TKK Reports in Information and Computer Science

Espoo 2008 TKK-ICS-R11

EVALUATION OF POINTER CLICK RELEVANCE FEEDBACK IN PICSOM

Deliverable D1.2 of FP7 project n° 216529 PinView

Ville Viitaniemi and Jorma Laaksonen

TEKNILLINEN KORKEAKOULU
TEKNISKA HÖGSKOLAN
HELSINKI UNIVERSITY OF TECHNOLOGY
TECHNISCHE UNIVERSITÄT HELSINKI
UNIVERSITE DE TECHNOLOGIE D'HELSINKI

TKK Reports in Information and Computer Science

Espoo 2008 TKK-ICS-R11

EVALUATION OF POINTER CLICK RELEVANCE FEEDBACK IN PICSOM

Deliverable D1.2 of FP7 project n° 216529 PinView

Ville Viitaniemi and Jorma Laaksonen

Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science

Teknillinen korkeakoulu
Informaatio- ja luonnontieteiden tiedekunta
Tietojenkäsittelytieteen laitos

Distribution:
Helsinki University of Technology
Faculty of Information and Natural Sciences
Department of Information and Computer Science
P.O.Box 5400
FI-02015 TKK
FINLAND
URL: http://ics.tkk.fi
Tel. +358 9 451 1
Fax +358 9 451 3369
E-mail: series@ics.tkk.fi

© Ville Viitaniemi and Jorma Laaksonen

ISBN 978-951-22-9671-2 (Print)
ISBN 978-951-22-9672-9 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)
URL: http://lib.tkk.fi/Reports/2008/isbn9789512296729.pdf

TKK ICS
Espoo 2008

Abstract: This report presents the results of a series of experiments where knowledge of the most relevant part of images is given as additional information to a content-based image retrieval system. The most relevant parts have been identified by search-task-dependent pointer clicks on the images. As such they provide a rudimentary form of explicit *enriched relevance feedback* and to some extent mimic genuine implicit *eye movement measurements* which are essential ingredients of the PinView project.

Keywords: content-based image retrieval, enriched relevance feedback, automatic image segmentation

Acknowledgement: The research leading to these results has received funding from the European Community's Seventh Framework Programme (FP7/2007–2013) under grant agreement n° 216529, Personal Information Navigator Adapting Through Viewing, PinView. All deliverables of the PinView project are available at `http://www.pinview.eu/` .

CONTENTS

1 OVERVIEW

This Deliverable of the *Personal Information Navigator Adapting Through Viewing*, PinView, project, funded by the European Community's Seventh Framework Programme under Grant Agreement n° 216529, constitutes the output of Task 1.2 *Evaluation of click location feedback*. Pointer clicks are examples of *enriched relevance feedback* that can be used to improve the performance of *content-based information retrieval* (CBIR). With them the user of a CBIR system is able to identify in an image the area that he or she considers to be the most relevant one for the current search task. The pointer clicks are also to some extent able to mimic the *implicit eye movements and gaze fixations* the test subjects make while evaluating the fitness of a seen image to a particular search topic. The eye movement recordings and the information that can be extracted from them are essential ingredients of the PinView project. In this regard, the current study paves the way for later research where actual eye movements are being employed for discovering relevance inside image collages and single images.

The research was implemented by using the existing CBIR system, PicSOM, developed at TKK. The PicSOM system contains a subsystem for automatic image segmentation and presentation of segment hierarchies. The extracted segments and features calculated from them were in the experiments used together with the click locations to model the most relevant or discriminative areas of the images. Overall, the retrieval performance obtained in the simulations was worse than hypothesised. Most likely, this result can be explained by difficulties related to accurate automatic image segmentation. As an alternative approach, we also used quadtrees for the regional presentation of images, and this approach gave better results than the use of the genuine image segments.

One part of the described work has been the setting up the evaluation database and the collection of the pointer click data together with eye movement recordings. Later these recordings have been post-processed to be included in the eye movement recording database created in Task 8.3 *Eye movement data collection campaign*.

The work will be continued further in TKK's participation in PinView's Tasks 6.1 *Local features for sub-categorisation* and 6.2 *Information fusion and confidence*.

2 INTRODUCTION

The research reported in this deliverable is part of the the *Personal Information Navigator Adapting Through Viewing*, PinView, project[1]. A central issue in the PinView project is to study how *enriched relevance feedback* can be used to improve *content-based information retrieval* (CBIR). In this work, we have collected pointer click locations on images shown to test users and studied if that kind of additional information can be used to improve the accuracy in interactive CBIR.

The click locations can identify in an image the area that the user considers to be the most relevant one for the current search task. This can be seen as an improvement over the traditional Boolean-valued relevance feedback where entire images are marked either as relevant or non-relevant. However, a single point in the image is seldom sufficient to reveal the true extent of the relevant object. This shortcoming can be partially relieved by combining the click location with results of image partitioning or automatic image segmentation. Then the image part or segment where the click location resides can be used to model the relevant object.

The *explicit* pointer click locations are also to some extent able to mimic the *implicit* eye movements and gaze fixations the test subjects make while evaluating the fitness of a seen image to a particular search topic. The eye movement recordings and the information that can be extracted from them are essential ingredients of the PinView project. In this regard, the current study paves the way for later research where actual eye movements are being employed for discovering relevance inside image collages and single images.

The PicSOM CBIR system, developed at Helsinki University of Technology, contains mechanisms for simulating interactive content-based image retrieval. We have also implemented an automatic image segmentation subsystem and techniques for matching click locations or other kinds of reference positions with the obtained image segments. Visual feature extraction and object indexing have consequently been possible not only for entire images, but also for their content-based segments and other geometrical regions.

In the current experiments, we have first recorded on-line pointer clicks with real subjects. The data collection setting has been such that also the eye movements and fixations of the subjects have been recorded simultaneously. In off-line simulations we have then evaluated the degree of change in the accuracy of content-based image retrieval when the pointer click locations have been utilised as additional source of information.

The rest of this report is organised as follows. In Section 3, we briefly describe the PicSOM content-based image retrieval system used in the experiments. This section includes the descriptions of the implementation of the relevance feedback mechanism, image segmentation and feature extraction. Section 4 introduces the image database used in the experiments and describes how the click location data was collected. In Section 5, the setting for the empirical evaluation is presented and the results of the experiments are presented. Finally in Section 6, we present conclusions and a discussion on the related future work in the PinView project.

3 PICSOM CBIR SYSTEM

PicSOM[2] [7, 8] is a content-based information retrieval system developed at Helsinki University of Technology since 1998, first in the Laboratory of Computer and Infor-

[1] http://www.pinview.eu/

[2] http://www.cis.hut.fi/picsom

mation Science and later in the Department of Information and Computer Science. The unique approach used in PicSOM is to have several Self-Organizing Maps (SOMs) [4] 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 the particular multi-dimensional feature vectors used in its training.

PicSOM uses the principles of *query by example* [1] and *relevance feedback* [13, 12] in implementing iterative and interactive image retrieval. 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 in the next round it will display objects that resemble those that had been marked as relevant. 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 by the system.

The PicSOM CBIR system was initially designed to index and retrieve images only. Segmentation was introduced into PicSOM [18], and later we have used image segments in parallel with entire images to improve retrieval results [15]. This algorithm was then generalised to be used with multi-part objects such as web-pages containing images and links [14] and video retrieval [5, 16, 6].

3.1 Self-Organizing Map

The Self-Organizing Map [4] is a neurally-motivated unsupervised learning technique which has been used in many data-analysis tasks. A genuine feature of the Self-Organizing Map is its ability to form a nonlinear mapping of a high-dimensional input space to a typically two-dimensional grid of artificial neural units. During the training phase of a SOM, the *model vectors* in its neurons get values which form a *topographic* or *topology-preserving* mapping. Through this mapping, vectors that reside near each other in the input space are mapped into nearby map units in the SOM grid. Patterns that are mutually similar in respect to the given feature extraction scheme are thus most often located near each other on the SOM surface.

The training of a Self-Organizing Map starts from the situation where the model vectors $\mathbf{m}_i$ of each map unit i are initialised with random values. For each input feature vector or sample $\mathbf{x}(t)$, the "winner" or *best-matching* map unit (BMU) $c(\mathbf{x})$ is identified on the map by the condition

$$\forall i: \quad \|\mathbf{x}(t) - \mathbf{m}_{c(\mathbf{x})}(t)\| \leq \|\mathbf{x}(t) - \mathbf{m}_i(t)\| \ , \tag{1}$$

where $\|\cdot\|$ is commonly the Euclidean norm. After finding the BMU, a subset of the model vectors constituting a neighbourhood centred around node $c(\mathbf{x})$ are updated as

$$\mathbf{m}_i(t+1) = \mathbf{m}_i(t) + h(t; c(\mathbf{x}), i)(\mathbf{x}(t) - \mathbf{m}_i(t)) \ . \tag{2}$$

Here $h(t; c(\mathbf{x}), i)$ is the "neighbourhood function", a decreasing function of the distance between the i-th and $c(\mathbf{x})$-th nodes on the map grid. The training is reiterated over the available samples, and the value of $h(t; c(\mathbf{x}), i)$ is allowed to decrease in time to guarantee the convergence of the prototype vectors $\mathbf{m}_i$. Large values of the neighbourhood function $h(t; c(\mathbf{x}), i)$ in the beginning of the training initialise the network, and small values on later iterations are needed in fine-tuning. The outcome of training a SOM is essentially the *topographic ordering* of the training data samples on the surface of the map grid.

3.2 Relevance feedback and image scoring

In iterative CBIR, the system presents in a visual interface some images or other objects to the user, who then marks a subset of them as relevant to the present query. This relevance information is fed back to the system, which then tries to find more similar objects and returns them in the next query round. In PicSOM, multiple SOMs are used in parallel, each created with different low-level visual features. Each image is therefore used multiple times as input to different feature extraction techniques and the resulting feature vectors are used in training different SOMs. When an image SOM has been trained, all the feature vectors $\mathbf{x}$ are once more mapped to it, each one in its best-matching unit. Every SOM unit and its model vector $\mathbf{m}$ is then assigned a *visual label* from the image whose feature vector was the nearest to the model vector.

The different feature extraction schemes and the SOMs that result from the training impose different similarity functions on the images. As a consequence, two images whose contents are similar to each other with respect to some feature will be mapped to nearby units on that SOM's surface, whereas they will be located far apart on another feature's SOM if that feature regards them as dissimilar. Every interactive image query is unique and each user of a CBIR system has his or her own transient view of images' similarity and relevance to the specific query. Therefore, a system structure capable of holding many simultaneous similarity representations can adapt to different kinds of retrieval tasks.

Relevance feedback has been implemented in PicSOM by using the parallel Self-Organizing Maps. Each image presented in the interactive on-line use of the system is graded by the user as either relevant or non-relevant. All these relevance grades are then projected to the best-matching SOM units of the graded images on all the different SOM surfaces. Maps where many relevant images are mapped in same or nearby SOM units agree well with the user's conception on the relevance and semantic similarity of the images. When we assume that similar images are located near each other on the SOM surfaces, we are motivated to spread the relevance information placed in the SOM units also to neighbouring units.

This *relevance spreading* is implemented in PicSOM as follows. All images marked as relevant are first given an equal positive value inversely proportional to their number. Likewise, non-relevant images receive negative values that are inversely proportional to the number of non-relevant images. The overall sum of these relevance values is thus zero. The values are then summed in the BMUs of the images, and the resulting sparse value fields are low-pass filtered to produce the final *relevance score* or *qualification value* for each SOM unit. With a SOM size of 64×64 map units, as used in the current experiments, a typical size for the radius of the low-pass filter is four map units. Figure 1 illustrates how the positive and negative responses are first mapped on a SOM surface, displayed there with red and blue map units, respectively, and how the responses are expanded in the convolution in order to spread the relevance assignments to the most similar images.

As the final step, each image is given a total qualification value obtained as a sum of the qualification values from its best-matching or index units from the SOM surfaces of the different features. Those yet unseen images which have the highest qualification values will then be shown to the user on the next query round. In PicSOM, features that fail to coincide with the user's conceptions always produce lower qualification values than those that match the user's expectations. As a consequence, the different features do not need to be explicitly weighted as the system automatically takes care of weighting their opinions.

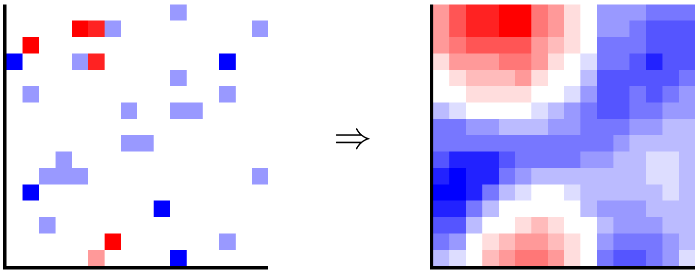

Figure 1: An example of how a SOM surface, on which the images selected and rejected by the user are shown with red and blue marks, respectively, are convolved with a low-pass filter.

3.3 Image segmentation

The images are segmented with a generic image segmentation method that employs an area-based region merging algorithm to construct segments that are homogeneous in terms of colour and texture. The segmentation procedure consists of two steps. In the first step the K-means algorithm with a K value 13 is used to compute an oversegmentation of the images based on RGB colour values. This step typically results in a few thousand separate segments. In the second step pairs of most suitable neighbouring segments are iteratively merged until only 25 segments are left. The suitability-for-merge criterion C_1

$$\mathrm{C}_1(r_1, r_2) = -\mathrm{s}_1\left(r_1, r_2\right) \mathrm{d}_1\left(r_1, r_2\right) \tag{3}$$

takes into account the visual similarity d_1 of the segments in combined feature space of CIE L*a*b* colour moments and neighbourhood brightness histogram (see Section 3.4), as well as the sizes $|r_i|$ of the to-be-merged regions r_i. Here

$$\mathrm{s}_1(r_1, r_2) = \min(|r_1|/|I|, |r_2|/|I|, a_1) + b_1 \tag{4}$$

is the size-weighting function where $|I|$ is the number of pixels in the image. a_1 and b_1 are free parameters of the method. The merging algorithm and parameter values have been selected to give visually feasible results for photographs and other images in earlier applications. The resulting 25 segments are subsequently denoted *leaf segments*.

Besides the leaf segments, we also record the hierarchy tree of *composite segments* that results from continuing the pairwise region-merging algorithm until only one region remains. Here we use a slightly different merging criterion

$$\mathrm{C}_2(r_1, r_2) = -\mathrm{s}_2\left(r_1, r_2\right) \mathrm{d}_2\left(r_1, r_2\right). \tag{5}$$

Here d_2 is distance in the combined visual feature space of CIE L*a*b* colour, neighbourhood brightness histogram and HSV colour histogram, and s_2 is the size-weighting function

$$\mathrm{s}_2(r_1, r_2) = \sqrt{\min(|r_1|/|I|, |r_2|/|I|, a_2) + b_2}. \tag{6}$$

Figure 2 shows an image of a dog along with the hierarchy of the segments the segmentation algorithm finds. The segment hierarchy divides into subtrees for the dog (left) and the background. For clarity, the figure shows only the part of the hierarchy that is recorded after the merging process has already reduced the number of segments to just eight.

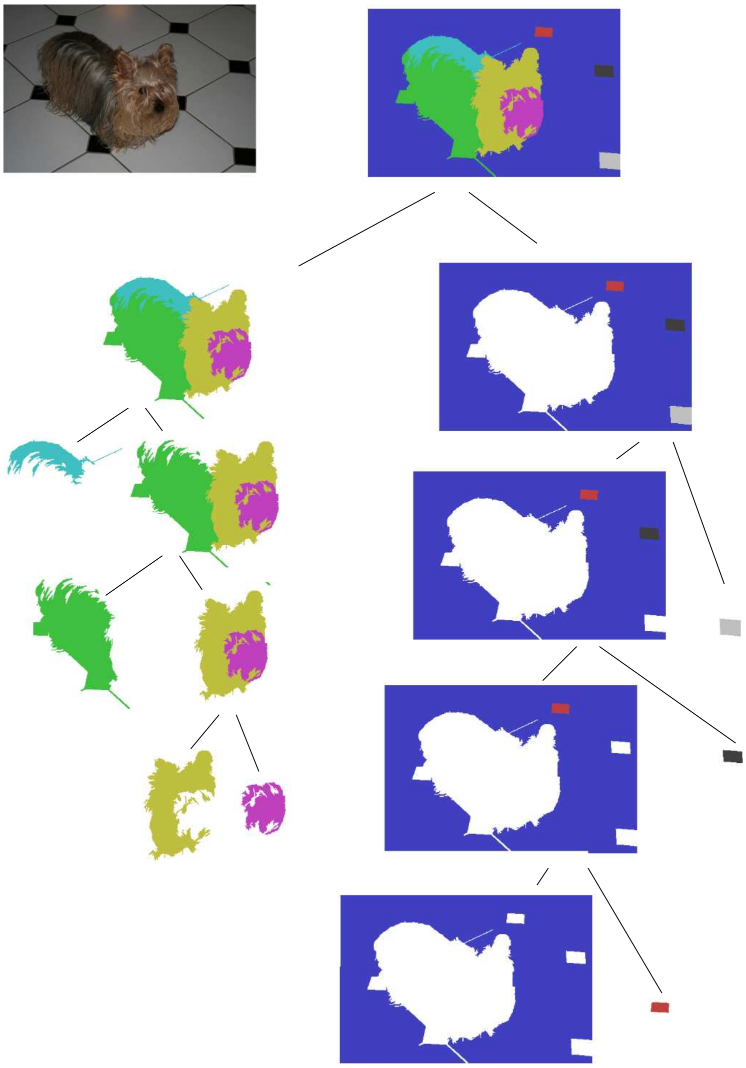

Figure 2: An image of a dog on top left corner and the segment hierarchy the segmentation algorithm discovers.

In addition to the above outlined content-driven segmentation algorithm, we also subdivide the images purely geometrically regardless of the image content. These two approaches will then be compared in the experiments. For geometrical partitioning, we employ a quadtree structure with three levels. The levels partition the image area with regular rectangular grids. Each upper level patch divides into a 2 × 2 subgrid on the next lower level. This results in levels dividing the image into 1 × 1, 2 × 2 and 4 × 4 rectangular patches. The patches form a three-level segment hierarchy in a natural fashion. Figure 3 illustrates the geometrical quadtree partitioning.

3.4 Features

Several different types of features are extracted from the database images as well as from the automatically obtained image segments in order to characterise their visual

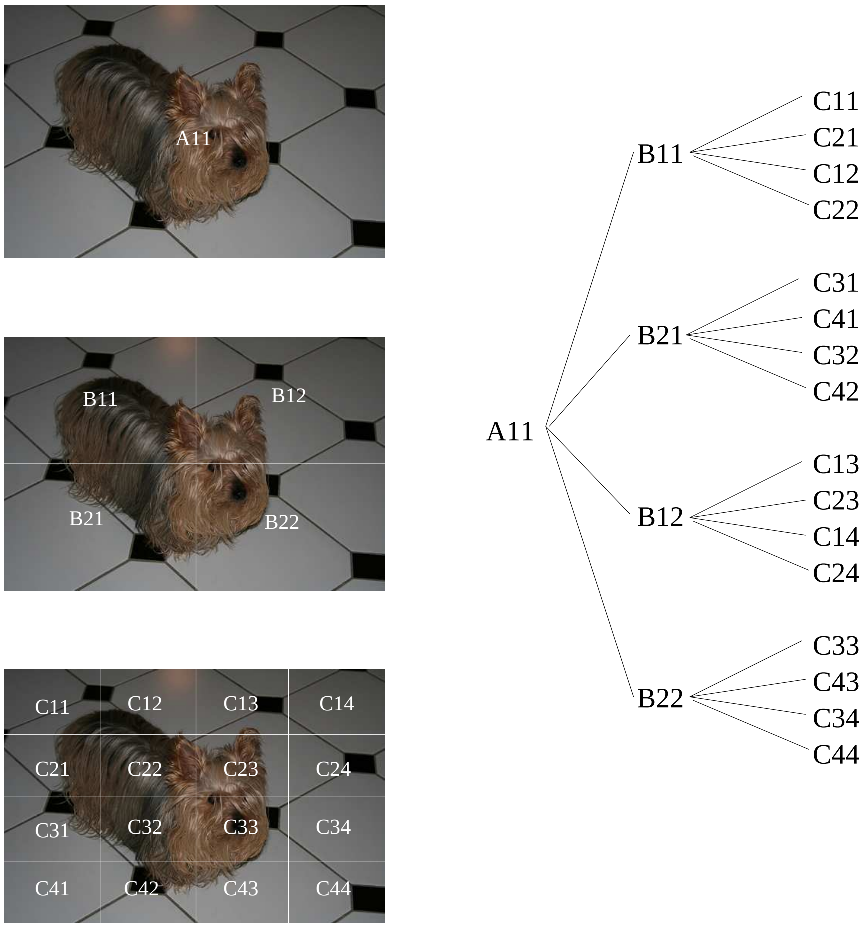

Figure 3: Left: quadtree division of a dog image into rectangular patches, right: the corresponding hierarchy tree of the patches.

properties. In our experiments we have observed (e.g.[19]) that rather extensive sets of features often are beneficial in image content analysis.

Image features

The extracted image feature types are listed in Table 1. Dimensionalities of the feature vectors are given in the rightmost column of the table. The first four rows correspond to features that more or less closely resemble the ColorLayout, DominantColor, EdgeHistogram and ScalableColor features of the MPEG-7 standard [3]. Of the non-standard descriptors, the colour moment feature encodes the first three central moments of the colour distribution in the CIE L*a*b* colour space. The low-order colour moments have been found [17] to be an efficient and robust way to characterise colour distributions.

Local shapes are described with several features. Three features are based on detecting local edge segments of four orientations with Sobel masks. The edge statistics are summarised with histograms, co-occurrence matrices and magnitude of a fixed-size Fourier transform of the edge image. In addition, local texture is described with a statistics of pixels' brightness (Y) values in the YIQ colour representation relative to the values of their 8-neighbours.

For the interest point SIFT histogram feature of the last row, the interest points are detected with a Harris-Laplace detector [11]. A histogram was formed of the SIFT descriptions [10]—based on local gradient orientation—of the interest points. The histogram bins were chosen by clustering the SIFT descriptors of all the interest

Feature	Tiling	Dim.
DCT coefficients of average colour in rectangular grid	global	12
CIE L*a*b* colour of two dominant colour clusters	global	6
Histogram of local edge statistics	4×4	80
Haar transform of quantised HSV colour histogram	global	256
Average CIE L*a*b* colour	5	15
Three central moments of CIE L*a*b* colour distribution	5	45
Histogram of four Sobel edge directions	5	20
Co-occurrence matrix of four Sobel edge directions	5	80
Magnitude of the 16×16 FFT of Sobel edge image	global	128
Histogram of relative brightness of neighbouring pixels	5	40
Histogram of interest point SIFT features	global	256

Table 1: Visual features extracted from images.

points in the training images with the Linde-Buzo-Gray (LBG) algorithm [9]. In our experiments [20] we have found that rather large codebooks with several thousand bins can be beneficial. For these experiments, however, we trade some accuracy of the representation for reduced computing time and storage requirements, and use a smaller histogram with 256 bins.

The column "Tiling" of the table shows that some of the features are calculated truly globally, such as the global colour histogram feature of the fourth row, others, such as the edge histogram feature of the third row, encode some spatial information by using a fixed image grid. These feature vectors are formed by concatenating the descriptors of individual tiles. The features calculated for five tiles employ a centre-surround tiling mask (Figure 4) where the image area is divided into four tiles by the two diagonals of the image, on top of which a circular centre tile is overlaid.

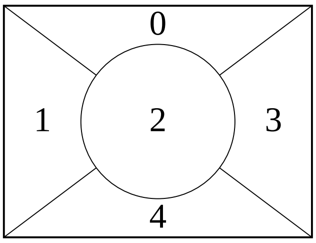

Figure 4: The five-part tiling mask employed in calculation of some of the features.

Segment features

The segment features have been obtained similarly to the image-wide features with the exception that only those pixels that reside inside that particular segment have been taken into account in the calculation. The features have been extracted for all the leaf as well as composite segments of each image. In contrast to whole image features, the segment features do not spatially subdivide the segments into tiles but describe the segments as whole. The extracted segment features, listed in Table 2, are a subset of the global image features of Table 1.

3.5 Utilisation of pointer click locations

The current implementation of pointer click utilisation in PicSOM is quite straightforward. When the user of the on-line CBIR system clicks a pointer button on an image, the image is considered to relevant for the ongoing retrieval task. In addition,

Feature	Dimensionality
DCT coefficients of average colour in rectangular grid	12
CIE L*a*b* colour of two dominant colour clusters	6
Haar transform of quantised HSV colour histogram	256
Moments of CIE L*a*b* distribution	9
Co-occurrence matrix of four Sobel edge directions	16

Table 2: Visual features extracted from image segments.

the location of the click is assumed to be the most informative or discriminative area of the object for the positive relevance judgement. If the image is not relevant, then the user hits a keyboard key to indicate his or her negative relevance judgement for it. After a fixed number of images, typically 20 in our experiments, have been so processed, the PicSOM system uses the collected relevance feedback information to select the next set of images to be shown to the user, one by one.

The on-line relevance assessment procedure can be simulated in off-line experiments if all necessary information is available for the PicSOM system's interactive query simulation subsystem. For the described setting one first needs ground truth relevance data for each image in the used database. In practice this means that a number of search tasks have been identified and for each task all relevant images have been identified. Such sets of relevant images for particular search tasks will in the subsequent sections be called also as *object classes*. For studying the possible advantage brought by the use of click locations, one will naturally also need to record the pointer click xy-coordinates for each relevant image. This information can be stored in files together with the object class definitions and then used in off-line simulations.

When automatic image segmentation is being used in PicSOM, the system will have two different kinds of SOMs, first those created from feature vectors extracted from entire images and then those created from features calculated from image segments. In our current implementation we have extracted segment-wise features from all leaf segment and all composite segments of the segment hierarchy tree. This means that from an image with 25 leaf segments, there have been a total of 49 segments for which the features have been extracted. In the notation of the PicSOM system, each one of these segments is an object of the image database in its own right. Consequently, all leaf and composite segments are able to carry user-specified relevance assessments. In addition, they will obtain their own relevance scores in similarity-based retrieval, as described above in Section 3.2 for entire image objects.

On the CBIR algorithm level the pointer click information is used as follows. When an image has been indicated as relevant, the leaf segment where the click location resides is resolved from the stored segmentation pixmap of the image. From the stored segment hierarchy information we can then resolve also all composite segments along the path from the leaf to the root of the tree. One can then associate the positive relevance judgement not only to the clicked leaf segment, but also to any subset of segments along that path. Once those segments of all relevant-marked images have been identified, their BMUs on the segment-wise SOMs are credited with positive relevance values and the SOM surfaces are then low-pass filtered. With 256×256-sized segment-wise feature SOMs used in the current experiments, a typical size for the radius of the low-pass filter is 14 map units. One may note that the image-level SOMs will combine both positive and negative relevance assessments given to the images, whereas the segment-level SOMs will contain only positive relevance values. As the final step, the resulting relevance scores of all image segment objects are summed to those of the corresponding image objects for determining the

Category	Object classes
Person	person
Animal	bird, cat, cow, dog, horse, sheep
Vehicle	aeroplane, bicycle, boat, bus, car, motorbike, train
Indoor	bottle, chair, dining table, potted plant, sofa, tv/monitor

Table 3: The 20 object classes of VOC Challenge 2007.

best image candidates to be presented to the user (or the simulation subsystem) as the next set of images.

4 DATA

4.1 Visual Object Classes Challenge 2007 data set

For the experiments we used a part of publicly available image collection that was compiled for the PASCAL NoE Visual Object Classes (VOC) Challenge 2007 image analysis performance evaluation campaign [2]. This image collection consists of 9963 photographic images of natural scenes. Each of the images contains at least one occurrence of the 20 object classes detailed in Table 3. The object classes are divided into categories to merely structure the table. In many images there are objects of several classes present. Altogether the images contain 24640 objects. The most common object class ("person") is present in 40% of the images, the rarest (sheep) in 1.9%. Often there are several objects of the same kind in an image, e.g. 2.3 persons and 2.7 sheep on average among the images where the named objects are present at all. Figure 5 shows some examples of the images and objects.

The VOC Challenge organisers have defined partitioning of the image collection equally into training (denoted "trainval") and test sets. Furthermore, they define the partitioning of the training half of the data evenly into two sets ("train" and "val") in order to encourage proper validation of supervised image processing methods within the training set. The experiments reported in this work, however, do not employ the supervised setting with separate training and test sets, but gradually search through a single set of images. As such image set, the "train" half of the original training set—i.e. 2501 images—will be used in the experiments. The images have been manually annotated by the VOC Challenge 2007 organisers with the bounding boxes of all the occurrences of the 20 object classes.

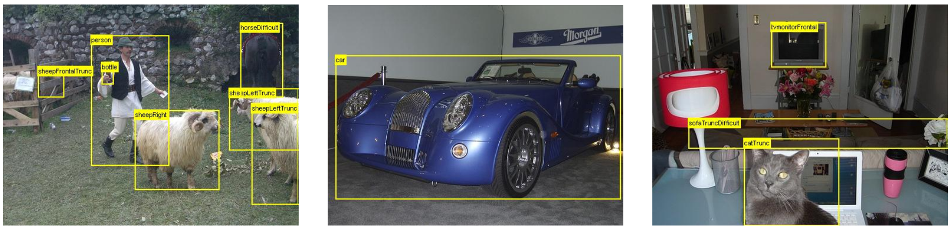

Figure 5: Examples of VOC Challenge 2007 images and their annotations

4.2 Click location data collection

For the click location data collection we have selected eight of the VOC Challenge 2007 object classes. For each class, we have identified images that display exactly one object from the class. Objects from other classes may or may not be present in

Object class	Single occurrences	Probability
bicycle	80	0.03198
bus	76	0.03038
car	217	0.08676
cat	145	0.05797
cow	30	0.01199
dog	169	0.06757
horse	106	0.04238
motorbike	88	0.03518

Table 4: Object classes selected for click location data collection along with the number of "train" set images containing exactly one class object. The rightmost column shows the *a priori* probability of the single-object images of the class.

the images. Table 4 lists the selected object classes along with numbers of the images with exactly one class object. The identified images are shown to a user and the user clicks with mouse the image location that is the most characteristic or discriminative for the class. This location information is recorded in a file.

In our setting of click location data collection, one image by time was shown to the test subject on the computer screen. Simultaneously with the pointer clicks, the user's eye movements, gaze fixation information and speech were also recorded with a Tobii eye tracker. The eight object classes were processed in four experiments, in each of which images of two separate object classes were presented. For example in the "cat versus dog" click experiment either a cat or a dog was shown. The user was asked to first explicitly shout "dog" or "cat" as soon as the content of the image was observed. Then he or she was supposed to click on the most discriminative point of the object by pressing the left mouse button once. The corresponding xy-coordinates of the click location were recorded and saved into Tobii event data (EVD) file. Similar class pairs were formed from cows versus horses, cars versus buses, and bicycles versus motorbikes. This paring of classes was not present in the off-line CBIR experiments where each of the eight object classes was studied separately.

Table 5 lists the columns created by the Tobii eye tracker in the EVD file. The eye movement, gaze fixation and click location data were later extracted from the EVD files and stored in XML-based COGAIN eye movement recording format[3] as shown in Figure 6. In addition, the click locations are stored for the PicSOM experiments in class definition files, an example which can be seen in Figure 7.

Data column	Description
Time	The timestamp in ms indicating when the click event occurred from the start of the recording
Event	LmouseButton, i.e.the left mouse button has been pressed
Event key	1, the unique number identifier for LmouseButton event
Data 1	X mouse coordinate
Data 2	Y mouse coordinate

Table 5: The columns that Tobii created in the raw event data (EVD) file.

[3] `http://www.cogain.org`

```
<?xml version="1.0"?>
<start>
  <!-- ident data -->
  <!-- setup data -->
  <image>
    <!-- gaze data -->
    <pointer>
      <clicks>
        <click>
          <timestamp>128</timestamp><number>1</number>
          <x>190</x><y>133</y>
          <buttons>left</buttons>
        </click>
      </clicks>
    </pointer>
  </image>
  <!-- keyboard data -->
  <!-- audio data -->
</start>
```

Figure 6: An example of click location data stored in COGAIN XML format.

```
# All cats found in the database with click locations.
01020304 xy=(99,56)
05060708 xy=(45,98)
22446688 xy=(100,123),(300,237)
```

Figure 7: An example of click location data stored in a PicSOM class definition file.

5 EXPERIMENTS

5.1 Simulation setup

Each of the eight object classes specified in Table 4 were analysed in a separate experiment. The task was to find images that portray exactly one object of the specified class. Consequently, for example an image with two cats in it was regarded as non-relevant or false positive in the "cat" experiment.

Each of the true positive images of the studied object class was used in its turn as an example image to initiate the content-based retrieval for finding more similar images. After the initial example had been presented to the CBIR system, the similarity-based scoring mechanism was able to select the 20 most similar ones among the 2500 remaining images of the used database as described in Sections 3.2 and 3.5. The simulation subsystem then judged the relevance of those 20 images and provided ground-truth-based relevance feedback and possibly also the stored pointer click location to the CBIR system for selecting the next set of image candidates. This simulated retrieval process thus resulted in total of 125 iterations between image selection and relevance assessment.

A set of evaluation measures was calculated for each retrieval simulation with a specific initial example image. These measures were then averaged to obtain object-class-wide performance measures. The same procedure was applied to all the eight object classes.

5.2 Evaluation measures

The results of the search experiments are evaluated with several related performance metrics. For a certain search task, each of the images is binarily judged as relevant or non-relevant. Here the search tasks are finding images with objects of certain object classes, making the relevance of individual images straightforward to judge.

For the calculation of the metrics, the images returned in the course of the search are incrementally appended to a set of result images. The result image sets form a sequence of telescoping image sets. A pair of statistics is calculated for each set in the sequence: *Recall* (R) and *Precision* (P). Recall is defined as the number of the relevant images in the set divided by the overall number of relevant images. Precision is defined as the number of relevant images in the set divided by the set size. These statistics can be interpreted as coordinates of points in two-dimensional space. By connecting the points, one can produce a *Recall–Precision* (RP) curve. The accuracy and time course of the search can be visually observed from the curve. The higher the curve goes the better the accuracy of the search.

Alternatively, one can generate the curve by taking another pair of statistics as coordinates. Often used are the *False Positive* (FP) and *True Positive* (TP) rates, leading to the *Receiver Operating Characteristic* (ROC) curve. True (false) positive rate is the number of relevant (non-relevant) images in the result set divided by total number of relevant (non-relevant) images. Higher sweeping ROC curves correspond to better search accuracy. If the total numbers of relevant and non-relevant images are known, the conversion between (R,P) and (FP,TP) statistics is straightforward.

Besides visual inspection, the RP and ROC curves can be summarised with a number of scalar statistics. Alternatively, the statistics can naturally be derived directly from the result image set statistics without a graphical interpretation. The *Area Under Curve* (AUC) and *Equal Error Rate* (EER) measures describe the ROC curve. AUC measures the area under the ROC curve. EER indicates the point of search where equal fractions of type 1 and type 2 errors have been made, i.e. the search has returned the same fraction of false alarms (of all possible ones) as it has still left true positives unrevealed. Graphically this corresponds to the point where the ROC curve intersects the diagonal TP=1-FP. We follow the practice of reporting the error rate at the EER point. Thus higher numerical values of AUC and lower values of EER correspond to a more accurate search. For a purely random search, expected value of both EER and AUC is 0.5.

The RP curve is summarised by the *Average Precision* (AP) statistic that corresponds to the area under the RP curve. The *Mean Average Precision* (MAP) measures the average AP performance over multiple search tasks. Higher numerical AP and MAP values indicate more accurate search. For completely random search, the expected value of AP would equal the *a priori* probability of a randomly selected image to be relevant for the task.

5.3 Results

We performed three sets of experiments. The first one formed the baseline for the evaluation by using only image-level relevance feedback. This setting matches that of traditional Boolean-valued relevance feedback where the user expresses that the image is either relevant or non-relevant without further specification of the most relevant position of the image. No click location data, automatic image segmentation nor segment-wise features and SOMs were thus required. Table 6 shows the results of the baseline retrieval experiment.

In the second set of experiments we used the results of automatic content-based

Class	AP	EER	AUC
bicycle	0.0701	0.3524	0.6886
bus	0.1019	0.2593	0.8118
car	0.1605	0.3268	0.7226
cat	0.1263	0.2954	0.7490
cow	0.0226	0.3863	0.6462
dog	0.1181	0.3674	0.6834
horse	0.1393	0.3489	0.7285
motorbike	0.0587	0.3837	0.6481
Mean	0.0996	0.3400	0.7097

Table 6: Baseline results with image-level relevance feedback.

image segmentation and segment-wise similarity scoring as described in Sections 3.3 and 3.5, respectively. Table 7 shows the retrieval results when the click-location-based relevance feedback has been used together with the image-level relevance feedback. The shown results correspond to the setting where the positive segment relevance has been associated with the leaf segment only, not propagating it to any composite segment along the path towards the root of the segment tree, as this policy was found to produce the best retrieval accuracy.

Class	AP	EER	AUC
bicycle	0.0570	0.3728	0.6652
bus	0.0569	0.3012	0.7608
car	0.1573	0.3102	0.7382
cat	0.1311	0.2949	0.7525
cow	0.0195	0.5020	0.5253
dog	0.1182	0.3509	0.6804
horse	0.1030	0.3328	0.7123
motorbike	0.0696	0.3741	0.6749
Mean	0.0931	0.3529	0.6912

Table 7: Retrieval results when click location relevance feedback and automatic image segmentation have been used together with image-level relevance feedback.

In the last experiments we used the quadtree division of the images, as described in the end of Section 3.3, instead of true content-based segmentation. Table 8 shows the retrieval results when the click location relevance feedback has been used together with the image-level relevance feedback. With the quadtree segments it was found that the best retrieval performance was attained when the positive relevance was associated not to the leaf segments, but to the composite or parent segment above the leaf. In practice this means that the content of the image quadrant of the click location is used together with the content of the entire image in the content-based retrieval.

When comparing the results obtained with true content-based image segmentation and click location relevance feedback in Table 7 with the baseline results in Table 6, one can clearly see that the segmentation-based results are worse with some object classes, especially with the cow class. This could be expected as in our earlier experiments with the same data we had already found out that the recognition of the cow class is more dependent on the *context* i.e. the surrounding image than the object itself. This phenomenon results from the fact that the colouring and shape of the cows vary more than those of the typically green grass environment where they appear. On the other hand, there exist some object classes, most notably the mo-

Class	AP	EER	AUC
bicycle	0.0675	0.3555	0.6903
bus	0.0761	0.2513	0.8153
car	0.1600	0.3152	0.7306
cat	0.1188	0.3013	0.7409
cow	0.0237	0.4083	0.6069
dog	0.1104	0.3618	0.6711
horse	0.1355	0.3144	0.7410
motorbike	0.0698	0.3468	0.6778
Mean	0.0952	0.3318	0.7092

Table 8: Retrieval results when click location relevance feedback and quadtree image segments have been used together with image-level relevance feedback.

torbike class, where all the evaluation measures are better when the click location relevance feedback is used.

On the contrary, the use of the simple quadtree image splitting method in Table 8 seems to improve the retrieval accuracy for some object classes when compared to that of the baseline. When this outcome is combined with the worse performance with the true image-based segmentations, a proper explanation for the failure of the latter might be the weakness of the automatic image segmentation. Of course, many of the versatile images of the studied object classes in the VOC Challenge 2007 collection are difficult for correct automatic segmentation.

Overall it seems to happen quite often that while some of the evaluation measures is improved another one gets worse. This makes the analysis of the results a bit problematic and hints to the direction that the differences might not be very significant. Nevertheless, if some method parameters would be selected by optimising the value of one of the measures, most likely the outcome would be suboptimal with respect to the other evaluation measures. Figure 8 displays a selection of ROC and RP curves obtained in the experiments. It can be seen that the RP curves seem to make somewhat larger differences between the performances of the three compared methods than he ROC curves.

6 CONCLUSIONS

The aim of the reported study was to evaluate the usefulness of the existing scheme for using automatic image segmentation and click location relevance feedback in the PicSOM CBIR system with the VOC Challenge 2007 data set and the retrieval tasks specified for the data collection in the PinView project. The used techniques and image features were based on those implemented previously in the PicSOM system. No substantial algorithmic development was made during this study which we regarded merely as an initial evaluation and baseline setting for further studies in the PinView project and our own further improvement of the PicSOM system.

Overall the results were clearly worse than hypothesised, and disappointing. To some extent this can be explained by the difficultness of some of the images in the VOC Challenge 2007 collection, which results in failures in automatic image segmentation. In some cases the objects of interest are so small or only partially visible that the image segmentation is unable to identify them. In these cases the context of the object is a more valuable source of information than the object itself, but the use of the surrounding of the click location neglects that. In addition to difficulties in segmentation, the role of object context is also determined by the relative descriptive

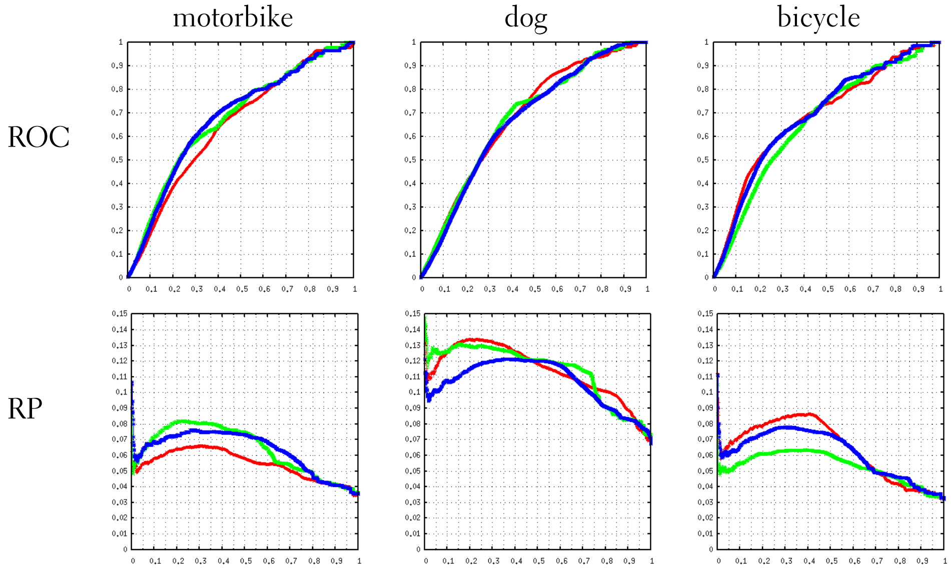

Figure 8: ROC and RP curves of the experiments with the *motorbike*, *dog* and *bicycle* image classes. The red curves represent the baseline with only image-level relevance feedback, the green curves experiments using also true content-based image segments, and the blue curves those using also quadtree image partitionings.

powers of the segment and whole image visual features. In the reported experiments, the set of segment features was not as powerful as the set of whole image features. This further emphasises the importance of the context.

A factor that should be taken into account when interpreting the results is the incompability of the queries and the simple within-image mechanism for aggregating segment-wise scores into the image-wise score. The queries required exactly one object to be present, whereas the relevance aggregation mechanism generally would sum together relevance scores of multiple objects, i.e. two cats would be regarded twice as relevant as one. The exact significance of this phenomenon to the results can only be speculated upon. In any case, this makes the results somewhat inconclusive and uncertain.

There exist clearly a line for further development. In future experiments we seek to employ enriched relevance feedback that would consist of more than just the click location. For example if a gaze pattern, rectangular bounding box or a more freehand sketch of the object outline could be passed to the search engine, the need for relying on the insecure outcome of automatic image segmentation could be relieved. This approach, however would prevent one from using precalculated features and image indices such as those based on the Self-Organizing Maps used in the PicSOM system. Another problem of this plan is that even though one would then have features that are more accurate in describing the relevant object, the bulk of images in the image database could still be efficiently described and retrieved only by their entire content. This would mean that relevant, but small objects would most likely remain unfound. Some image collections, such as that of the VOC Challenge 2007, do contain bounding-box information for some object classes, but such image sets are rare and even with them it would be possible to retrieve only objects belonging to those predefined classes.

Despite the difficulties encountered in this evaluation and those discussed above, we will continue the development of our PicSOM CBIR system for being used together with true eye movement and gaze fixation pattern information in the PinView project. We believe that the additional information available in the gaze measurements will help us to reveal the location and approximate area of the relevant objects in the images. When such recordings are stored in the search engine's database it will become possible to compare the current gaze pattern and the visual features extracted from its location to those of the previous retrieval sessions. This in turn would improve the retrieval accuracy of such objects in images that some other users have retrieved earlier, leading to a form of collaborative filtering in gaze-enhanced content-based image retrieval.

ACKNOWLEDGEMENTS

We wish to thank Teofilo de Campos of XEROX for his valuable comments on the draft versions of this report and He Zhang and Lu Wei of TKK for their contributions in the collection of the click location data.

References

[1] N.-S. Chang and K.-S. Fu. Query-by-Pictorial-Example. *IEEE Transactions on Software Engineering*, 6(6):519–524, November 1980.

[2] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html.

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

[4] Teuvo Kohonen. *Self-Organizing Maps*, volume 30 of *Springer Series in Information Sciences*. Springer-Verlag, Berlin, third edition, 2001.

[5] Markus Koskela, Jorma Laaksonen, Mats Sjöberg, and Hannes Muurinen. PicSOM experiments in TRECVID 2005. In *Proceedings of the TRECVID 2005 Workshop*, pages 262–270, Gaithersburg, MD, USA, November 2005. Available online at `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`.

[6] Markus Koskela, Mats Sjöberg, Ville Viitaniemi, Jorma Laaksonen, and Philip Prentis. PicSOM experiments in TRECVID 2007. In *Proceedings of the TRECVID 2007 Workshop*, Gaithersburg, MD, USA, November 2007. Available online at `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`.

[7] 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.

[8] 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.

[9] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantizer design. *IEEE Transactions on Communications*, 28(1):84–95, January 1980.

[10] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[11] K. Mikolajcyk and C. Schmid. Scale and affine point invariant interest point detectors. *International Journal of Computer Vision*, 60(1):68–86, 2004.

[12] Yong Rui, Thomas S. Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: A power tool in interactive content-based image retrieval. *IEEE Transactions on Circuits and Systems for Video Technology*, 8(5):644–655, September 1998.

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

[14] Mats Sjöberg and Jorma 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)*, pages 841–846, Warsaw, Poland, September 2005. Available online at `http://dx.doi.org/10.1007/11550907_133`.

[15] Mats Sjöberg, Jorma Laaksonen, and Ville Viitaniemi. Using image segments in PicSOM CBIR system. In *Proceedings of 13th Scandinavian Conference on Image Analysis (SCIA 2003)*, volume 2749, pages 1106–1113, Halmstad, Sweden, June/July 2003. Springer Verlag. Available online at `http://www.springerlink.com/content/65xgpgqc9cjquqcl/`.

[16] Mats Sjöberg, Hannes Muurinen, Jorma Laaksonen, and Markus Koskela. PicSOM experiments in TRECVID 2006. In *Proceedings of the TRECVID 2006 Workshop*, Gaithersburg, MD, USA, November 2006. Available online at `http://www-nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html`.

[17] Markus Stricker and Markus Orengo. Similarity of color images. In *Storage and Retrieval for Image and Video Databases III (SPIE)*, volume 2420 of *SPIE Proceedings Series*, pages 381–392, San Jose, CA, USA, February 1995.

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

[19] Ville Viitaniemi and Jorma Laaksonen. Improving the accuracy of global feature fusion based image categorisation. In Bianca Falcidieno, Michela Spagnuolo, Yannis S. Avrithis, Ioannis Kompatsiaris, and Paul Buitelaar, editors, *Proceedings of the 2nd International Conference on Semantic and Digital Media Technologies (SAMT 2007)*, volume 4669 of *Lecture Notes in Computer Science*, pages 1–14, Genova, Italy, December 2007. Springer.

[20] Ville Viitaniemi and Jorma Laaksonen. Experiments on selection of codebooks for local image feature histograms. In Monica Sebillo, Giuliana Vitiello, and Gerald Schaefer, editors, *Proceedings of the 10th International Conference on Visual Information Systems (VISUAL 2008)*, pages 126–137, Salerno, Italy, September 2008. LNCS, Springer.

TKK REPORTS IN INFORMATION AND COMPUTER SCIENCE

TKK-ICS-R1	Nikolaj Tatti, Hannes Heikinheimo Decomposable Families of Itemsets. May 2008.
TKK-ICS-R2	Ville Viitaniemi, Jorma Laaksonen Evaluation of Techniques for Image Classification, Object Detection and Object Segmentation. June 2008.
TKK-ICS-R3	Jussi Lahtinen Model Checking Timed Safety Instrumented Systems. June 2008.
TKK-ICS-R4	Jani Lampinen Interface Specification Methods for Software Components. June 2008.
TKK-ICS-R5	Matti Koskimies Applying Model Checking to Analysing Safety Instrumented Systems. June 2008.
TKK-ICS-R6	Alexander Ilin, Tapani Raiko Practical Approaches to Principal Component Analysis in the Presence of Missing Values. June 2008.
TKK-ICS-R7	Kai Puolamäki, Samuel Kaski Bayesian Solutions to the Label Switching Problem. June 2008.
TKK-ICS-R8	Abhishek Tripathi, Arto Klami, Samuel Kaski Using Dependencies to Pair Samples for Multi-View Learning. October 2008.
TKK-ICS-R9	Elia Liitiäinen, Francesco Corona, Amaury Lendasse A Boundary Corrected Expansion of the Moments of Nearest Neighbor Distributions. October 2008.
TKK-ICS-R10	He Zhang, Markus Koskela, Jorma Laaksonen Report on forms of enriched relevance feedback. November 2008.

ISBN 978-951-22-9671-2 (Print)
ISBN 978-951-22-9672-9 (Online)
ISSN 1797-5034 (Print)
ISSN 1797-5042 (Online)