them with a smoothing kernel. The size and shape of the convolution kernel is selected in a suitable way in order to produce a smooth value map. In the resulting map each location is assigned a relevance value according to the number of positive objects mapped to the nearby units. This process of obtaining smooth relevance maps can be seen as nonparametric density estimation of the class of positive images in the latent spaces of the SOM grids.

After forming the positive relevance map for each SOM surface, the same procedure is repeated with the negative examples. These negative examples are obtained from the auxiliary or background images. Then the estimates of the positive $P_i^+(x,y)$ and negative $P_i^-(x,y)$ densities in SOM coordinates (x,y) of the feature SOM i are combined by map-wise weighting and subtraction:

$$P_i(x,y) = P_i^+(x,y) - \lambda P_i^-(x,y) \; . \tag{9}$$

Here λ is a free parameter of the method. The appropriate value of the parameter seems to be a function of the sizes of the positive and negative example image sets.

At this point each of the SOMs has a relevance map associated with it. For each image segment, a final relevance score is then calculated by summing the relevance scores of the segment's BMUs on all the feature SOMs. Those features that seem to distinguish well between positive and negative example images are in this process implicitly weighted more than the others. This is because the good distinction ability results in dense clusters of example images on the corresponding SOMs and high concentrations of relevance in the kernel estimation. If the images are scattered evenly on the map, the local peaks of relevance are much smaller in amplitude.

6.5 Propagating Relevance within Segment Hierarchy

To augment the implementation of the statistical correlation principle, we implement a mechanism for propagating relevance scores along the segmentation hierarchy within a single image. The propagation takes place after the relevance of individual segments has been evaluated by the SOMs. The statistical correlation is only able to indicate whether an individual segment or segment combination is relevant to the keyword. In contrast, the propagation algorithm simultaneously considers the relevance of several of the segments and their combinations that appear in the hierarchical segmentation of an image. By explicitly using the hierarchical relationship of the segments, the propagation algorithm is able to identify the largest combination of segments that is likely to correspond to the keyword.

Further motivation for the propagation step stems from the keyword focusing setting where we project both the leaf and the composite image segments, contained in the beforehand formed segmentation hierarchy (see Figure 3), to the same feature SOMs. We would then let the relevance scoring select the most appropriate scale of segmentation among the alternatives. Many of the useful low-level features are averaging, by which we mean that e.g. the average colour of a segment formed by combining two blue segments is still blue. We would like the relevance scoring mechanism to favour the merging of segments to a composite segment if they are similar in the sense of an averaging feature. However, some features are not averaging, e.g. the shape features of a composite region are not weighted averages of the constituent leaf regions. The composite shape will in some cases be characteristic to the positive example images while the shapes of the constituent regions are not. The scoring mechanism should favour also this type of composite segments for non-averaging features.

We would want the scoring mechanism not to use any *a priori* information whether the individual features are averaging or not. We require that the scoring method favours combining of segments that are relevant and whose combination is also relevant. On the other hand, also such combined segments should be favoured whose children are not relevant, but the combined segments themselves are. Considering the two requirements, a non-linear score propagation mechanism is needed for post-processing the relevance scores of the segments in the hierarchy after the SOM-based relevance assessments.

To this end, the proposed system implements a simple multiplicative model for propagating the relevance score p_i of segment i upwards in the segmentation hierarchy like the one seen in Figure 3. The propagation is initialised by setting $p_i = r'_i$ for the leaf segments. Here r'_i denotes the score obtained by normalising the relevance scores r_i within the image I_i containing the segment i:

$$r'_i = \frac{r_i - \bar{r}_i}{\max_{I_j = I_i} r_j - \bar{r}_i} \tag{10}$$

$$\bar{r}_i = \frac{1}{|\{j | I_j = I_i\}|} \sum_{I_j = I_i} r_j \tag{11}$$

For composite segments, the propagation rule is given by

$$p_i = \begin{cases} +r'_i(r'_i + \sum_{j \in \text{children}(i)} p_j^+) & \text{, if } r'_i > 0 \\ -r'_i(r'_i + \sum_{j \in \text{children}(i)} p_j^-) & \text{, otherwise.} \end{cases} \tag{12}$$

The set children(i) refers to all segments j to whom i is the immediate parent. Furthermore,

$$p_i^+ = \max(p_i, 0)$$
$$p_i^- = \min(p_i, 0).$$

7 Experiments

We have tested the feasibility of our approach to keyword focusing with two image databases. As the input we use, along with the images of the databases, the knowledge about each image being either annotated or not annotated with a specific keyword. The method is unsupervised: it does not learn to reproduce the image-level training labels but extracts qualitatively different labelling—labels for the image segments. The method's performance is evaluated by comparing the data-driven segment labels with manually defined ground truth.

In these experiments, the parameters of the proposed method, most importantly the used visual features, have been selected to optimise the performance in the test tasks. For a quantitative study, this would not be an adequate procedure for parameter selection. Some other means—such as optimisation with an independent data set—should be used. However, the used procedure serves to demonstrate the viability of our approach if the parameters are chosen appropriately.

The rest of the section is organised as follows. In Section 7.1 the methods for performance evaluation are described. Section 7.2 describes the two databases used. In Section 7.3 we gain insight to the inner workings of the proposed method by looking the SOM surface distributions corresponding to quantized representations of different feature spaces. Section 7.4 presents more quantitative results as the method's outputs are compared to ground truth via Receiver Operating Characteristic (ROC) curves.

7.1 Performance Evaluation

To evaluate the system's performance in the focusing tasks, we have manually defined a ground truth against which the system's performance is compared. To this end, we have first applied the automatic segmentation algorithm for the images. Then we have manually annotated the segments in those images which have been annotated with the studied keyword. In this annotation, we have marked which of the segments cover the object the keyword refers to. This annotation has been used only as a ground truth for the performance evaluation.

In the last experiments we also evaluate the system's ability to select the appropriate level in the segmentation hierarchy among all the alternatives produced by the hierarchical segmentation algorithm. To this end we define another set of segment-level ground truth annotations—the *best* segment. The best segment is defined to be the segment combination contained in the automatically produced hierarchical segmentation that best resembles the annotating keyword. Of course, choosing the best alternative is usually somewhat subjective, especially as the results of the hierarchical segmentation are often quite far from ideal. However, the task of selecting the best segment defined in this manner still captures an important aspect of the performance evaluation problem.

After having defined the ground truth classes for performance evaluation, we measure the system's performance with receiver operating characteristic curves. For the shown figures, we have first generated separate ROC curves for each database image annotated with the particular keyword. The curves have then been averaged by considering the true positive rate to be a function of the false positive rate. Thus, the curves are averaged graphically along the vertical direction. Generating a separate curve for each image is motivated by the fact that here we consider the task of focusing the keyword inside individual images. The ordering of the segments in different images is not regarded as relevant. The averaging procedure may not correspond to any actual classification scenario, but is adequate to get a measure of average performance for different images.

For the linear combination coefficient λ in Eq. (9) the value $\lambda = 1/30$ was found to produce good results in the current experiments. This means that the effect of an individual negative example segment is considerably smaller than the effect of a positive segment. However, since the negative example images were much more numerous, the resulting weight of the distribution of negative segments is much larger than that of the positive examples.

7.2 Databases

The first of the two databases has been used to provide an easily understandable lightweight testbed for the framework. For this purpose, we have selected a 900 image subset of the commercial Corel database [10]. The images depict people, most of them models. We thus call this database the *models database* from here on. As the keyword to be focused we chose the word "red". We have manually annotated the database for this keyword by defining the keyword for an image if it contained a salient red object captured by the automatic segmentation algorithm. 107 images of the database were judged to portray red objects. The chosen keyword has straightforward connections to both image segmentation and the visual features, and we thus hope the results to be straightforward to interpret. This way we could focus our attention solely to the keyword focusing mechanism, not the image segmentation. As the visual features for this database we used one colour feature, colour moments, and one local texture feature, the MPEG-7 Edge Histogram (cf. Section 6.2). We expect to observe a different behaviour of these two features in the keyword focusing task as colour is directly related to the target keyword whereas the edge histogram is not.

The second database we used is the *101 Object Categories* database [17] of the PASCAL *Visual Object Classes Challenge*[2]. The database contains 9197 images divided into 101 semantic categories, each containing between 31 and 800 images, and a background or auxiliary class of 520 miscellaneous images. The database has been created mostly for object recognition purposes and therefore does not contain detailed image-wise annotations. For the experiments, we chose one of the categories, "lobster", as the keyword to be focused. A lobster is portrayed in 41 of the database images. The keyword does not have a direct connection to the image segmentation algorithm or to any specific feature representations as in the case of the first database. For this database the set of visual features is selected by considering the concerted ROC curves for the focusing tasks. In these curves all the segments of the images corresponding to the keyword are ordered in a common list according to their relevance scores. Features are added to the set until further addition of features no longer improves the equal error rate (EER) of the ROC curve.

7.3 Feature Distributions on SOM Surfaces

Models Database

Figure 4 displays how the feature distributions of the example segments become projected on the feature SOM surfaces in the case of the models database. The map surfaces have toroidal topology, i.e. the top edges are connected to the bottom edges and the left edges to the right edges. This way the visual analysis is more straightforward as we do not have to consider the edge effects that can be significant in rectangular SOMs. The actual focusing performance of the both topologies is according to our observations approximately the same. Distribution of the colour feature is shown in the left column and that of the edge histogram feature in the right column. The input densities to the algorithm are densities on the rows (a) and (b). The row (c) is the outcome of the algorithm, and row (d) can be seen as the desired outcome. However, note that outside the feature space regions where there are segments on row (a), the final qualification values on row (c) have no significance to the outcome of the keyword focusing task.

As expected, the colour feature captures well the keyword "red", as indicated by the dense concentration of the positive example segments to specific map surface regions. The segments are more widely spread on the edge histogram map surface. Furthermore, by comparing the distributions of true "red" segments and all the keyword segments, we note that the distributions peak at approximately same locations corresponding to the truly "red" segments. This happens even though the majority of the keyword segments are false positives, i.e. they are not "red". This is explained by the fact that the non-red segments in the example images are distributed throughout the colour

[2] http://www.pascal-network.org/challenges/VOC/

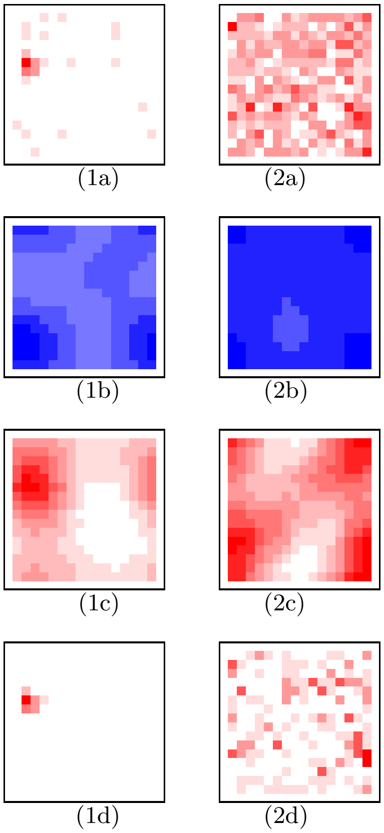


Fig. 4. Distributions of models database segments on two feature SOM surfaces. The left column (1) shows the distribution of colour feature, the right column (2) the distribution of MPEG-7 Edge Histogram feature. Dark colour corresponds to high density. Note that the distributions are normalised so that the maximum value always corresponds to black colour. Therefore the shades in different subimages are not comparable, only the shapes of the distributions. The top row (a) displays the distribution of all the segments in the images that are annotated with the keyword “red”. The second row (b) shows the distribution of all the segments in the models database. The third row (c) shows the linear combination of the first and the second row according to Eq. (9) which is used as the final qualification value of the segments with respect to that feature. The relevance is spread on the second and third row by the convolution mechanism discussed in Section 6.4. The fourth row (d) shows the distribution of manually confirmed “red” segments (true positives).

feature space. Therefore, in any specific region of the feature space their concentration is still low and easily dominated by the locally peaking distribution of the true positives.

101 Object Categories

Figures 5 and 6 display the projections of the segments on the feature SOM surfaces in a manner similar to Figure 4. In addition, however, the figures include the distribution of *best* segments (cf. Section 7.1) on row (e) as we also consider the problem of finding the most representative combined segment among the segments in the hierarchical segmentation.

From the figures it can be seen that, in addition to the concentration of positive example segments, the application of Eq. (9) and subtraction of the background density due to the auxiliary images (row (b)) can be essential to the focusing performance. For instance, by comparing the distributions on the first and last rows of the Fourier feature (column (3) of Fig. 5), we notice areas of false positives in the lower part of the first-row map near the left and right edges. Successful keyword focusing requires suppression of these regions and the subtraction of the background relevance seems to offer an effective means for achieving this.

On the other hand, by comparing rows (a) and (d) of the texture feature (column (4) of Fig. 5), we see that keyword focusing would require suppression of the lower right quadrant of the first-row SOM surface. However, the background density is quite low in that region and therefore the region gets amplified in application of Eq. (9). This leads one to expect the texture feature to perform poorly in keyword focusing, which is indeed confirmed to be the case.

It can be seen that the shape features (columns (3) and (5) in Fig. 5 and columns (3) and (4) in Fig. 6) are promising candidates for favouring the best segments over other “lobster” segments. The large values of background relevance coincide with some regions that are more pronounced in the distribution of all “lobster” segments (row (d)) than in the distribution of the best “lobster” segments (row (e)).

7.4 Performance in Keyword Focusing

Models Database

Figure 7 shows some examples of image segmentation and Figure 8 the ROC curve, when the keyword focusing experiment applied onto keyword “red” in the models database. In Figure 7c the focusing algorithm erroneously considers the segment 4 to be more “red” than the segment 5. This can be explained by the unusual shades of red and some dark areas in segment 5.

The almost ideal ROC curve of Figure 8 indicates the performance of the system to be very satisfactory in general, with some rare exceptions. This is also confirmed by manual inspection of the focusing results of the individual images. We can thus confirm that when the feature spaces, image segmentation and the studied keyword are compatible, the statistical correlation method is an effective means for keyword focusing.

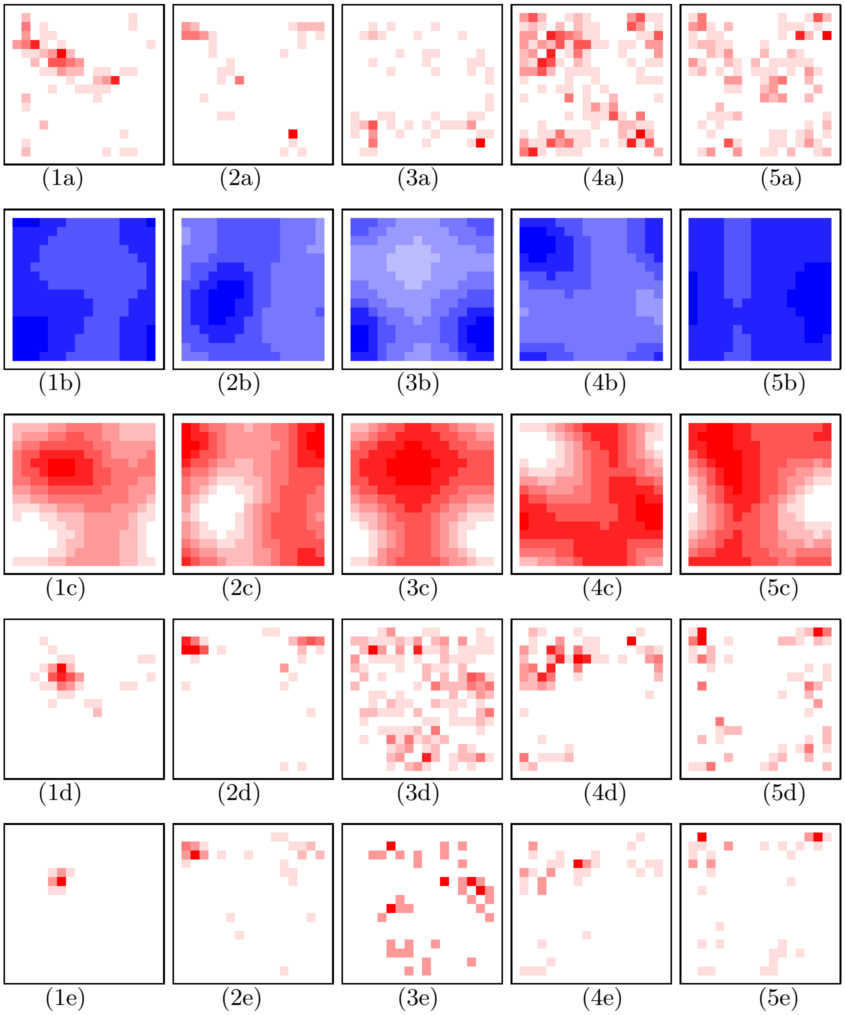


Fig. 5. Projections of segments associated with keyword "lobster" on 101 Object Categories database to the feature SOM surfaces of the non-standard feature spaces. Columns: (1) colour moments, (2) average colour, (3) Fourier shape, (4) texture and (5) Zernike moments. Rows: (a) all segments of images annotated with keyword lobster, (b) all segments in the database, (c) the combination of (a) and (b) according to Eq. (9), (d) true "lobster" segments, (e) best "lobster" segments.

101 Object Categories

The system's keyword focusing performance with keyword "lobster" in the 101 Object Categories was evaluated separately in two tasks: (1) identifying any lobster segments in the segmentation hierarchy, and (2) selecting the single best segment combination from the hierarchy. Both of these tasks were

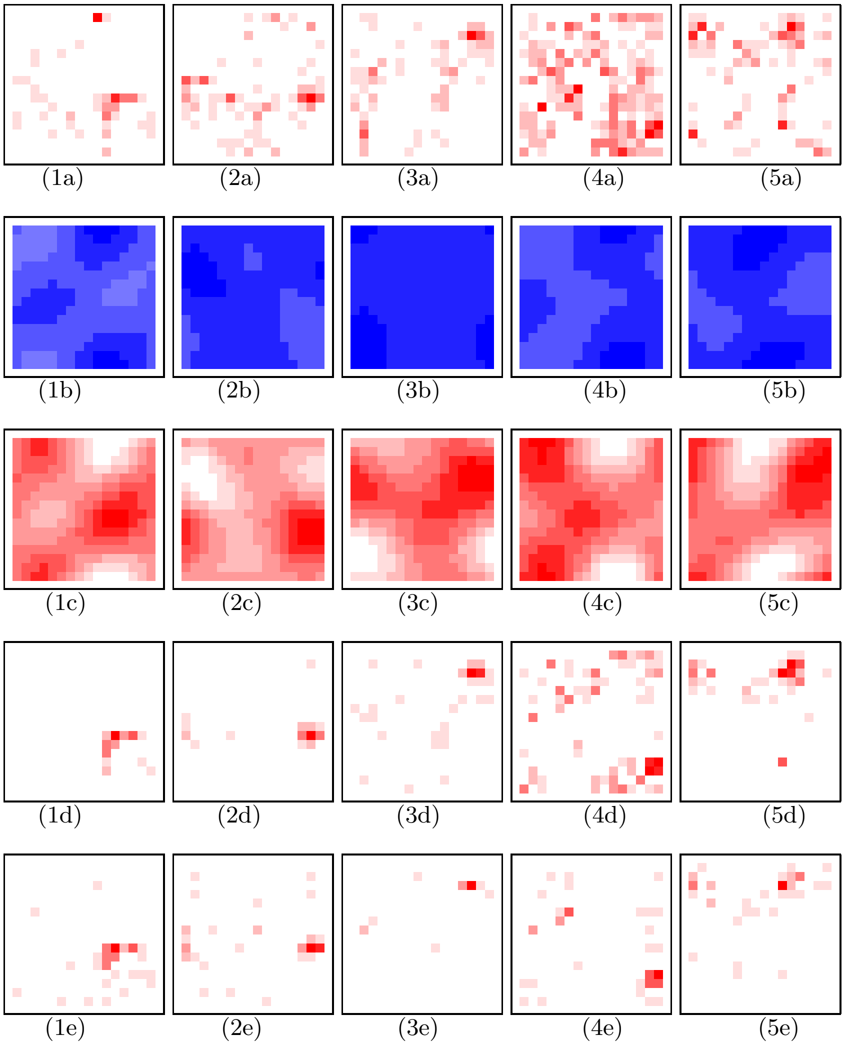


Fig. 6. Projections of segments associated with keyword "lobster" on 101 Object Categories database to the feature SOM surfaces of MPEG-7 feature spaces. Columns: (1) Color Layout, (2) Dominant Color, (3) Edge Histogram, (4) Region Shape and (5) Scalable Color. Rows: (a) all segments of images annotated with keyword lobster, (b) all segments in the database, (c) the combination of (a) and (b) according to Eq. (9), (d) true "lobster" segments, (e) best "lobster" segments.

performed with and without the intra-image relevance propagation mechanism (cf. Section 6.5). This gives four variants of the problem altogether.

For each problem variant we state the results of using the set of features that was found to perform best. The optimal feature sets were found to be

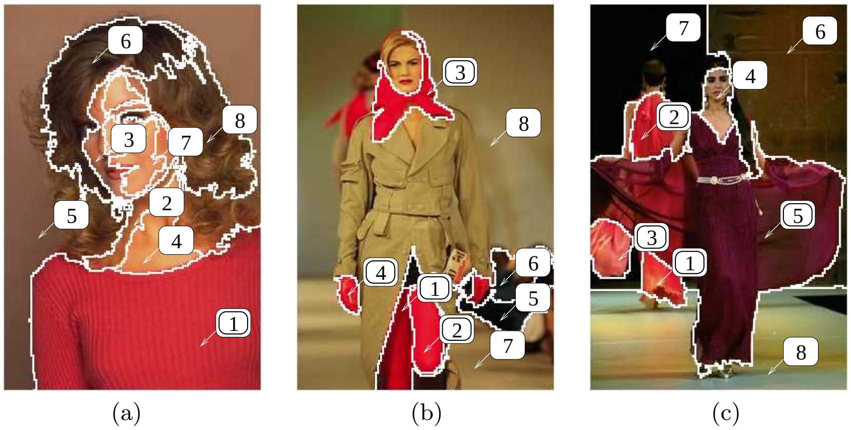


Fig. 7. Examples of focusing the keyword "red" in the models database. The white borders in the image indicate the eight regions found by the segmentation algorithm. The number tags reflect the ordering of the segments produced by the focusing algorithm. The tags of truly red segments (defined for evaluation purposes only) are shown with double border.

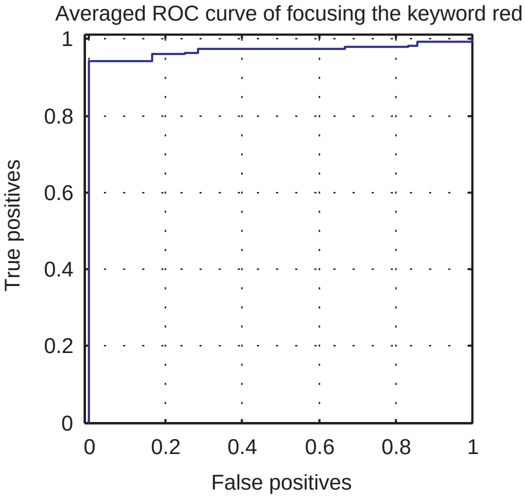


Fig. 8. The averaged ROC curve of focusing keyword "red" in the models database.

different for each problem variant, although the performance was not strongly dependent on the choice of features. The robustness can be partially attributed to the PicSOM system's ability to automatically emphasise the most useful features. Table 1 shows the optimal feature sets in the ROC EER sense, listed in order of decreasing significance.

Table 1. The feature sets optimised for variants of the focusing task

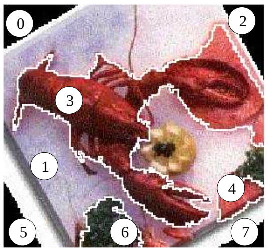
	No propagation	With propagation
Any segment	colour moments Fourier shape average colour MPEG-7 Edge Histogram MPEG-7 Color Layout	colour moments average colour Fourier shape MPEG-7 Dominant Color MPEG-7 Color Layout MPEG-7 Scalable Color MPEG-7 Edge Histogram
Best segment	colour moments MPEG-7 Edge Histogram Zernike Moments MPEG-7 Scalable Color	average colour MPEG-7 Edge Histogram Zernike moments

Figure 9 provides some example cases of keyword focusing. In general, the performance of the system in this task is satisfactory, although there are cases where the system does not function as well as desired. In many cases of failure, the reason can be tracked down to the unsatisfactory segmentation of images. The lowermost row (c) of Figure 9 exemplifies such a situation. The white background and kitchen tool cause the lobster to divide into two parts and the segmentation algorithm does not even consider the merging of these regions.

Comparison of columns (2) and (3) in Figure 9 shows the effect of the relevance propagation algorithm. On the rows (a) and (c) the typicality in the parallel feature spaces alone has been enough to capture the proper ordering of the "lobster" segments (marked with a +), even placing the *best* segments (marked with a *) on the top of the lists. On the row (b), however, the relevance propagation step is required for the correct re-ordering of the list.

Figure 10 shows the ROC curves for three cases of the keyword focusing task. The subfigures (a) and (b) correspond to the four variants identified in the Table 1. It can be noted that the propagation of relevance along the segmentation hierarchy improves performance in finding the single best segment in (b), but does not significantly affect the performance in the task of finding any lobster segments in (a). This was to be expected, as the rationale for the relevance propagation is to re-order the segments that were found to be relevant so that the most representative segment combinations are favoured.

Figure 10c shows the algorithm's performance in finding the best segment (* in Fig. 9) among the true lobster segments (+ in Fig. 9). This way the the effect of the algorithm's performance in finding any lobster segment among all the segments is excluded. Figure 10c can thus be regarded as a residual performance that remains when the effect of the good performance in the easier subtask (a) is eliminated from the task of subfigure (b). In Figure 10c the relative ordering of the algorithms with and without relevance propagation is similar to that in subfigure (b). This happens because the performance in



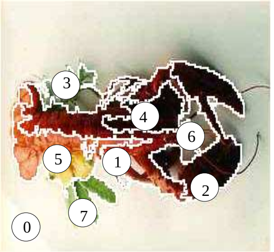
(1a)

* 3+
0,1,2,3,4,5,6,7
0,1,2,3,4,6,7
1,3,4,6,7
1,2,3,4,6,7
1,3,4,6
4
0
7
5
2
1
1,4
1,4,6
6

(2a)

* 3+
1,3,4,6
0,1,2,3,4,5,6,7
0,1,2,3,4,6,7
1,3,4,6,7
1,2,3,4,6,7
4
0
7
5
2
1
1,4
6
1,4,6

(3a)



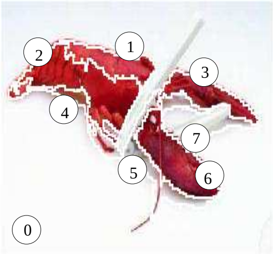
(1b)

1
1,4+
* 1,2,4+
1,2,4,6+
7
1,2,4,5,6,7
2+
5,7
5+
4+
6+
3
0,1,2,3,4,5,6,7
0,3
0

(2b)

* 1,2,4+
1,2,4,6+
1,4+
1,2,4,5,6,7
1
7
5,7
2+
5+
4+
6
3
0
0,1,2,3,4,5,6,7
0,3

(3b)



(1c)

* 1,2,4 +
2,4 +
3 +
6 +
2 +
1 +
4 +
0,3,5,6,7
0,3,5,7
0,5,7
0,1,2,3,4,5,6,7
5
0,7
0
7

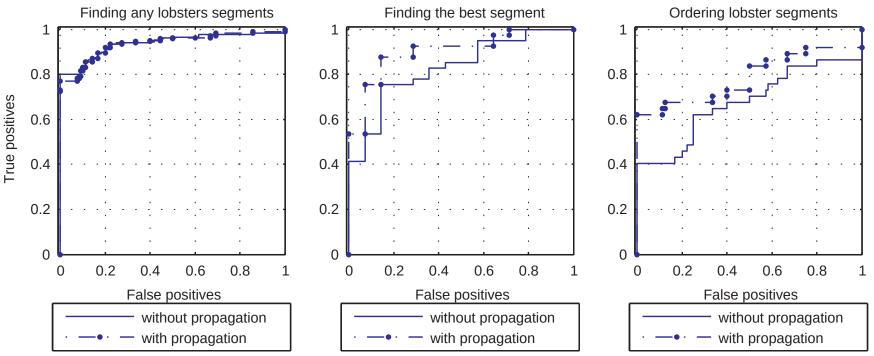
(2c)

* 1,2,4+
2,4+
3+
6+
2+
1+
4+
0,3,5,6,7
5
0,1,2,3,4,5,6,7
0
7
0,3,5,7
0,7
0,5,7

(3c)

Fig. 9. Examples of focusing the keyword "lobster" in the 101 Object Categories database. The white borders in the images in column (1) indicate the eight regions found by the segmentation algorithm. The numbers in the tags are arbitrarily chosen segment labels. Columns (2) and (3) list the ordering of the segments output by the focusing algorithm. Column (2) shows the algorithm results without relevance propagation along the segmentation hierarchy. In column (3) the propagation is included. The segments more likely to be associated with the keyword "lobster" are on the top of the lists. In the lists segments marked with + have been manually judged to consist of mostly lobsters. The asterisk (*) beside a segment label indicates that the segment has been manually judged as the best segment, i.e. the segment most representative of the keyword lobster. Note that we have considered only the alternatives generated by the hierarchical segmentation algorithm. Therefore, for instance, in the figure of row (b) the combined segment segment 1,2,4 is chosen as the best segment as the combination 1,2,4,5 is not offered as an alternative by the segmentation stage.

finding any lobster segment is practically the same for the two algorithm alternatives, as shown by subfigure (a). However, from the absolute magnitude of the curves we see that also without relevance propagation the algorithm performs considerably better than random selection. Thus the principle of typicality in the selected feature spaces partly manages to favour the appropriate composite segments over their constituent parts. Nonetheless, in a significant proportion of cases the ordering is improved by augmenting the typicality assessment with the relevance propagation step.



(a) (b) (c)

Fig. 10. The averaged ROC curves of focusing keyword “lobster” in the 101 Object Categories database. Solid lines correspond to the focusing algorithm without the relevance propagation along segmentation hierarchy, the dashed line with the propagation. Subfigure (a) measures the focusing accuracy of finding any “lobster” segments. Subfigure (b) measures the accuracy of pinpointing the segment combination that is manually judged to be the best. Subfigure (c) measures in which order the true lobster segments are found. Images with only one lobster segment are excluded from this subfigure.

8 Conclusions and Future Views

In the light of the experiments, it is evident that the proposed statistical correlation principle offers a viable approach to the keyword focusing problem. However, it is clear that to function as a part of a real-world application, this technique should be augmented with other learning principles in order to produce keyword focusings that utilise the information contained in the data more efficiently. This is kind of self-evident: *a priori* information makes the learning problem easier. A lower-level, more general learning principle is necessarily more laborious in the case where the *a priori* assumptions hold.

The presented experiments demonstrate the potential of a system architecture, where image data is first pre-processed by a feedforward type region segmentation and description front end. The inference algorithms are subsequently applied to the representations generated by the front end. Parallel Self-Organising Maps provide a feasible means for constructing such a front end. An analogy can be drawn between this and the cortical maps of the human visual system.

The straightforward image segmentation algorithm and low-level visual features in our current implementation are by no means optimal. The framework, however, is useful and we have tried to make it easy to improve, add or replace the individual components. There exists inevitably a limit to the performance of low-level feedforward image segmentation, which can be overcome only by interlinking the image segmentation with higher-level image understanding. A low-level preprocessing algorithm cannot be expected to connect parts of objects that are visually sufficiently dissimilar. The borderline where the feedforward front end should yield up the processing to the higher-level inference algorithms, in this case the focusing algorithm, is quite vague. It might be better to let the image segmentation interact more closely with the focusing procedure and not consider the segmentation as a part of the front end.

One could also include more information about the alternative segmentations in the image representation by using a more versatile data structure than a tree. The data structure could be equipped with the probabilities of region merges. On the other hand, the data structure could include just more alternative segmentations, resembling the data structures, e.g. lattices, used in automatic speech recognition.

All in all, in the feasibility studies of this chapter we have demonstrated that soft computing methods can be successfully used to explore the image–word correspondence and the emergence of semantical concepts. More rigorous experiments would have to be performed for more quantitative analysis. On the other hand, it has been elucidated that the subproblems of content-based image retrieval offer a challenging application area for soft computing techniques.

References

1. Shivani Agarwal, Aatif Awan, and Dan Roth. Learning to detect objects in images via a sparse, part-based representation. *IEEE Transactions on Pattern Analysis Analysis and Machine Intelligence*, 26(11):1475–1490, November 2004.
2. K. Arbter. Affine-invariant Fourier descriptors. In J. C. Simon, editor, *From Pixels to Features*, pages 153–164. Elsevier Science Publishers B.V.(North-Holland), 1989.
3. Kobus Barnard, Pinar Duygulu, Nando de Freitas, David Forsyth, David Blei, and Michael I. Jordan. Matching words and pictures. *Journal of Machine Learning Research, Special Issue on Machine Learning Methods for Text and Images*, 3:1107–1135, February 2003.

4. S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):509–522, April 2002.
5. I. Biederman. A theory of human image understanding. *Psychological Review*, 94:115–147, 1987.
6. Peter Carbonetto, Nando de Freitas, and Kobus Barnard. A statistical model for general contextual object recognition. In *Proceedings of the Eight European Conference on Computer Vision*, Prague, May 2004.
7. Chad Carson, Serge Belongie, Hayit Greenspan, and Jitendra Malik. Blobworld: Image segmentation using expectation-maximization and its application to image querying. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1026–1038, August 2002.
8. Yixin Chen and James Z. Wang. Looking beyond region boundaries: Region-based image retrieval using fuzzy feature matching. In *Multimedia Content-Based Indexing and Retrieval Workshop, September 24-25*, INRIA Rocquencourt, France, September 2001.
9. Supplement No. 2 to CIE publication No. 15 Colorimetry (E-1.3.1) 1971: Official recommendations on uniform color spaces, color-difference equations, and metric color terms, 1976.
10. The Corel Corporation WWW home page, `http://www.corel.com`, 1999.
11. Alexander Dimai. Unsupervised extraction of salient region-descriptors for content based image retrieval. In *10th International Conference on Image Analysis and Processing (ICIAP), September 27-29*, pages 686–691, Venice, Italy, September 1999.
12. J. P. Eakins. Automatic image retrieval — are we getting anywhere? In *Third International Conference on Electronic Libraries and Visual Information Research (ELVIRA3), April 30 - May 2*, pages 123–135, Milton Keynes, UK, 1996. De Montfort University.
13. Mark Everingham, Andrew Zisserman, and Christopher K. I. Williams et al. The 2005 PASCAL Visual Object Classes Challenge. In F. d'Alche Buc, I. Dagan, and J. Quinonero, editors, *Selected Proceedings of the first PASCAL Challenges Workshop*. Springer, 2006.
14. The Fine Arts Museum of San Francisco `http://www.thinker.org`, 2005.
15. Jianping Fan, Yuli Gao, and Hangzai Luo. Multi-level annotation of natural scenes using dominant image components and semantic concepts. In *Proceedings of the 12th annual ACM international conference on Multimedia*, pages 540–547, New York, NY, October 2004.
16. Jianping Fan, Yuli Gao, Hangzai Luo, and GuangYou Xu. Automatic image annotation by using concept-sensitive salient objects for image content representation. In *Proceedings of the 27th annual international conference on Research and development in information retrieval*, pages 361–368, Sheffield, England, July 2004.
17. Li Fei-Fei, Rob Fergus, and Pietro Perona. Learning generative visual models from few training examples: An incremental Bayesian approach tested on 101 object categories. In *Proceedings of the Workshop on Generative-Model Based Vision*, Washington, DC, June 2004.
18. W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(9):891–906, September 1991.

19. Hervé Glotin and Sabrina Tollari. Fast image auto-annotation with visual vector approximation clusters. In *Proc. of IEEE EURASIP Fourth International Workshop on Content-Based Multimedia Indexing (CBMI2005)*, June 2005.
20. L. Guan, P. Muneesawang, J. Lay, I. Lee, and T. Amin. Recent advancement in indexing and retrieval of visual documents. In *Proceedings of the Ninth International Conference on Distributed Multimedia Systems / The 2003 Conference on Visual Information Systems (VIS'2003)*, pages 375–380, Miami, FL, USA, September 2003.
21. V. N. Gudivada and V. V. Raghavan. Content-based image retrieval systems. *IEEE Computer*, 28(9):18–22, 1995.
22. ISO/IEC. Information technology - Multimedia content description interface - Part 3: Visual, 2002. 15938-3:2002(E).
23. J. Jeon, V. Lavrenko, and R. Manmatha. Automatic image annotation and retrieval using cross-media relevance models. In *Proceedings of 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 119–126, Toronto, Canada, July-August 2003.
24. F. Jing, M. Li, L. Zhang, H. Zhang, and B. Zhang. Learning in region-based image retrieval. In *Proceedings of International Conference on Image and Video Retrieval*, volume 2728 of *Lecture Notes in Computer Science*, pages 198–207. Springer, 2003.
25. A. Khotanzad and Y. H. Hong. Invariant image recognition by Zernike moments. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 12(5):489–497, 1990.
26. Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, third edition, 2001.
27. 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, December 2000.
28. Jorma Laaksonen, Markus Koskela, Sami Laakso, and Erkki Oja. Self-organizing maps as a relevance feedback technique in content-based image retrieval. *Pattern Analysis & Applications*, 4(2+3):140–152, June 2001.
29. Jorma Laaksonen, Markus Koskela, and Erkki Oja. PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. *IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing*, 13(4):841–853, July 2002.
30. Jia Li and James Z. Wang. Automatic linguistic indexing of pictures by a statistical modeling approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(9):1075–1088, September 2003.
31. Nikos K. Logothetis and David L. Sheinberg. Visual object recognition. *Annual Review of Neuroscience*, 19:577–621, 1996.
32. David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.
33. Krystian Mikolajczyk and Cornelia Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(10):1615–1630, October 2005.
34. Anuj Mohan, Constantine Papageorgiou, and Tomaso Poggio. Example based object detection in images by components. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(4):349–361, April 2001.

35. Florent Monay and Daniel Gatica-Perez. On image auto-annotation with latent space models. In *Proceedings of the eleventh ACM international conference on Multimedia*, pages 275–278, Berkeley, CA, 2003.
36. Yasuhide Mori, Hironobu Takahashi, and Ryuichi Oka. Image-to-word transformation based on dividing and vector quantizing images with words. In *Proceedings of First International Workshop on Multimedia Intelligent Storage and Retrieval Management*, 1999.
37. Jia-Yu Pan, Hyung-Jeong Yang, Pinar Duygulu, and Christos Faloutsos. Automatic image captioning. In *Proceedings of the 2004 IEEE International Conference on Multimedia and Expo*, Taipei, Taiwan, June 2004.
38. Jia-Yu Pan, Hyung-Jeong Yang, Christos Faloutsos, and Pinar Duygulu. GCap: Graph-based automatic image captioning. In *Proceedings MDDE '04, 4th International Workshop on Multimedia Data and Document Engineering*, Washington, DC, USA, July 2004.
39. G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. Computer Science Series. McGraw-Hill, 1983.
40. Robert J. Schalkoff. *Pattern Recognition: Statistical, Structural and Neural Approaches*. John Wiley & Sons, Ltd., 1992.
41. Arnold W. M. Smeulders, Marcel Worring, Simone Santini, Amarnath Gupta, and Ramesh Jain. Content-based image retrieval at the end of the early years. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(12):1349–1380, December 2000.
42. Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis and Machine Vision*. International Thomson Computer Press, 1993.
43. S. Ullman. *High-Level Vision: Object recognition and cognition*. MIT Press, 1996.
44. A. Ultsch. Data mining and knowledge discovery with emergent self-organizing feature maps for multivariate time series. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 33–45. Elsevier, 1999.
45. Ville Viitaniemi and Jorma Laaksonen. Keyword-detection approach to automatic image annotation. In *Proceedings of 2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT 2005)*, pages 15–22, London, UK, November 2005.
46. James Z. Wang, Jia Liu, and Gio Wiederhold. SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(9):947–963, September 2001.