# GalSOM - Colour-Based Image Browsing and Retrieval with Tree-Structured Self-Organising Maps

Philip Prentis*
Department of Mathematics
Faculty of Nuclear Sciences and Physical Engineering
Czech Technical University in Prague
Trojanova 13, 120 00 Prague 2, Czech Republic
email: prentisp@km1.fjfi.cvut.cz

Keywords: image browsing, content-based image retrieval (CBIR), colour histograms
tree-structured self-organising maps (TS-SOM), image theft detection

***Abstract*—** This paper describes an image browsing and retrieval application called GalSOM. Bitmap images are described by their colour histograms and sorted using an improved variant of the tree-structured self-organising map (TS-SOM) algorithm. The advantages of using such a system are discussed in detail, and their application to the problem of image theft detection is proposed.

## 1 Introduction

Today's widespread availability of digital cameras, art packages and publicly available collections of downloadable artwork and photos have resulted in many users accumulating large collections of bitmap images on their home computers. These collections or 'galleries' of images will usually be organised in a directory structure to help the user locate images at a later date, but due to the eclectic manner of their creation, they may easily become confusing to navigate, especially if insufficient effort has been made to categorise and/or label the images. Users are faced with two basic tasks: image browsing and image retrieval.

When browsing, a user is presented with images, either in series (one at a time), or in parallel (many images at once). Typically, common image browsers such as Irfanview[1] present the images in series, after having ordered them using meta-data such as filename, extension, date, etc.

For parallel browsing it is usually necessary to scale down the images to a small standardised size; these scaled down images are called 'thumbnails'. Most modern operating systems allow folders of picture files to be browsed in this manner. Browsing multiple folders of images at once can be achieved by using *quantum tree maps* and *zoomable browsing* [3] with applications such as PhotoMesa[2]. It is also possible to use more advanced methods such as $NN^k$ networks [4], or hyperbolic image viewers that display the images on a *Poincaré map* [5].

Image retrieval consists of locating a desired image or set of images within a gallery. There are many methods of achieving this [6], which may include standard file-search methods such as searching for files or folders of a specific name or date, using image annotation and keywords, or by extracting feature vectors describing image content.

The task of *content-based image retrieval* (CBIR) is a complex one. We are faced with the problem of locating an image in a database using a simplified or inaccurate description of its content, which is usually based on user input. This can include locating a specific picture the contents of which is known to the user, or of finding all pictures that fit a given description.

Descriptions can contain both high order and low order information. For example, if we take a holiday photo, then a human might describe it as a sandy beach with deep blue water and sky. The information that the picture contains *a beach*, *water* and *sky* is high order and may be difficult to obtain automatically without the use of meta-data (annotation/keywords), while the dominant colours and their respective quantities are low order and may be easily obtained.

There are a number of systems available for performing CBIR, such as QBIC [9], Photobook [10] and NeTra [11], which work with various low-level features such as colour, texture, shape and spatial information and allow different variations of query-by-example retrieval [7]. The PicSOM system [15, 16] also incorporates relevance feedback [14]. It is an important example, because it is similar to the application described in this paper in that it also uses the 'tree-structured self-organising map' algorithm described in section 3.2.

## 2 GalSOM

### 2.1 Motivation

We designed the user application GalSOM to solve the following problem: in a database or 'gallery' of RGB-coloured bitmap images such as digital photos or artwork,

*Currently visiting the Laboratory of Computer and Information Science, Helsinki University of Technology.
[1] http://www.irfanview.com
[2] http://www.cs.umd.edu/hcil/photomesa/

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

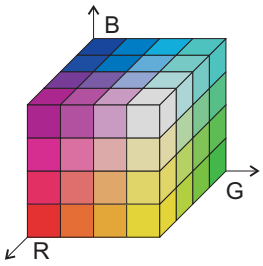


Figure 1: RGB colour space divided into bins representing different sets of colours.

locate an image based on the user's memory of its appearance. Rather than attempt to interpret high-order queries, we chose to use a low-order description as the basis of our retrieval method. The most obvious and intuitive low-order descriptor is colour content. It was necessary to find colour features that would mimic the human description (i.e. "This picture contains a lot of light blue, some dark blue and large areas of sandy yellow").

## 2.2 Colour Histograms

The colour histogram is the mathematical equivalent of the afore-mentioned type of description. To acquire a colour histogram of an RGB-coded bitmap we simply count the number of pixels of each shade of colour present in the picture, creating an $N$-dimensional feature vector, where $N$ is the number of shades of colour in the palette being used. Typically, bitmaps are coded using a palette of $256^3$ (16 million) colours. To produce meaningful histograms, it is necessary to reduce the palette to few enough different shades that they may be *visibly* distinguished from one another. This can be achieved by segmenting 3-dimensional colour space (RGB) into bins that represent different sets of colours. See figure 1.

Too many bins results in similar pictures being classified as different, while too few groups dissimilar ones together. Determining the optimal number of colour bins is highly subjective, because there is no objective measure of which colours a user will find similar. After experimenting with different-sized palettes, 64 colours was determined to be the optimal number, assuming we limit ourselves to cubes of $2^x, x = 1..8$, which guarantees that RGB colour space ($(2^8)^3$ colours) will be divided into bins of equal size. See figure 2.

One problem with RGB colour space is the lack of perceptual uniformity. This could be solved by transforming the image to a different colour systems such as HVS, which is perceptually more uniform than RGB [8]. However, using RGB is faster as the input data is coded in it by default. For this reason it was used in GalSOM.

Colour histograms are invariant to rotation and mirror-imaging. If normalised, they are also scale-invariant.

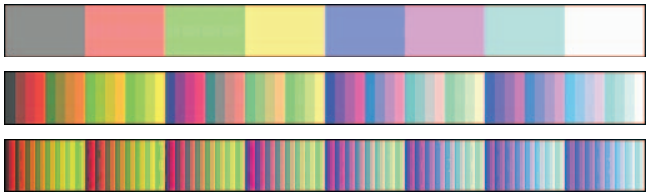

Figure 2: Colour palettes of 8, 64 and 512 colours. These are all cubes of 2, 4 and 8 respectively, which guarantees they will divide colour space ($256^3$ colours) into bins of equal size.

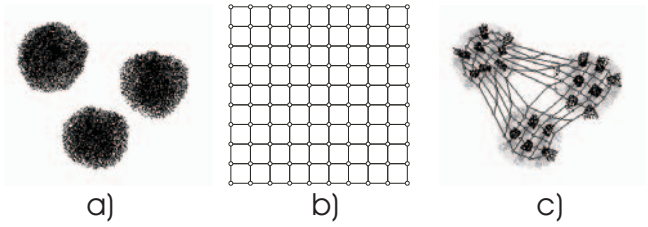


Figure 3: a) 2-dimensional input vectors in 3 clusters. b) A SOM, topologically ordered in a grid of $10 \times 10$ neurons c) The SOM adapts itself to match the input space, each neuron's codebook vector quantifying a set of inputs. GalSOM uses 64-dimensional input vectors.

## 2.3 Visualisation with SOM

GalSOM uses the iterative self-organising map (SOM) algorithm described in [2]. A SOM with a 2D topology will map a given input space of N-dimensional feature vectors to a 2D grid. Each map neuron is associated with a "codebook" vector located in input space. The SOM algorithm results with these vectors being spread more or less evenly through input space, approximating the probability distribution of the input vectors, while preserving their topological relationships (i.e. the codebook vectors will be positioned closer to the other codebook vectors that are their topological neighbours). Finally, input space is divided into regions defined by the closest codebook vector using vector quantisation. See figure 3.

The next step is to visualise the grid (or matrix) of codebook vectors. GalSOM does this by creating a colour map, using the mean colour values of the colour histograms described by the codebook vectors. See figure 4 a). Each node is associated with a list of input files for which its codebook vector is the closest to their feature vectors. Due to their close proximity in input space, these pictures will be of similar appearance, containing similar colour distributions (figure 4 b)).

The colour map described above is particularly useful for locating images with a single dominant colour as the mean value of their colour histogram will tend towards it. This allows fast location of images based solely on the memory of their appearance. However, pictures containing a variety of colours cannot be easily distinguished by their mean

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

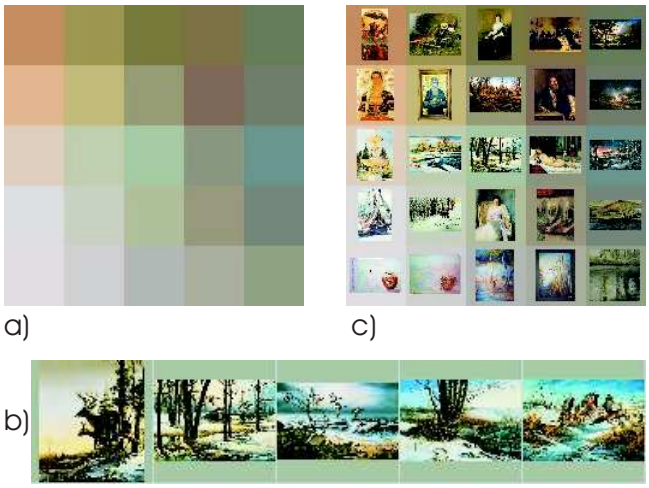


Figure 4: a) A $5 \times 5$ SOM visualising a set of bitmap images characterised by colour-histogram vectors. Each square is coloured in the mean colour of the codebook vector's histogram. b) Thumbnails of the images quantified by a selected node, in this case the centre node. Each node in the map quantifies a small easily-browsable number of similar images. c) Thumbnails of closest images may be used to describe the codebook vectors.

colour. In this case, we may get a better representation of the node's codebook vector by using a "thumbnail" picture of the input file closest to it. See figure 4 c).

### 2.4 Query by Image

An alternate means of using GalSOM is to use a bitmap image as a query. GalSOM takes the image, calculates its colour histogram and locates the best-matching neuron. Images of similar appearance to the query will be located at that map node. This can be useful if the query picture is part of a set of similarly themed images, or if we are looking for similar images to go with a given colour combination for design purposes.

### 2.5 Optimal Size of SOM

GalSOM works by taking input space and dividing it into "chunks" of similar data associated with the nodes of a map. The parameters of the SOM algorithm have been optimised primarily towards making those chunks of as equal a size as possible, while at the same time minimising the *average quantisation error* (determined by the average distance of an input from its closest neuron) and where possible, preserving the topology of the SOM. Each node will quantify on average approximately $A$ inputs:

$$A =_{\text{def}} I/N, \tag{1}$$

where $I$ is the number of inputs and $N$ is the number of nodes in the map. The optimal map size will be one where $A$ is the number of images that may be comfortably displayed (in thumbnail format) and browsed at once.

However, there is no guarantee that all nodes will quantify exactly $A$ images. Typically, there will be a few nodes centred in tight clusters that quantify a much higher number of images, while some nodes may end up quantifying few or no images. These things should be taken into consideration when determining the map size. A method of circumventing the problem of unevenly mapped input spaces using multiple mappings is discussed in section 3.4.

It should be noted that larger mappings (i.e. ones with more neurons than there are inputs) can be used for alternate visualisation methods, currently not fully implemented in GalSOM. These methods rely on the emergent properties of the SOM [17], which use so-called U-matrices to locate clusters in input space [18, 19]. Because various sizes of map may be needed to acquire various results, GalSOM offers multi-resolution mappings using the "tree-structured" SOM variant described in section 3.

### 2.6 Comparison to other Systems

The greatest difference between GalSOM and the other CBIR systems described in section 1, is that *it is limited to a single intuitively understandable feature* — the colour histogram. This affords a degree of transparency that allows it to be used effectively as a tool. Also, limiting the system to a single feature type allows for the development of customised outputs such as the colour maps shown in figure 4.

In contrast, PicSOM [16] — which is one of the most similar CBIR systems — uses multiple features in an attempt to describe high-order information. Its query-by-image interface relies on correlation between high-order information (the query) and low-order (the features being used) to be successful. The use of multiple, often complicated features makes the system a black box to users, as the connection between the SOM feature mappings and the higher-order information being queried is generally unapparent.

## 3 TS-SOM

The tree-structured self-organising map (TS-SOM) [12, 13] is a hierarchical structure of SOM of exponentially increasing size. Each level of the TS-SOM adapts separately, but in the lower levels, the search for the best-matching neuron (BMN) is limited to those hierarchically connected to the BMN of the previous layer. Each layer has double the dimensions of the previous one, that is 4 times as many neurons. Hierarchical connections lead from each neuron on level $l$, $n_{ij}^{(l)}$, to the four neurons immediately beneath it on level $l+1$: $n_{2i,2j}^{(l+1)}$, $n_{2i-1,2j}^{(l+1)}$, $n_{2i,2j-1}^{(l+1)}$ and $n_{2i-1,2j-1}^{(l+1)}$. See figure 5.

Proceedings of the 6th International Workshop on Self-Organizing Maps (WSOM 2007)
Published by the Neuroinformatics Group, Bielefeld University, Germany, ISBN 978-3-00-022473-7
All contributions to WSOM 2007 are available online at: http://biecoll.ub.uni-bielefeld.de

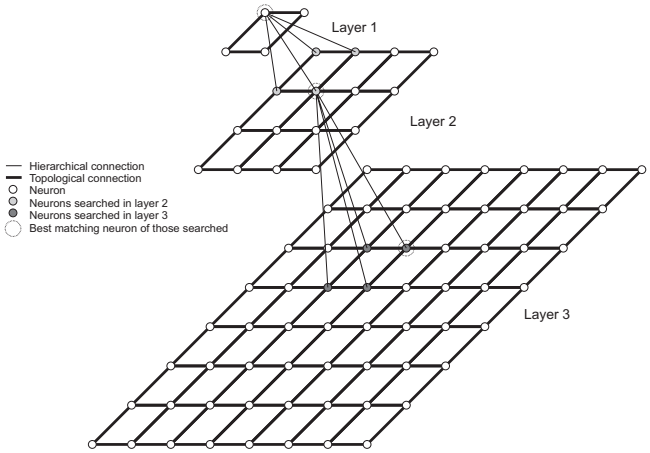


Figure 5: A 3-layer TS-SOM with 4 neurons at top layer and 64 at the bottom.

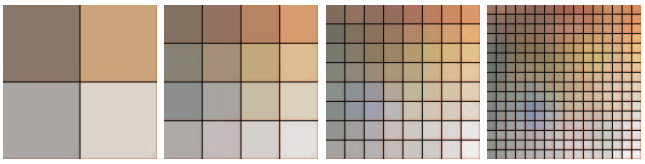

Figure 6: 4 layers of a TS-SOM show colour distribution of an input space of images at different resolutions.

## 3.1 The Basic Algorithm

The algorithm works as follows:

(1) Perform one iteration of the SOM algorithm on the top layer.
(2) Perform one iteration of the SOM algorithm on the next layer, but restrict the search for the BMN to the neurons located under the winning neuron of the previous layer.
(3) Repeat 2 until all layers have been updated.
(4) Repeat 1 to 3 until the adaptation process is complete.

The advantages of this structure are obvious. Instead of performing a full-search for the best matching neuron at the lower layers, which would be time-consuming given their size, we restrict ourselves to a constant number of neurons per a given layer, thus greatly increasing the adaptation speed. Also, due to the hierarchical structuring, all the SOMs will be orientated similarly in input space and the TS-SOM as a whole may be considered a multi-resolution mapping of the given data set. See figure 6.

## 3.2 Wide-Search TS-SOM

One unfortunate property of the TS-SOM is the propagation of errors to the lower layers. Inputs bordering between two neurons on a higher layer may gradually become more and more poorly quantified as the search for the BMN becomes more and more restricted. During our experiments we found that on the lower layers, inputs had a tendency to group together rather than spread themselves evenly over the map as is typical in well-tuned SOM. As noted in [14], better results may be achieved by allowing searching for the BMN in a wider scope, which includes neurons adjacent to those directly under a higher layer (figure 7).

## 3.3 Multi-Resolution Correction

The unfortunate side effect of the wide-search improvement of the TS-SOM algorithm is that the separate layers become desynchronised. This degrades the quality of the TS-SOM as a multi-resolution mapping. In [1] we presented the following simple yet effective solution, which we call *multi-resolution correction* (MRC).

### 3.3.1 The Algorithm

(1) Adapt the TS-SOM using the algorithm described in 3.1 (or alternatively using the wide-search modification described in 3.2).
(2) Replace all neurons above the lowest layer using this formula:

$$m_{ij}^{(l)} = \frac{\sum_{a=a_1}^{a_2} \sum_{b=b_1}^{b_2} m_{ab}^{(L)}}{4^{L-l}} \qquad (2)$$

$$\forall i,j : i = 1 \dots 2^l, j = 1 \dots 2^l,$$
$$l = 1 \dots L$$

where

$$a_1 = 2^{L-l}(i-1)+1, \qquad a_2 = 2^{L-l}i,$$
$$b_1 = 2^{L-l}(j-1)+1, \qquad b_2 = 2^{L-l}j,$$

and $m_{ij}^{(l)}$ is the new position of the neuron topologically placed at coordinates $i, j$ on level $l$. $L$ is the index of the lowest level. Levels are indexed from 1 (highest) to $L$ (lowest).

This converts the higher levels into perfect lower resolution images of the lowest level. Each neuron becomes the arithmetic mean of the lowest level neurons hierarchically connected to it.

### 3.3.2 Benefits of MRC

Synchronising the separate layers of the TS-SOM greatly improves the efficiency of the search for the BMN on the lower layers, especially when coupled with wide-searching. This can be measured by calculating the percentage of input vectors that are correctly classified by the tree-search algorithm on a given level (i.e. the BMN produced by the algorithm is the same as the full-search BMN on that level).

In the example shown in table 1 we can see that applying MRC to a 7-level TS-SOM substantially increased the

| Tree search quality [%] | | |
|---|---|---|
| Level | TS-SOM | TS-SOM+MRC |
| 1 | 100 | 100 |
| 2 | 98.5 | 98.3 |
| 3 | 95.4 | 94.6 |
| 4 | 88.4 | 92.7 |
| 5 | 81.5 | 91.2 |
| 6 | 74.6 | 90.3 |
| 7 | 65.9 | 87.4 |

Table 1: This example shows how multi-resolution correction increases the TSQ on the lowest levels of a 7-level TS-SOM.

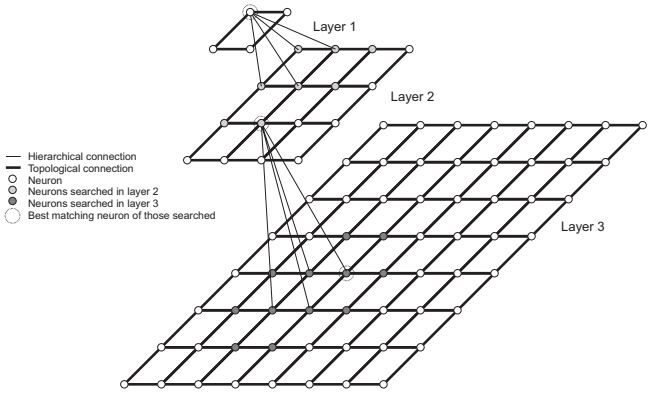


Figure 7: Wide-search for the BMN in neighbouring neurons.

tree-search quality (TSQ) on the lowest level from 65.9% to 87.4% at the cost of fractionally decreasing the scores on the highest levels. This fractional difference can be disregarded, because most applications (including GalSOM) make use of the TS-SOM for creating large fast mappings, which are represented by the lower levels.

Increased TSQ translates into greater accuracy, which in turn improves all aspects of the mapping including measures such as *average quantisation error*. For more details and discussion of these results, see [1].

### 3.4 Image Retrieval with TS-SOM

When querying a TS-SOM, we are searching for the BMN on each level. A TS-SOM with $L$ levels will return $L$ nodes associated with $L$ different sets of images $I^{c_i}$ ($c_i$ is the BMN on level $i$, $i = 1 \dots L$). Because the average number of inputs per node $A$ (see equ. (1)) decreases with each level, we may expect $|I^{c_i}|$ to decrease also (although it is not guaranteed). Therefore, we are presented with a list of set sizes from which we may select the node with the optimal number of images for our purpose (e.g. the one which displays the highest number of images that can be displayed onscreen at once).

## 4 Development

Although GalSOM was developed for a specific application, it can easily be expanded to solve other tasks. In particular, we are currently examining its applicability to the task of *image theft detection*.

### 4.1 Image Theft Detection

Image databases are often run on a commercial basis. Clients may browse images for free, but are required to pay a fee before they may use them for their own purposes, e.g. on their websites. The companies are faced with the problem of protecting themselves from copyright theft by users who take the images, modify them and reuse them without permission. *Image theft detection* (ITD) differs from standard CBIR in that instead of searching for all similar images to a given description, we are searching for all *equivalent* images to the query. Images are considered equivalent if one has been derived from the other by *quality reduction*, *radiometric or colour distortions*, *cropping*, *local changes* (such as added logos) or combinations of these.

The cutting edge methods currently under development at the *Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic* [20] preprocess the images using *invariant picture regions*. I.e., they detect areas of the picture that should remain stable even after the afore-mentioned transformations. The preprocessed images are then classified with the aid of a binary decision tree using stochastic indexing.

### 4.2 ITD with GalSOM

Even without modifications, GalSOM can be used for ITD. When a user finds a suspicious image, he or she queries it with GalSOM. The TS-SOM multi-level mapping returns a screen full of thumbnails taken from the BMN from the level that returns the optimal number of pictures (as described in section 3.4), which allows the user to tell at a glance, whether or not the image is present.

Colour histograms are invariant to scaling, rotation and mirroring, so pictures transformed in this manner will be detected by it. Blurring will affect the histogram, though. Preliminary experiments have shown that the small logos (2% of image size) that are typically found on stolen images do not significantly affect the histogram, so these will be found too. However, radiometric or colour distortions such as changes to brightness, contrast or colour hue will fundamentally change the picture's colour distribution, so they are not detected. Cropping is also problematic, unless the picture is largely homogenous.

### 4.3 Combining Methods

To make GalSOM an effective tool for ITD, it will be necessary to implement different feature vectors than the ones

currently in use (i.e. colour histograms). The invariant picture regions used in [20] could be used as a basis for calculating features invariant to cropping, while replacing colour histograms with features defined by point intensity within the regions should improve invariance to radiometric and colour distortions. It will then be possible to compare the TS-SOM (used as a classifier) with the binary trees currently in use.

## 5 Summary and Conclusions

In this paper we have described a content-based image browsing and retrieval application called GalSOM, which uses an improved variant of tree-structured self-organising maps to sort and visualise 'galleries' of bitmap images described by their colour histograms. While GalSOM was developed as a solution to a specific problem, the techniques it uses could easily be used with other forms of data than digital graphics. With some modifications, the system has promise as a means of image theft detection.

The main benefit of this kind of application is that it describes the data with a single intuitively understandable feature type that facilitates the development of customised visualisation techniques, and makes it easy for untrained users to use.

## References


[1] P. Prentis, "Multi-Resolution Visualization of Data with Self-Organizing Maps". *Neural Network World*, volume 16-5. 2006.

[2] T. Kohonen, *Self-Organizing Maps*, Third Edition, Springer-Verlag Berlin 2001.

[3] B. Bederson "PhotoMesa: A Zoomable Image Browser Using Quantum Treemaps and Bubblemaps", in proc. of UIST 2001, ACM Symposium on User Interface Software and Technology, *CHI Letters*, 3(2), pp. 71-80. 2001.

[4] D. Heesch, S. Rger, "Image Browsing: Semantic Analysis of NNk Networks". International Conference on Image and Video Retrieval, Singapore. 2005.

[5] J. Walter, D. Wessling, K. Essig, and H. Ritter, "Interactive hyperbolic image browsing - towards an integrated multimedia navigator". In *ACM MDM/KDD Multimedia Data Mining and Conf Knowledge Discovery and Data Mining*, Philadelphia, USA. August 2006.

[6] Y. Rui T.S. Huang and S.F. Chang, Image Retrieval: Current Techniques, Promising Directions, and Open Issues, *J. Visual Comm. and Image Representation*, vol. 10, no. 1, pp. 39-62. 1999.

[7] C. C. Yang, "Content-Based Image Retrieval: A Comparison between Query by Example and Image Browsing Map Approaches", *Journal of Information Science*, Vol. 30, No. 3, 254-267, 2004.

[8] J. R. Smith and S. Chang, "Single Color Extraction and Image Query", in Proc. of International Conference on Image Processing (ICIP-95), Washington, DC, Oct. 1995.

[9] M. Flickner, H. Sawhney, W. Niblack, "Query by image and video content: the QBIC system", *Computer* Volume 28, Issue 9, Pages: 23-32. Sep 1995.

[10] A. Pentland, W. Picard, S. Sclaroff, "Photobook: Content-Based Manipulation of Image Databases", In *SPIE Storage and Retrieval for Image and Video Databases* II, number 2185, San Jose, CA. Feb. 1994.

[11] W. Y. Ma and B. S. Manjunath, "NETRA: A toolbox for navigating large image databases", in IEEE International Conference on Image Processing, 1997.

[12] P. Koikkalainen, E. Oja, "Self-organizing hierarchical feature maps". Proceedings International Joint Conference on Neural Networks; II:279-284, San Diego, CA, 1990.

[13] P. Koikkalainen, "Progress with the tree-structured self-organizing map". 11th European Conference on Artificial Intelligence, August 1994.

[14] 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*, 4(2-3): 140-152, June 2001.

[15] J. Laaksonen, M. Koskela and E. Oja, "Application of Self-Organizing Maps in Content Based Image Retrieval". in Proceedings of ICANN'99, Edinburgh, 1999.

[16] J. Laaksonen, M. Koskela, S. Laakso and E. Oja, "PicSOM - content-based image retrieval with self-organizing maps". *Pattern Recognition Letters* 21, 2000.

[17] A. Ultsch, "Data Mining and Knowledge Discovery with Emergent Self-Organizing Feature Maps for Multivariate Time Series", In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 33–45, Elsevier. Amsterdam, 1999.

[18] A. Ultsch, "Clustering with SOM: U*C", in Proceedings of the 5th Workshop on Self-Organzing Maps, Paris 2005.

[19] F. Moutarde, A. Ultsch, "U*F Clustering: A New Performant "Cluster-Mining" Method Based On Segmentation of Self-Organizing Maps", in Proceedings of the 5th Workshop on Self-Organzing Maps, Paris 2005.

[20] O. Horáček, J. Kamenický and J. Flusser, "Image Retrieval for Image Theft Detection", in Proceedings of the fifth International Conference on Computer Recognition Systems, Warsaw 2007.