AUTOMATIC IMAGE SORTING USING MPEG-7 DESCRIPTORS

Kai Uwe Barthel, Steve Mammani, Nicholas Wyatt

FHTW Berlin, Internationaler Studiengang Medieninformatik
Treskowallee 8, 10313 Berlin, Germany
barthel@fhtw-berlin.de

ABSTRACT

The increasing use of digital images has led to a growing problem how to organize these images in such a way that particular images can easily be retrieved or searched. Most images usually do not contain any meta-data that can be used for image retrieval or sorting. Content-Based Image Retrieval (CBIR) systems use the query-by-example principle, which suffers from a number of limitations. If no query image is available, this technique cannot be used. In addition, useful sorting of images is not possible with CBIR systems.

This paper presents an automatic image sorting technique that can help to organize large sets of images. Image retrieval can be accomplished by successively narrowing down the set of images by performing a consecutive image sorting and selection of image sub-sets. Our automatic image sorting system uses MPEG-7 visual features which are organized by a self-organizing map.

1. INTRODUCTION

Over the last decade the amount of digital images has increased tremendously. In order to make use of these images, efficient methods for archiving, organising, searching and retrieving had to be developed. Initial approaches in image retrieval were an extension of existing text-based search technologies that used a keyword-based scenario. In this approach, all images need to be annotated manually before being stored; they can be retrieved by searching for corresponding keywords.

To avoid the manual annotation (the addition of content based meta-data) and to automate the process of image retrieval, Content-Based Image Retrieval (CBIR) techniques have been developed since the early 1990s. CBIR systems use low-level features (mostly color, texture, and shape) to represent an image’s content.

During the search process images containing similar features are retrieved. However, despite all the research efforts, the results of CBIR systems have not reached the performance of text based search engines. There are two main reasons for this inefficiency: There is an important (semantic) gap between the low level features and high level semantic feature concepts of image contents that are up to now only understandable by humans. In addition the human perception of images may differ for the same visual content.

Another problem consists in finding useful sorting techniques for huge collections of images. Most imaging programs do allow a sorting of images by size, creation date or image format. Usually none of these sorting criteria leads to a “useful” visual list or arrangement of the images.

Although it is not sure if the problems of current CBIR systems will be solved in the near future, the technique of automatic low-level feature extraction can be used to organize and sort image sets. Assuming a technique that could provide a sorted arrangement of images, which humans would consider useful, is found, larger collections of images could easier be conceived. This also would allow an easier navigation within huge image data sets.

In our paper we propose a technique that allows an automatics sorting of large image sets. Our approach uses MPEG-7 low-level feature vectors that are mapped to lower dimensions by the use of a self-organising map (SOM) (Kohonen map). These 2D- or 3D-maps can then be used for navigation within the large image sets. Image retrieval can be accomplished by successively narrowing down the set of images by performing a consecutive image sorting and selection of image sub-sets.

The rest of this paper is organized as follows: Section 2 recalls the main concept of the MPEG-7 visual feature extraction. Section 3 describes the basic data processing steps for constructing self-organizing maps. In section 4 our algorithm to generate sorted image maps is proposed.
2. MPEG-7 VISUAL FEATURES

The visual part of the MPEG-7 standard defines several descriptors [1, 5, 6]. For still images these descriptors describe color (Color Layout, Color Structure, Dominant Color, Scalable Color), texture (Edge Histogram, Homogeneous Texture, Texture Browsing), and shape (Region-based Shape, Contour-based Shape).

Descriptor algorithms describe how to extract feature vectors from images. This feature vector extraction achieves an important reduction of the dimensionality. Images having millions of pixels are described by several feature vectors each with dimensions in the range from some tens to a few hundreds. For operations like retrieval, browsing, etc., the feature vectors are interpreted as points in a metric space [2]. In order to be able to measure similarities between different images, the distances between these points have to be determined.

CBIR systems usually use a query image for searching similar images. The feature vectors of this image are determined and then matched against the feature vectors of the image set that is to be searched. Figure 1 shows the basic scheme of a CBIR system. Despite intense research efforts, there are still several unsolved problems with content based image retrieval systems:

- The search for particular images is difficult if no query image is available. Some approaches do use manually drawn sketches. However, most users are not skilled to draw appropriate sketches; in addition to the complicated usage and the slow execution time of such a system, statistics of these sketched images do differ significantly compared to those of “real” images.
- The visual features do only represent low-level metadata of the images. Due to this users might find some of the retrieved result images not similar to the query image.
- The similarities between different kinds of feature vectors are measured using different appropriate metrics. It is not clear how these different metrics should be weighted or combined if several feature vectors are combined for the search.
- The locations of the feature vectors in the high dimensional vector space represent an ordered arrangement. Due to the high dimensions, however, this order – the “inter image relationships” – are unimaginable for human users.

3. SELF-ORGANIZING MAPS

Self-organizing maps (SOMs) are a data visualization technique introduced by T. Kohonen, which reduce the dimensions of data through the use of self-organizing neural networks [4]. SOMs produce a map of usually one or two dimensions which plot the similarities of the data by grouping similar data items together. The basic Self-Organizing Map can be visualized as a neural-network array. The nodes of this array are trained and get specifically tuned to various input signal patterns. The learning process of the network is competitive and unsupervised.

The learning or training of a SOM consists of the following procedure: Initially an array of vectors (nodes) \( m_i \) is set up with random vector values. Next a sequence of training vectors is matched against all vectors of the array. For each training vector \( x \) the best matching node \( m_c \) (in the sense of minimal distance) is determined.

\[
c = \arg \min_i \| x - m_i \|
\]

The vector of the winning node \( m_c \) and its neighborhood are updated such that

\[
m_i(t+1) = m_i(t) + h(t)[x - m_c(t)],
\]

where \( t \) are discrete time steps, \( h_c(t) \) is a neighborhood function that decays for increasing values of \( t \) and for nodes with larger distance from \( m_c \). This means that the winning node and its neighborhood are adapted to the training vector. After successive training the locations of the responses in the array tend to become ordered in the learning process as if some meaningful nonlinear coordinate system for the different input features were being created over the network.

Kohonen SOMs have been used for CBIR systems [3]. Similar images were retrieved by determining the closest
neighbors in the SOM. This technique was combined with a relevance feedback system and a tree-structured SOM was used to reduce computation time.

4. PROPOSED ALGORITHM

So far, the way humans understand image content has not fully been understood. The main problem of CBIR systems still consists in extracting image features that describe the perceptual and semantic meanings. Despite of the ongoing research, currently the CBIR systems are still far from being mature.

The main idea of the proposed algorithm is not to try to improve the results of CBIR systems, but rather to use the techniques of CBIR and SOMs in such a way that a user can navigate within large image set in a convenient way. Figure 2 shows the basic scheme of the proposed iterative image sorting theme. The blue lines indicate the possible user interactions.

At first the basic idea will be depicted, followed by a more detailed description.

0. Before starting, an initial selection of the image sets that are to be processed (searched or browsed) has to be made. (This could be the choice of particular folders.)

The initial amount of images can be reduced by filtering the images according to some technical metadata. (Older images or particular formats could be excluded from the further process.)

1. In the next step the feature vectors of the images are extracted. (At an early stage this could be the color layout descriptor.)

2. The feature vectors are used to generate a self-organizing map.

3. Depending on the topology of the map, different visualizations are possible, …

4. … that can be browsed and navigated by the user.

5. Within this sorted map the user can make selections of subsets that will be used for further processing.

After the selection of a sub-set a new iteration can be initiated (starting again with step 1). The descriptor selection allows choosing another feature extraction mode. After several iterations the initial huge set of images will have been significantly reduced in size, making it much easier to find particular images.

Implementation details

Feature Extraction

We used the MPEG-7 visual descriptors. It turned out that color descriptors should be used at an early stage of the sorting process.

Sorting methods / Construction of the SOM

In contrast to normal SOMs the construction rules for the maps had to be modified to ensure that no node position was occupied by more than one image, because otherwise these images would be overlapping when displayed.

We investigated three different SOM-types. Two 2D-maps (a square and the surface of a sphere) and a three-dimensional cube were realized. The cube can preserve more of the original topology; however both two-dimensional maps were easier to navigate for most users.

Visualization methods / Navigation

For each map type appropriate display and navigation concepts were developed. For all map types it is very important to allow a smooth continuous navigation in order “not to get lost”. The 2D-map can be zoomed and panned. For the sphere map a rotation and a zoom was realized. For fast inspection of individual images a lens mode was realized that can locally magnify images. The 3D-cube mapping allows transitions and rotations. Figure 3 shows the two 2D-visualisations.

Selection of subsets

The user can make a selection from the displayed images. This subset will be used for a new sorting step, possibly with another descriptor. The selection of this image subset can be made in an additive or subtractive way. Particular image sets (regions in the map) can be added or a new
subset may be constructed by removing partial image sets (regions) from the actual set. The remaining image subset will then be used for a new sorting step.

**Execution speed**

For a smooth operation a fast calculation of the maps is necessary. The features vectors do not have to be recalculated at each sorting step, for fast execution speed they are cached.

## 5. RESULTS

We proposed a system for automatic sorting of images. This system can be used to organize large image sets, in addition also image retrieval or search is possible. A prototype (Figure 3) for this system has been realized allowing three different maps and visualisation types. Although the 3D-map could preserve most of the initial inherent relationship of the images, it turned out that the 2D-maps were easier to navigate.

Practical test showed that image sets with up to 2000 images could be handled with the proposed scheme. The sorting and retrieval results using this scheme were very promising. Several users that evaluated our prototype could easily retrieve searched images.

The navigation speed is sufficient on normal computers; the calculation speed of new maps however has to be increased.

Further investigations will concentrate on finding the optimal order for the descriptor selection.

### 6. REFERENCES


