CHAPTER 9

# EMERGENCE OF SEMANTICS FROM MULTIMEDIA DATABASES

**Erkki Oja, Mats Sjöberg, Ville Viitaniemi, and Jorma Laaksonen**

## I. INTRODUCTION

Information in its every form is becoming more and more important in the world of today. Modern computer systems can store huge amounts of data, and new data are acquired at an ever increasing rate. In a recent study [1] it was estimated that we collectively produced around 5 exabytes ($5\times10^{18}$ bytes) of new information in the year 2003! All this, coupled with the fact that we are becoming more and more dependent on finding relevant information quickly in our daily private and professional lives, calls for new and better information retrieval methods to be developed.

A typical situation in information retrieval might be searching a large database of multimedia content, like text, pictures, music or video. Such a database could for example be the World Wide Web or a subset of it. The traditional indexing approach of manually entering metadata, like textual descriptions and keywords associated with the stored data, quickly becomes infeasible when we have thousands or millions of data objects. Manually entering metadata is not only labor intensive but also error prone. Additionally, the descriptions provided by humans can vary from person to person, from language to language, and can depend on different interpretations and points of view.

The contemporary solution to the indexing problem is *content-based information retrieval* (CBIR), where the data objects are indexed by feature vectors automatically calculated from the data objects themselves. This allows us to search by the actual contents and we are no longer constrained by keywords and metadata descriptions. Automatic feature extraction is of course also much faster than manual indexing, and in some cases even the only possibility.

On the other hand, creating a good feature extraction algorithm is not a trivial task. A CBIR system cannot "understand" the data that it is indexing in the sense that humans can, and therefore the retrieval results can never be perfect as judged by a human being. This fundamental problem is sometimes called the *semantic gap*. This means that a CBIR system is most effective as a semi-automatic tool, with a human expert pointing it in the right direction. Used in the correct manner we can get a best-of-both-worlds solution where the computational system calculates and compares low-level features, and the human user provides the high-level abstract overview. In reality, the CBIR systems are far from perfect, and the human interaction with the system, trying to point it in the right direction, might be tedious and frustrating.

To improve existing CBIR systems we have looked at how humans do recognition and association. Humans do not only rely on immediate low-level data, but also benefit from *a priori* knowledge and an understanding of the context and relationships of the data

items. This observation is at the focal point of this chapter; because data objects in a database are seldom unrelated, it might be fruitful to use any existing object dependencies in an information retrieval system. Here we study such relations in two settings, first between segments of images and then between different modalities of video. In general, similar dependencies can be found in the contexts of words, images found on the same web page, links to other web pages, or an e-mail message containing attachments.

Our approach is to model dependency relationships as multi-part hierarchical objects and share the user-given relevance assessments between related objects. In this way, for example an image attachment of an e-mail message that has been deemed relevant (e.g., on the basis of its color properties) can also increase the relevance of the e-mail message that it is attached to. Thus objects that are related, but not similar in the view of low-level features, can still be found in the retrieval process.

The rest of the chapter is organized as follows. Section II first introduces the basic concepts for content-based retrieval of multimodal information and our PicSOM retrieval framework. Section III then presents our research on extracting semantic information from automatically segmented images and Section IV describes the application of hierarchical multimodal objects for content-based video retrieval. Conclusions and future views are presented in Section V.

## II. MULTIMODAL CONTENT-BASED INFORMATION RETRIEVAL

In content-based information retrieval the focus is on the database objects themselves and how we best can extract relevant and meaningful information from these. Notice that by CBIR we mean content-based *information* retrieval rather than the more common *image* retrieval. This is because we also consider objects of other types than images, for example text, video, and multimedia objects in general. Furthermore, when modeling dependency relationships in a CBIR database, the fusion of multimodal information becomes a crucial aspect of the retrieval system.

### The Semantic Gap

Three semantic levels of image queries, which can be applied more generally for any information query, can be identified [2]: (1) retrieval by *primitive* features, such as colors of an image or words or character combinations in a text, (2) retrieval by *logical* features or *semantic* attributes, which can be more high-level descriptions such as “a picture of a car” or “a medical text,” and (3) retrieval by *abstract* attributes such as “a picture depicting happiness” or “an ironic text.” This division into semantic levels also clearly illustrates the *semantic gap*, i.e., the large separation between the high-level semantic description used by humans and the low-level features used by computer systems [3]. Object descriptions in CBIR systems are mostly of semantic level 1, while humans naturally use levels 2 or 3 when describing their query target. For example, in the case of a digital image, the automatically extracted low-level features only “see” local phenomena, such as colors, patterns, and textures. A human analysis of the same image might be more holistic, describing for example the objects seen in the image and their relationships, not necessarily even noticing particular colors or textures.

The core of the problem lies in the difficulty of automatically extracting semantic content from a data object. The same semantic concept might have many totally different low-level representations, and also objects with similar low-level features might have distinct semantic meanings. This has also been pointed out in the domain of digital images [4]. Humans on the other hand are experts in recognizing semantic content. And from what we know, low-level information, such as specific words in a text or colors in an image, is only one part of the information used in the recognition process of humans. A

lot of *a priori* knowledge is involved, such as previous experience of similar situations, cultural context and so on. Also the current situation and the context is important. Such information is hard to incorporate into a computer system.

To achieve something analogous to our human understanding of information content we would need to incorporate other sources, such as the aforementioned *a priori* knowledge, into our CBIR systems. One way to introduce a certain amount of learning from previous similar query situations has previously been demonstrated in our experiments using the PicSOM CBIR system [5]. In that work, the system stores the relevance history of each object, i.e., how relevant the object has been in different past queries. This information is then used as a separate statistical feature in future queries, giving a measure of semantic similarity based on previous experience. But this is of course still a long way from the wide spectrum of information, cultural knowledge, and superb recognition and association skills that humans possess.

Another approach is to take advantage of known relationships and context of data items in the retrieval process. This often means using information coming from different modalities, like text, audio, or video content that have some relevance to the query target. This problem, sometimes called *multimodal information fusion*, is not trivial as there is no obvious general solution. The information from different modalities may often correlate in some manner which can be beneficial to the information retrieval, but in other cases the information from different sources may even contradict each other. A general multimodal information retrieval system needs to be able to handle all such situations and combine the information from different modalities in a useful way.

There are several ways to implement multimodal fusion in information retrieval, and the problem can be addressed on different levels. On the feature level, feature vectors from different modalities can simply be concatenated, essentially creating a new feature. This is the approach taken for example in ImageRover [6] where visual and textual features are combined into one unified vector. Another strategy is to process the different modalities separately and then merge the retrieval results in some manner in the end. Finally, the most promising technique is to implement cross-modality into the information retrieval process itself. This usually results in associating objects across different modalities. This cross-modality can for example be based on context that has been stored beforehand, for example, that a certain sound clip was recorded at the same time and place as a certain video shot was taken. Another way to implement cross-modality is by statistical correlation, using for example latent semantic indexing (LSI) or cross-modal factor analysis [7] and Bayesian network models [8]. Most of the existing multimodal information retrieval systems are highly specialized to work with specific types of media, for example audio and video, or even very specific domains, e.g., videos of sporting events.

**Multi-Part Hierarchical Objects and Relevance Sharing**

We have chosen to model relationships between objects in a database by grouping the objects involved into *multi-part objects*. A multi-part object is a collection of objects in the database and does not have any intrinsic properties other than those of the contained objects. We focus mostly on multi-part objects that can be represented in a hierarchical manner, and thus can be organized in an object tree in the database. In some situations an object in the tree can have many parents, and then the structure is technically no longer a tree but rather a graph. Still, we usually use the term “object tree.” Also, if we consider each multi-part object as being formed of one parent plus its children (on one or more levels), a graph can be considered as a tree that only happens to share some children with other object trees. These object trees or *hierarchical objects* can be

multimodal, i.e., consist of objects of different types, and the object tree can be of any depth.

Two examples of real-world multimedia objects are shown in Fig. 1. On the left side we have a typical media-rich e-mail message with image, audio, and video attachments, where the different parts have been highlighted and numbered. On the right there is an example of a web page with textual content, embedded images and links to other web pages. Examples of hierarchical object trees created from these examples are shown in the same figure, below the multimedia objects.

The properties of each object in the hierarchical tree, i.e., the calculated feature vectors (using one or many different feature extraction methods), can be considered to be characteristic not only of the object itself, but to some extent also of its parents, children, and siblings in the tree structure. We call this idea *relevance sharing*, which means that the relevance will be transferred from the object to its parents, children, and siblings. This means, for example, that if a certain e-mail message is considered relevant by the user in a query, its attachments would also get elevated relevance values. Additionally, when we retrieve new attachments which are similar to the previous ones, this relevance will in time also propagate to e-mail messages with similar attachments.

Figure 2 shows an illustration of relevance sharing in the previous e-mail example. On the left, the relevance goes upwards from a child video-clip object, which has perhaps been indicated as relevant by the user. On the right, the relevance is spread downwards from the parent to its children. This process will result in the multimodal fusion of the information concerning different object types.

**PicSOM CBIR System**

PicSOM [9, 10] is a content-based information retrieval system developed at the Laboratory of Computer and Information Science at Helsinki University of Technology since 1998. The unique approach used in PicSOM is to have several *Self-Organizing Maps* (SOMs) [11] in parallel to index and determine the similarity of data objects. These parallel SOMs have been trained with separate data sets obtained by using different feature extraction algorithms on the same objects. So each SOM arranges the same objects differently, according to its particular multi-dimensional feature vectors.

PicSOM uses the principles of *query by example* [12] and *relevance feedback* [13, 14]. This means that the system shows the user a set of database objects, which the user then indicates as relevant or non-relevant to the current query, i.e., close to or far from what he is looking for. Based on this relevance feedback information PicSOM modifies its internal parameters so that it will display better objects in the next round. This is done by increasing the influence of those SOMs that give the most valuable similarity evaluation according to the current relevance feedback information. The user thus becomes an integral part of the query process, which can be seen as a form of supervised learning, where the user steers the system by providing feedback. A CBIR system implementing relevance feedback essentially tries to learn the optimal correspondence between the high-level human concepts and the low-level internal features used in the system.

The PicSOM CBIR system was initially designed to index and retrieve images only. Segmentation was introduced into PicSOM [15], and later we have used image segments in parallel with entire images to improve retrieval results [16]. This algorithm was then generalized to be used with multi-part objects [17] such as web-pages containing images and links [18] and video retrieval [19]. The PicSOM web page, with a list of publications on PicSOM and a working demonstration is located at http://www.cis.hut.fi/picsom.

**Figure 1**. A typical e-mail message with inline attachments (left), and a web page with text, link information and images (right). Hierarchical object trees are shown below.

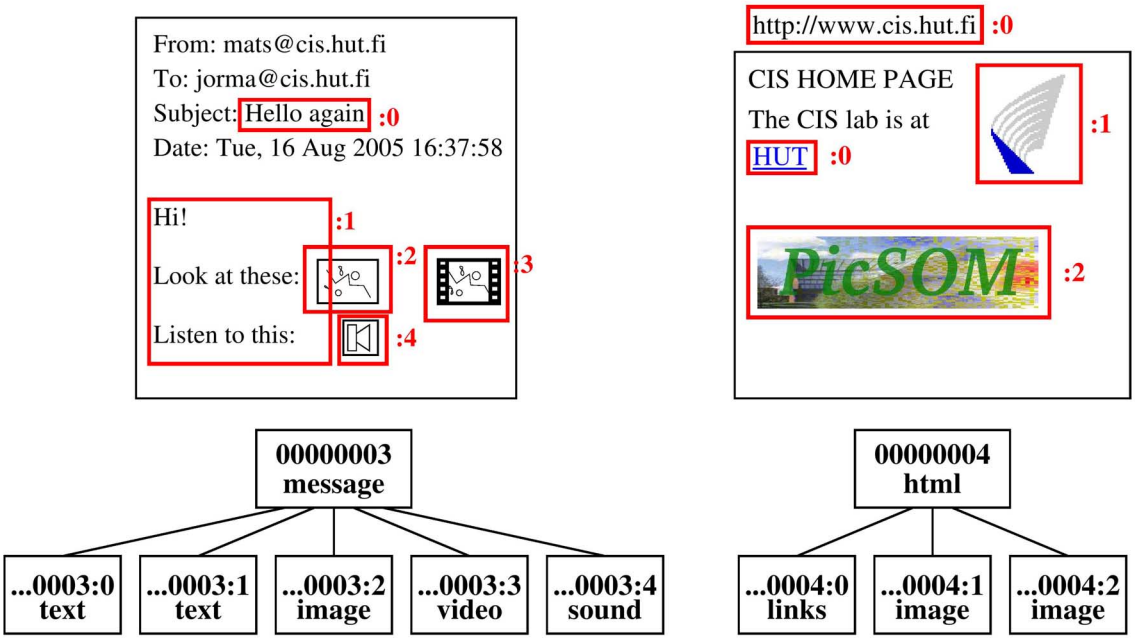
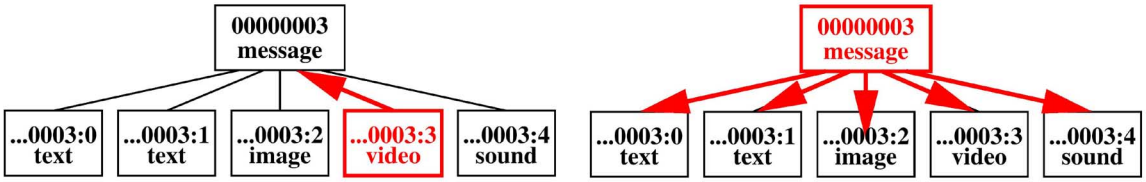



**Figure 2**. Relevance sharing in a multi-part e-mail object tree, going up from a child to its parent (left) or down from the parent to its children (right).



**Low-Level Features**

The PicSOM system implements a number of methods for extracting different low-level features, such as statistical visual features from images and image segments, and aural and motion-based features from video clips. These features include a set of MPEG-7 content descriptors [20, 10] and additionally some non-standard descriptors for color, shape and texture.

*Color*

Of the used MPEG-7 descriptors, Color Layout, Dominant Color, and Scalable Color describe the color content in image segments. In addition to the MPEG-7 color descriptors, both the average color in the CIE L*a*b* color space [21] and three first central moments of the color distribution are used as color features.

*Texture*

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

*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 [22]. In the feature vector, we include a fixed number of low-order Fourier expansion coefficients of the contour, interpreted as an arc-length parameterized complex function. The coefficients are normalized 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 [23] of the overall binary region shape. Also this descriptor is made invariant to affine transformations.

*Audio*

The Mel-scaled cepstral coefficient (MFCC), or shortly Mel cepstrum, was used as an aural feature. Mel cepstrum is the discrete cosine transform (DCT) applied to the logarithm of the mel-scaled filter bank energies. Mel cepstrum is commonly used for speech recognition, but can be used with other sounds as well [24].

*Motion*

The Motion Activity feature standardized by the MPEG-7 group [20] was used for video clips. The descriptor tries to capture the intuitive notion of "intensity" or "pace" of a video clip using the quantized standard deviation of motion vectors for classification. For example, the scoring of a goal in a soccer game is usually perceived as fast paced, while a normal TV interview is slow. Furthermore, using still image descriptors we can generate so called temporal video features which try to capture how the averaged still image features change over time in different spatial zones of the video frames.

## III. IMAGE SEGMENTATION AND SEMANTICS

Segmented images offer one of the simplest examples of a hierarchical object structure. Image segmentation essentially means dividing a digital image spatially into smaller disjoint parts according to some rule. The rule could, for example, simply be to divide the image into two by splitting it in the middle. To obtain more meaningful results most segmentation methods use some visual features of the image, for example color, to determine the segments.

Intuitively, one can easily understand the importance of segmentation in content-based image retrieval. Typically a picture contains several real-world objects, of which some are not relevant to a certain query. One might, for example, be searching a large image database looking for pictures of cars, but the surroundings of the car in a specific picture are not interesting. Then it would not matter if the retrieved image has a lot of grass or asphalt or if one can see a lot of the blue sky in the image or not, as long as there is a car in it. In this case it would thus be useful to be able to automatically segment the car images into at least two parts: the car and its surroundings.

Also if we consider the general problem of image understanding, we find that it is intrinsically linked to the problem of image segmentation. That is, if one understands an image, one can also tell what the distinct parts of it are. Segmentation thus seems to be a natural part of image understanding, which of course is one of the main problems in computer vision. A solution to the image understanding problem, however far away that may seem, will almost certainly contain segmentation in some form. With this in mind, the segmentation algorithm should ideally produce segments that correspond to our high-level semantic understanding of the image. This means that the segments should correspond directly to what humans see as different objects in the image, like the car and its surroundings, or even smaller elements, like the wheels of the car, the sun, etc.

The relationship between image segmentation and semantic concepts has become a subject of recent intensive study in the field of CBIR. The goal has been given various names ranging from "image-to-word transformation," [25] "matching words and pictures," [26] "image auto-annotation," [27] "automatic image captioning," [28] to "automatic image annotation," [29] depending on the selected viewpoint and the specific tasks the authors have been addressing. Various different technical methods and their combinations have been applied, including co-occurrence statistics [25], Expectation Maximization (EM) [26], Support Vector Machines (SVM) [29], Latent Semantic Analysis (LSA) [27], and Markov random fields (MRF) [30].

### Feature Extraction from Segmented Images

In reality, the automatically generated segments seldom correspond directly to our understanding of the image because they are created using only low-level visual features (e.g., color or texture). So for example, if color is used for segmentation, two nearby but different objects with similar color might end up in the same segment. This is actually an instance of the problem discussed in Section II, i.e., the semantic gap between the high-level semantic description used by humans and the low-level features used by computer systems. A computer vision system simply cannot understand an image based on only low-level feature information. But even so, we think that segmentation is useful in image retrieval because different, visually homogeneous regions somehow characterize the objects and scenes in the image. That is, we believe that the use of segmentation can give more information of the composition of the image than calculating features from the entire image alone. So for example, if you say that an image has three red segments and a green one, it is semantically much more informative than saying that its average color is yellow.

When we have segmented an image, we can create an object hierarchy where the original unsegmented entire image is the parent object and the segments, i.e. parts of the image, are child objects in a tree. Thus relevance feedback, together with the relevance sharing in hierarchical objects discussed in Section II, gives us a system where both similar images and images with similar segments contribute to each others' relevance values.

In Fig. 3, one can see an example of the use of segments in feature extraction. On the left we have calculated the average RGB color feature (the average of the red, green and blue color components of the image pixels) for the entire image. On the right we have segmented the image using *k*-means combined with region merging and calculated the feature for each segment separately. The segmented image provides much more information than by averaging over the entire image. One could argue that the same result could be achieved by simply concatenating the color features calculated from each segment to generate one long feature vector with all the results. And furthermore, that the extra information gained is just because we have a feature vector with higher dimensionality. But this is not the case, as each segment is treated as an object in its own right with its own low-dimensional color feature. So image similarity increases not only from images having similar features, but also from images having similar segments. For example, red segments might be found to be crucial to the retrieval in a particular case, but other colors might not be very important. Such a situation would be hard to benefit from in the retrieval process if all the color feature data is simply concatenated in some more or less arbitrary order.

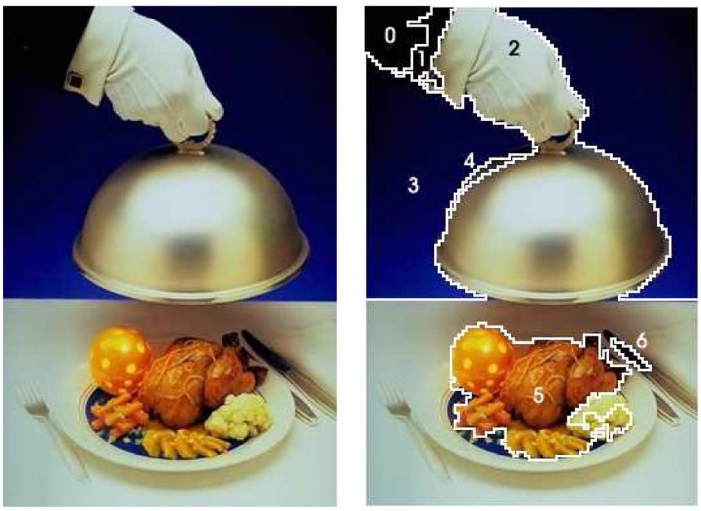
**Task of Semantic Keyword Focusing**

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 regarded as an unsupervised type 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 semantic keyword. We denote this database-wide ranking of locations according to 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 [31]. Many museums have annotated collections of digital images (e.g. [32]). 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

**Figure 3**. On the left we have calculated the average RGB color feature of an image. On the right we have segmented the image using *k*-means combined with region merging and calculated the average color feature for each segment separately.




| segment | red | green | blue |
|---|---|---|---|
| whole | 0.4263577 | 0.4061035 | 0.3861140 |
| 0 | 0.0086115 | 0.0086115 | 0.0084055 |
| 1 | 0.0301994 | 0.0261996 | 0.0217827 |
| 2 | 0.6279142 | 0.6435601 | 0.5504292 |
| 3 | 0.0186285 | 0.0175655 | 0.2216980 |
| 4 | 0.1802614 | 0.1599783 | 0.2017541 |
| 5 | 0.6643879 | 0.3807732 | 0.0891608 |
| 6 | 0.0406402 | 0.0320094 | 0.0552036 |

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

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 analyzed 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”. In a broader sense, any semantic 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.

**Keyword Focusing in the PicSOM Framework**

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 consist 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 semantically 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 [13] 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).

*Automatic Image Segmentation*

For the purpose of demonstrating keyword focusing to automatically obtained image segments, we employ a generic image segmentation method which is simple and somewhat rudimentary. In essence, the method is 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 color in the CIE L*a*b* color space [21].

The images in the database are segmented in two steps. In the first step ISODATA variant of *K*-means algorithm [33] with a *K* value 15 is used to compute an oversegmentation based on the color 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* color of regions $r_1$ and $r_2$ is used as the basis for the merging criterion. In addition, the multi-scale edge strength $e(r_1, r_2)$ 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 $r_i$:

$$C(r_1, r_2) = s(r_1, r_2)\,(d_{Lab}(r_1, r_2) + G\,e(r_1, r_2)), \quad (1)$$

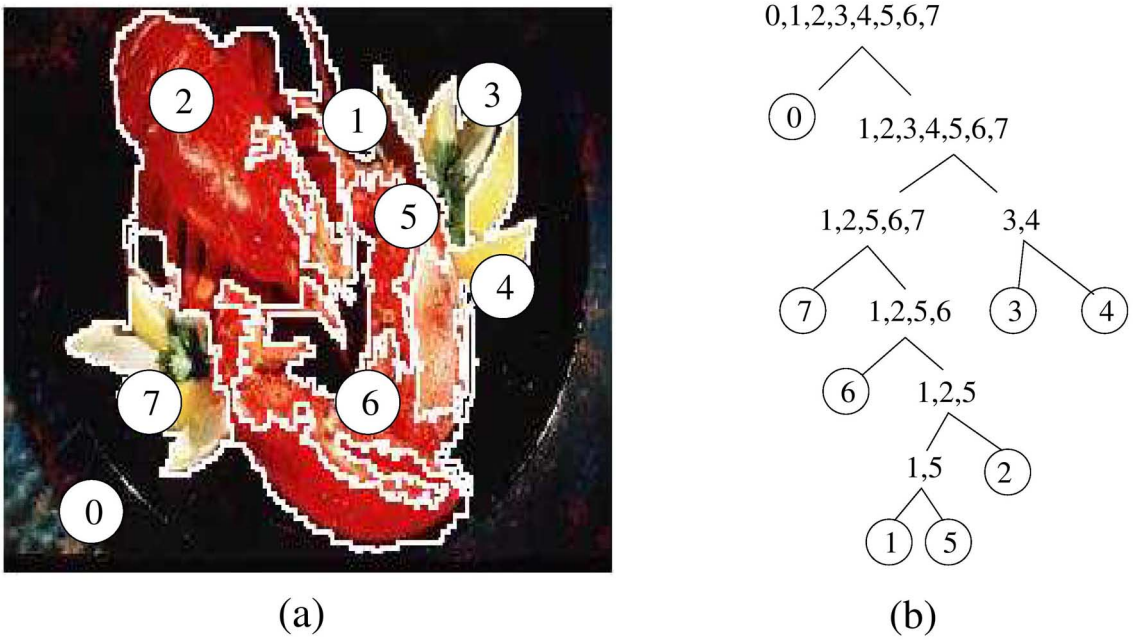
where

$$s(r_1, r_2) = \min(|r_1| / |I|, |r_2| / |I|, a) + b \quad (2)$$

is the size-weighting function, $|I|$ is the number of pixels in the image and $G$, $a$ 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 other applications. The same values 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 keyword focusing experiments alongside with the leaf segments. Figure 4a shows an example of a segmented image and Fig. 4b the corresponding segmentation hierarchy.

**Figure 4**. 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 have been circled in the tree.




*Using PicSOM as Keyword Focusing System*

The PicSOM CBIR framework can now be used to implement the unsupervised keyword focusing principle of the previous section on the segmented images. The PicSOM system takes two sets of images: a set of images annotated with a certain keyword (*positive examples*), and a set of auxiliary background images (*negative examples*). In the keyword focusing setting, the framework is used in off-line mode where the sets of example images are provided in their entirety at once. This is in contrast with an interactive CBIR setting where the example object sets are accumulated incrementally. 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.

In this context, the CBIR system can be considered to consist of a feedforward preprocessing stage, followed by an *inference stage*. In the preprocessing stage, both sets of example images are first hierarchically segmented (cf. Section III) and statistical visual features are then extracted from the segments (cf. Section II). The features are grouped into multiple feature spaces that are finally quantized using a variant of the

Self-Organizing Map (SOM) [11]. The inference stage implements the statistical correlation method for semantic keyword focusing. 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.

*Inference Stage in Keyword Focusing*

The statistical correlation inference is in PicSOM performed separately in 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 correlation process, 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 and negative densities are combined by map-wise weighted subtraction. 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 best-matching units (BMUs) on all the feature SOMs.

*Propagating Relevance within Segment Hierarchy*

To augment the implementation of the statistical correlation principle in the keyword focusing system, 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 semantic content of the keyword. To this end, the PicSOM keyword focusing system implements a simple multiplicative model for propagating the relevance scores of image segments upwards in the segmentation hierarchy like the one seen in Fig. 4.

**Examples of Keyword Focusing**

We demonstrate the use of PicSOM to intra-image keyword focusing with two image collections. As the input the PicSOM system uses, 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 a qualitatively different labeling, i.e. labels for the image segments.

To evaluate the system's performance in the focusing tasks, we have manually defined a ground truth against which the system's output 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. After having defined the ground truth classes for performance evaluation, we measure the system's performance with receiver operating characteristic (ROC) curves.

*Models Image Database*

The first database in which we demonstrate keyword focusing provides an easily understandable lightweight testbed for the framework. For this purpose, we have selected a 900-image subset of the commercial Corel database [31]. 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" as it has straightforward connections to both the image segmentation and the visual features, and therefore the results are easy to interpret. As the visual features for this database we used one color feature, color moments, and one local texture feature, the MPEG-7 Edge Histogram (cf. Section II). We expect to observe a different behavior of these two features in the keyword focusing task as color is directly related to the target keyword whereas the edge histogram is not.

To give insight on the operation of the outlined keyword focusing method, Fig. 5 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. Distribution of the color feature is shown on the top row and that of the MPEG-7 Edge Histogram descriptor on the bottom row. The input densities to the algorithm are densities in the columns (a) and (b). The column (c) is the outcome of the algorithm, and column (d) can be seen as the desired outcome. However, note that outside the feature space regions where there are segments in column (a), the final qualification values in column (c) have no significance to the outcome of the keyword focusing task.

As expected, the color 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 color 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.

Figures 6a and 6b show two example results and Fig. 6c the ROC curve for the whole models database when keyword focusing is applied on keyword "red." In Fig. 6b 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 Fig. 6c indicates the performance of the system to be very satisfactory in general, with some rare exceptions. This is also confirmed

**Figure 5**. Distributions of models database segments on two feature SOM surfaces. The top row (1) shows the distribution of color feature, the bottom row (2) the distribution of MPEG-7 Edge Histogram feature. Dark color corresponds to high density. Note that the distributions are normalized so that the maximum value always corresponds to black color. Therefore the shades in different subimages are not comparable, only the shapes of the distributions. The leftmost column (a) displays the distribution of all the segments in the images that are annotated with the keyword "red". The column (b) shows the distribution of all the segments in the models database. The column (c) shows the linear combination of the columns (a) and (b) which is used as the final qualification value of the segments with respect to that feature. The relevance is spread on in the columns (b) and (c) by the convolution outlined in Section III. The rightmost column (d) shows the distribution of manually confirmed "red" segments (true positives).

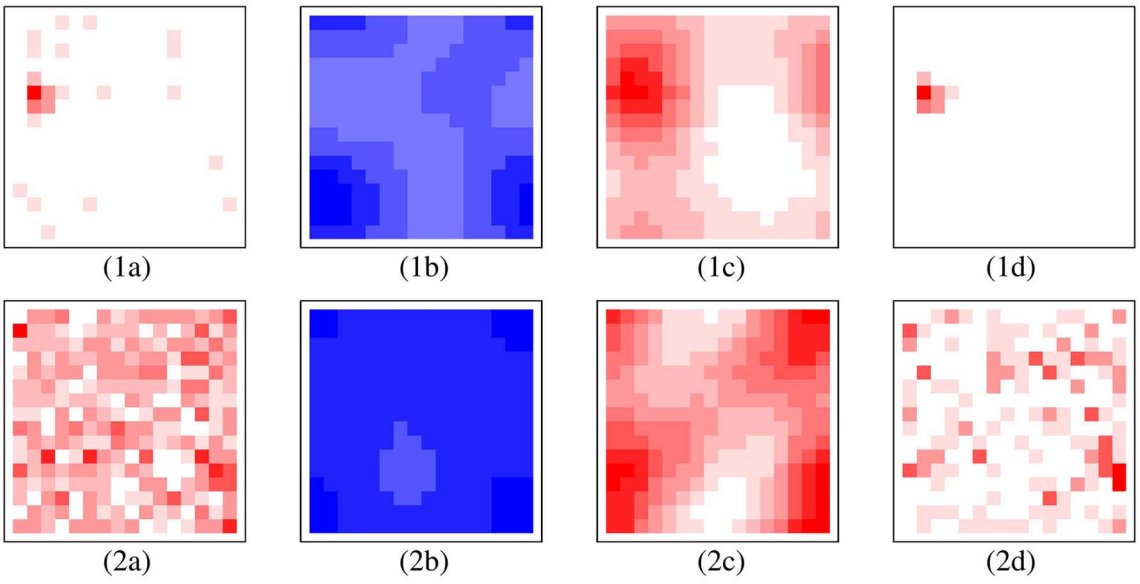
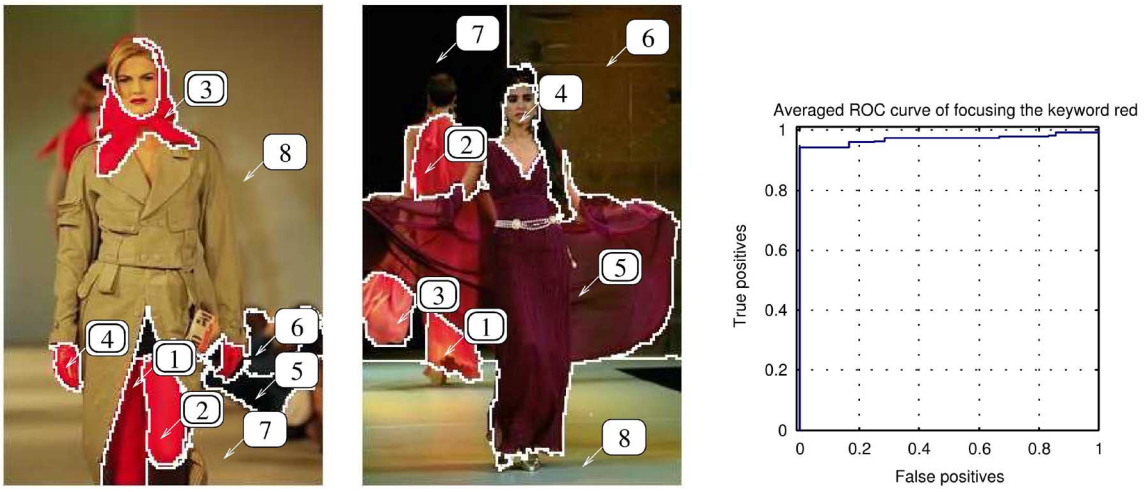


**Figure 6**. Subfigures (a) and (b) show examples of focusing the keyword "red" in the models database. The white borders in the image indicate the eight regions found by the segmentation algorithm. The number tags reflect the ordering of the segments produced by the focusing algorithm. The tags of truly red segments (defined for evaluation purposes only) are shown with double border. Subfigure (c) shows the averaged ROC curve of focusing keyword "red" for the whole models database.




(a) (b) (c)

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 and extraction of image semantics.

*101 Object Categories Database*

The second, more realistic and challenging example is the *101 Object Categories* database [34] of the PASCAL *Visual Object Classes Challenge* (http://www.pascal-network.org/challenges/VOC/). 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 color, texture or shape feature representations as was the case with "red" segments in the first database.

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

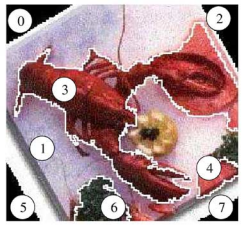
in the segmentation hierarchy, and (2) selecting the single best segment combination from the hierarchy. Both of these tasks were performed with and without the intraimage relevance propagation mechanism (cf. Section III). This gives four variants of the problem altogether. Figure 7 provides some example cases of keyword focusing in the "lobster" case. 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 Fig. 7 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 Fig. 7 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 8 shows the ROC curves for three cases of the keyword focusing task. 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 favored. Figure 8c shows the algorithm's performance in finding the best segment (* in Fig. 7) among the true lobster segments (+ in Fig. 7). This way the effect of the algorithm's performance in finding any lobster segment among all the segments is excluded. Figure 8c 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 Fig. 8c, the relative ordering of the algorithms with and without relevance propagation is similar to that in subfigure (b). This happens because the performance in 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 partly manages to favor 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.

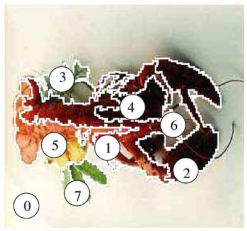
## IV. SEMANTIC VIDEO RETRIEVAL USING HIERARCHICAL OBJECTS

The PicSOM group participated for the first time in the NIST TRECVID video retrieval evaluation experiments for the TRECVID 2005 workshop [19]. In addition to implementing the necessary functionality into PicSOM we wanted to combine multimodal features with a text query and both positive and negative class models. The parallel SOMs of the PicSOM system were augmented with inverted file indices created from automatic speech recognition (ASR) and machine translation (MT) data provided by NIST.

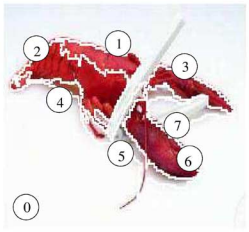
The TRECVID evaluations consisted of many different tasks, of which only the fully automatic search tasks will be described here. The automatic search tasks were run with the PicSOM system by using predefined search topics and no user interaction. The result of each run was a list of 2000 video clips ordered by their deemed relevance to the search topic. To our delight, the PicSOM results compared very well with the other systems taking part in the TRECVID 2005 evaluation.




(1a)




(1b)




(1c)

| (2a) | (3a) |
|---|---|
| * 3+ | * 3+ |
| 0,1,2,3,4,5,6,7 | 1,3,4,6 |
| 0,1,2,3,4,6,7 | 0,1,2,3,4,5,6,7 |
| 1,3,4,6,7 | 0,1,2,3,4,6,7 |
| 1,2,3,4,6,7 | 1,3,4,6,7 |
| 1,3,4,6 | 1,2,3,4,6,7 |
| 4 | 4 |
| 0 | 0 |
| 7 | 7 |
| 5 | 5 |
| 2 | 2 |
| 1 | 1 |
| 1,4 | 1,4 |
| 1,4,6 | 6 |
| 6 | 1,4,6 |

| (2b) | (3b) |
|---|---|
| 1 | * 1,2,4+ |
| 1,4+ | 1,2,4,6+ |
| * 1,2,4+ | 1,4+ |
| 1,2,4,6+ | 1,2,4,5,6,7 |
| 7 | 1 |
| 1,2,4,5,6,7 | 7 |
| 2+ | 5,7 |
| 5,7 | 2+ |
| 5+ | 5+ |
| 4+ | 4+ |
| 6+ | 6 |
| 3 | 3 |
| 0,1,2,3,4,5,6,7 | 0 |
| 0,3 | 0,1,2,3,4,5,6,7 |
| 0 | 0,3 |

| (2c) | (3c) |
|---|---|
| * 1,2,4 + | * 1,2,4+ |
| 2,4 + | 2,4+ |
| 3 + | 3+ |
| 6 + | 6+ |
| 2 + | 2+ |
| 1 + | 1+ |
| 4 + | 4+ |
| 0,3,5,6,7 | 0,3,5,6,7 |
| 0,3,5,7 | 5 |
| 0,5,7 | 0,1,2,3,4,5,6,7 |
| 0,1,2,3,4,5,6,7 | 0 |
| 5 | 7 |
| 0,7 | 0,3,5,7 |
| 0 | 0,7 |
| 7 | 0,5,7 |

**Figure 7**.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 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.

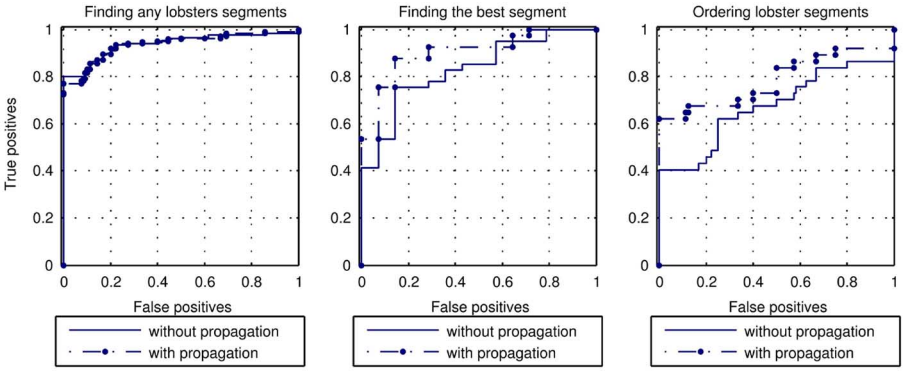



**Figure 8**.The averaged ROC curves of focusing keyword "lobster" in the 101 Object Categories database. Solid lines correspond to the focusing algorithm without the relevance propagation along segmentation hierarchy, the dashed line with the propagation. Subfigure (a) measures the focusing accuracy 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.

### Video and Audio Multi-Part Structure

A video clip consists of a series of still images shown rapidly in a sequence. Typical frame-rates are around 24 or 30 images per second. There can also be a sound track associated with the video sequence. Videos can thus be stored hierarchically so that the image frames and sound tracks are sub-objects in an object tree. Alternatively we can store only important key frames of the video as separate images, and have the entire video clip as the parent object. If the video itself is very long it might also be a good idea to segment it into shorter segments, for example into different scenes. Then the entire video would be the parent, with the shorter segments as sub-objects. Additionally, key frames and sounds can be assigned as children to the video segments creating a tree with three levels. In this way we can, for example, share relevance between two different videos having similar sounds or frames.

The videos in the supplied TRECVID 2005 database were several hour long segments of news broadcasts in three languages: Arabic, Chinese (Mandarin) and English. The long videos were divided into "stories", generally 1-2 minutes long, containing one news story or some other appropriate segment. These were held as sub-objects of the entire videos in a hierarchical object tree. The stories were further segmented into short "shots" of a few seconds each, containing one internally homogeneous scene. For example, an instantaneous change of the camera angle would usually indicate the beginning of a new shot, while a slow panning of the camera might be contained in the same shot. The shots belong to a new layer of sub-objects as children to the stories. From the individual shots, audio clips and key frame images were extracted as sub-objects. Finally textual content, created by automatic speech recognition and machine translation from the non-English languages was also added. The speech recognition and translation (ASR/MT) outputs were generated by off-the-shelf products and provided by NIST. This entire hierarchical structure is depicted in the left part of Fig. 9.

The search topics predefined by NIST were given with a textual instruction, a set of example images and a set of example videos. These were composed as a hierarchical object as shown in the right part of Fig. 9, and could thus easily be inserted in the system. Appropriate features were then calculated from the objects given for the search topic.

### Semantic Class Models

Some semantic classifications of the video clips in the training set were provided by NIST, for example a list of clips showing an explosion or fire. In the PicSOM system, a very informative visualization can be gained by mapping the feature vectors of the objects belonging to a specific semantic class as impulses on the SOM surfaces. This gives insight into how well a certain feature can cluster the vectors associated with that semantic concept. When used in the retrieval, the sign of the impulses can be adjusted to represent relevant (positive) and non-relevant (negative) concepts. The sparse value fields on the maps are low-pass filtered as usual to spread the information. This also helps visual inspection as the marked areas become larger and more uniform.

An example of such a mapping of the concept *explosion/fire* on the MPEG-7 Color Layout descriptor SOM can be seen in Fig. 10. Areas corresponding to objects of the concept are shown in shades of grey. As can be seen, the objects cluster quite well on the map into nearby locations. Theoretically the class distributions must be considered as estimates of the true distributions as they are finite and mapped on to the discrete two-dimensional grids of the SOMs, while the original feature space usually has a much

**Figure 9**. The hierarchical tree generated from the TRECVID 2005 videos (left), and the tree of a search topic (right).

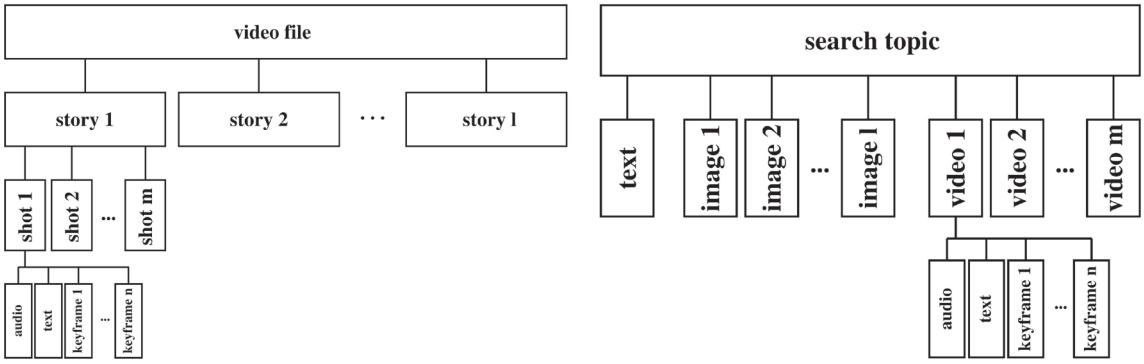
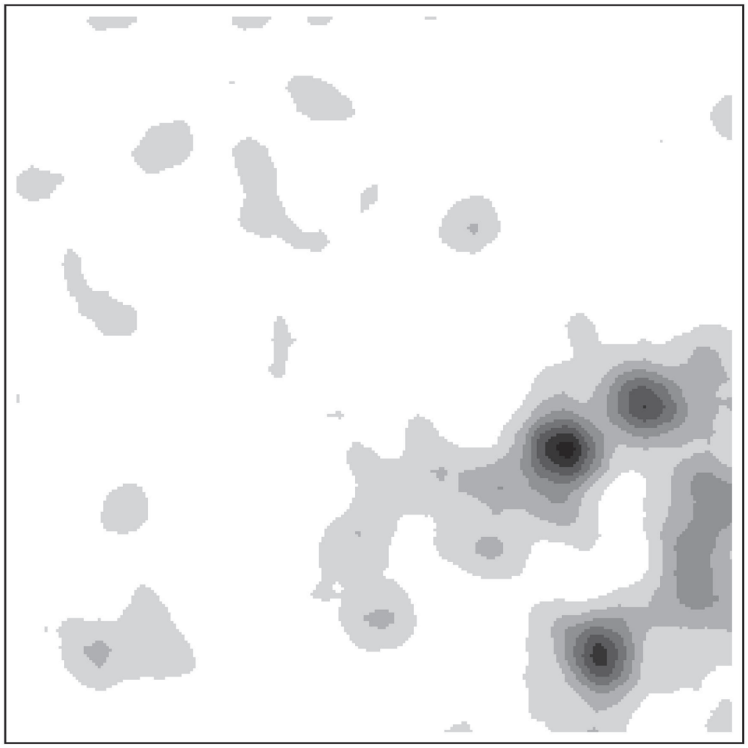



**Figure 10**. The *explosion/fire* class model mapped on the MPEG-7 Color Layout SOM, adapted from [19].



higher dimensionality. The class model idea was initially used in PicSOM for comparing different features [35] and for image group annotation in [36].

**Text Query Processing**

The ASR/MT output of non-English videos included additional information, such as if a certain proper name was a person, location or organization. Of these we used the person and location information to create an index of "known" proper names and whether they referred to persons or locations. Furthermore, discriminative words were picked up from the ontology descriptions (provided by NIST) to create a word-concept index. For example the word "minister" would map to the semantic class *government_leader*. This information was used in processing the text queries in the automatic search experiments before being used in the retrieval.

Proper names were initially identified in the text query by recognizing single or consecutive words with a capitalized first letter. These proper names were then compared with the index of known proper names by using the Levenshtein distance [37]. If the index name with the shortest distance was sufficiently close to the query name then the query name was deemed to be a misspelled version of the index name. The tolerance was dependent on the length of the query name, so that for short names a shorter Levenshtein distance was needed for acceptance. The identified misspelled words were corrected and the query string was cleaned, i.e., lowercased, dots and commas removed, and unnecessary text such as the preceding "Find shots of" discarded. Additionally, the word-concept index was used to identify words that might indicate useful class models. The presence of negative words, like a preceding "not" negated the class model. Finally if a person's name was identified previously, the class models *face* and *person* were added automatically.

Table 1 shows the transformations that would be made to this example query string: "Find shots of Omar Karammi, the former prime minister of Lebannon" (spelling errors intentional). The first row in the table shows the original string, and the second row the identifications found by the system. "Omar Karammi" is identified as a person and "Lebannon" as a location (even though they are misspelled). The identification WORD-CONCEPT under the word "minister" signifies that the word has been found in the word-concept index. The third row shows the actions or transformations performed. The initial "Find shots of" is deleted and the misspelled names are corrected. The corrections are marked by CORR with the corrected version in parenthesis. The fourth row shows the class models added, the sign before the class name identifies a positive or negative class model. The identified person automatically adds the class models *face* and *person* and "minister" adds *government_leader*. The last row shows the final processed text, capital letters, dots and commas removed, that was processed with the inverse file.

**Experiment Setting and Results**

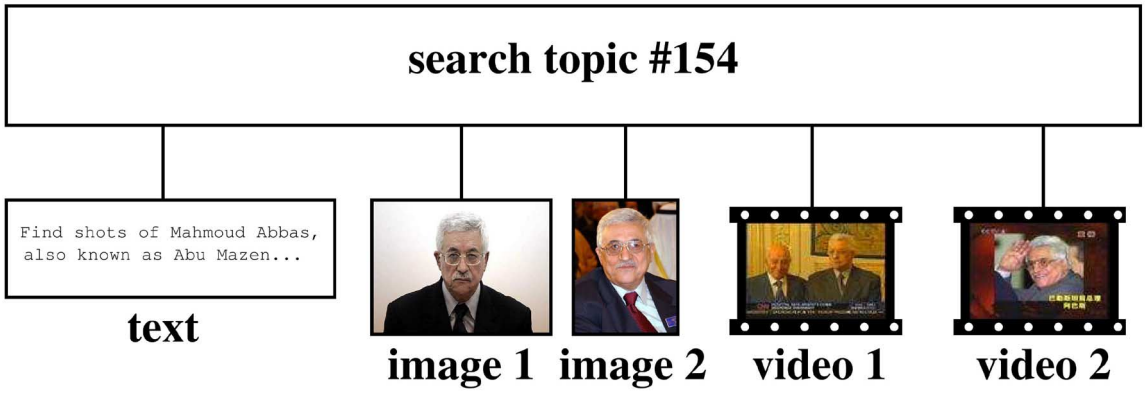
The experiments were run automatically as batch runs with 24 different search topics, ranging from finding a specific person to specific situations or events (e.g. an object burning or exploding). In Fig. 11 an example query taken from the actual TRECVID 2005 search topics is shown. The entire text query was "Find shots of Mahmoud Abbas, also known as Abu Mazen, prime minister of the Palestinian Authority." Two example images were given and nine video segments, of which only two are represented in the figure by their key frames.

In total the video shots were indexed using four video features (the MPEG-7 Motion activity and temporal versions of Average color, Color moments and Texture neighbor-

**Table 1**. An example of text query processing in automatic search.

| | | | | | | |
|---|---|---|---|---|---|---|
| **original** | *Find shots of* | *Omar Karammi,* | *the former prime* | *minister* | *of* | *Lebannon* |
| **identification** | | PERSON | | WORD-CONCEPT | | LOCATION |
| **actions** | DELETE | CORR(Omar Karami) | | | | CORR(Lebanon) |
| **classes** | | +face, +person | | +government_leader | | |
| **processed** | | *omar karami* | *the former prime* | *minister* | *of* | *lebanon* |

**Figure 11**. An example search topic from TRECVID 2005.




hood), three still image features from MPEG-7 (Color layout, Edge histogram, and Homogeneous texture) and one audio feature (Mel cepstrum). The sizes of the SOM layers were 4 × 4, 16 × 16, 64 × 64 and 256 × 256 units. Only one query round was performed returning 2000 video objects.

For most topics the PicSOM system performs on or above the median when compared to the other TRECVID 2005 participants. In only one case the result remains clearly under the median, whereas in seven cases out of a total of 24 it is clearly above it. By calculating the mean average precision over all topics we get an overall score for the retrieval accuracy. If we take only the best result from each research group into account, the PicSOM system ranked third out of nine groups. Considering that this was our first year in TRECVID and our main goal was just to make our system compatible with the evaluation interfaces, the results were gratifying and served to show that our approach for extracting semantic information from hierarchical multimodal data works efficiently.

## V. CONCLUSION AND FUTURE VIEWS

The novel idea presented in this work was to take advantage of known semantic relationships between different data objects in content-based information retrieval with Self-Organizing Maps. This was done by using hierarchical data objects where each object can have parents and children. In the relevance feedback process these hierarchies were used for relevance sharing, so that objects deemed relevant in a query would influence related objects as well.

Two examples of the application of this idea were addressed by using the PicSOM CBIR system: images with their segments, and videos with image frames, audio and motion features. In the case of segmented images we presented experimentation results showing how the semantic annotation can be focused from image level to specific image segments. The experiments with content-based video retrieval were carried out in the framework of the TRECVID 2005 evaluations and produced results that compared well with other participating retrieval systems.

In the light of the experiments, it is evident that the proposed hierarchical object model offers a viable approach for studying emergence of semantics from multimedia databases. 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 results that utilize the information contained in the data more efficiently. In the task of keyword focusing, the presented experiments also demonstrate the potential of a system architecture, where image data is first preprocessed 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-Organizing 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.

We started by discussing the increasing amounts of data that we produce and have to understand and use efficiently in the modern world. The goal of the research discussed here is to reduce this continuous stream of information into something that is manageable by humans by using content-based information retrieval methods equipped with semantic processing. The purpose is to create a semiautomatic search tool to aid humans in finding relevant information quickly. This problem will probably be one of the defining questions in information science of the 21st century. We would like to think that by enhancing existing CBIR methods with mechanisms that induce emergence of semantics, we could take a step towards the solution of this problem.

## REFERENCES

[1] P. Lyman and H. R. Varian, "How Much Information," http://www.sims.berkeley.edu/how-much-info-2003/, 2003.

[2] J. P. Eakins, "Towards intelligent image retrieval," Pattern Recognition, vol. 35, no. 1, pp. 3-14, January 2002.

[3] Y. Rui, T. S. Huang, and S.-F. Chang, "Image retrieval: Current techniques, promising directions, and open issues," Journal of Visual Communication and Image Representation, vol. 10, no. 1, pp. 39-62, March 1999.

[4] A. Gupta and R. Jain, "Visual information retrieval," Communications of the ACM, vol. 40, no. 5, pp. 70-79, May 1997.

[5] M. Koskela, J. Laaksonen, and E. Oja, "Inter-query relevance learning in PicSOM for content-based image retrieval," in Supplementary Proceedings of 13th International Conference on Artificial Neural Networks / 10th International Conference on Neural Information Processing (ICANN/ICONIP 2003), Istanbul, Turkey, June 2003, pp. 520-523.

[6] M. L. Cascia, S. Sethi, and S. Sclaroff, "Combining textual and visual cues for content-based image retrieval on the world wide web," in IEEE Workshop on Content-Based Access of Image and Video Libraries, Santa Barbara, CA, USA, 1998, pp. 24-28.

[7] D. Li, N. Dimitrova, M. Li, and I. K. Sethi, "Multimedia content processing through cross-modal association," in MULTIMEDIA '03: Proceedings of the eleventh ACM international conference on Multimedia. New York, NY, USA: ACM Press, 2003, pp. 604-611.

[8] J. M. Rehg, K. P. Murphy, and P. W. Fieguth., "Vision-based speaker detection using bayesian networks," in IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'99), vol. 2, 1999, pp. 110-116.

[9] J. Laaksonen, M. Koskela, S. Laakso, and E. Oja, "Self-organizing maps as a relevance feedback technique in content-based image retrieval," Pattern Analysis & Applications, vol. 4, no. 2+3, pp. 140-152, June 2001.

[10] J. Laaksonen, M. Koskela, and E. Oja, "PicSOM – Self-organizing image retrieval with MPEG-7 content descriptions," IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing, vol. 13, no. 4, pp. 841-853, July 2002.

[11] T. Kohonen, Self-Organizing Maps, 3rd ed., ser. Springer Series in Information Sciences. Berlin: Springer- Verlag, 2001, vol. 30.

[12] N.-S. Chang and K.-S. Fu, "Query-by-Pictorial-Example," IEEE Transactions on Software Engineering, vol. 6, no. 6, pp. 519-524, November 1980.

[13] G. Salton and M. J. McGill, Introduction to Modern Information Retrieval, ser. Computer Science Series. McGraw-Hill, New York, 1983.

[14] Y. Rui, T. S. Huang, M. O. , and S. Mehrotra, "Relevance feedback: A power tool in interactive content-based image retrieval," IEEE Transactions on Circuits and Systems for Video Technology, vol. 8, no. 5, pp. 644-655, September 1998.

[15] V. Viitaniemi, "Image segmentation in content-based image retrieval," Master's thesis, Laboratory of Computer and Information Science, Helsinki University of Technology, 2002.

[16] M. Sjöberg, J. Laaksonen, and V. Viitaniemi, "Using image segments in PicSOM CBIR system," in Proceedings of 13th Scandinavian Conference on Image Analysis (SCIA 2003), Halmstad, Sweden, June/July 2003, pp. 1106-1113.

[17] H. Muurinen, "Implementing Support for Content-Based Multimedia Message Retrieval in the PicSOM System," 2003, special assignment, Laboratory of Computer and Information Science, Helsinki University of Technology.

[18] M. Sjöberg and J. Laaksonen, "Content-based retrieval of web pages and other hierarchical objects with Self- Organizing Maps," in Proceedings of 15th International Conference on Artificial Neural Networks (ICANN 2005), Warsaw, Poland, September 2005, pp. 841-846.

[19] M. Koskela, J. Laaksonen, M. Sjöberg, and H. Muurinen, "PicSOM experiments in TRECVID 2005," in Proceedings of the TRECVID 2005 Workshop, Gaithersburg, MD, USA, November 2005, pp. 262-270.

[20] ISO/IEC, "Information technology - Multimedia content description interface - Part 3: Visual," 2002, 15938-3:2002(E).

[21] CIE, "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.

[22] K. Arbter, "Affine-invariant Fourier descriptors," in From Pixels to Features, J. C. Simon, Ed. Elsevier Science Publishers B.V.(North-Holland), 1989, pp. 153-164.

[23] A. Khotanzad and Y. H. Hong, "Invariant image recognition by Zernike moments," IEEE Transaction on Pattern Analysis and Machine Intelligence, vol. 12, no. 5, pp. 489-497, 1990.

[24] S. B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," in Readings in speech recognition, A. Waibel and K. Lee, Eds. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1990, pp. 65-74.
[25] Y. Mori, H. Takahashi, and R. 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.
[26] K. Barnard, P. Duygulu, N. de Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching words and pictures," Journal of Machine Learning Research, Special Issue on Machine Learning Methods for Text and Images, vol. 3, pp. 1107-1135, February 2003.
[27] F. Monay and D. Gatica-Perez, "On image auto-annotation with latent space models," in Proceedings of the eleventh ACM international conference on Multimedia, Berkeley, CA, 2003, pp. 275-278.
[28] J.-Y. Pan, H.-J. Yang, P. Duygulu, and C. Faloutsos, "Automatic image captioning," in Proceedings of the 2004 IEEE International Conference on Multimedia and Expo, Taipei, Taiwan, June 2004.
[29] J. Fan, Y. Gao, and H. 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, New York, NY, Oct. 2004, pp. 540-547.
[30] P. Carbonetto, N. de Freitas, and K. Barnard, "A statistical model for general contextual object recognition," in Proceedings of the Eight European Conference on Computer Vision, Prague, May 2004.
[31] "The Corel Corporation WWW home page", http://www.corel.com, 1999.
[32] "The Fine Arts Museum of San Francisco", http://www.thinker.org, 2005.
[33] R. J. Schalkoff, Pattern Recognition: Statistical, Structural and Neural Approaches. JohnWiley & Sons, Ltd., 1992.
[34] L. Fei-Fei, R. Fergus, and P. 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.
[35] J. Laaksonen, M. Koskela, and E. Oja, "Probability interpretation of distributions on SOM surfaces," in Proceedings of Workshop on Self-Organizing Maps (WSOM'03), Hibikino, Kitakyushu, Japan, September 2003, pp. 77-82.
[36] M. Koskela and J. Laaksonen, "Semantic annotation of image groups with Self-Organizing Maps," in Proceedings of 4th International Conference on Image and Video Retrieval (CIVR 2005), Singapore, July 2005, pp. 518-527.
[37] P. E. Black, "Algorithms and theory of computation handbook," in NIST Dictionary of Algorithms and Data Structures. CRC Press LLC, 1999, http://www.nist.gov/dads/HTML/Levenshtein.html.