Using Image Segments in PicSOM CBIR System

Mats Sjöberg, Jorma Laaksonen, and Ville Viitaniemi

Laboratory of Computer and Information Science,*
Helsinki University of Technology,
P.O.BOX 5400, 02015 HUT, Finland
E-mail: {mats.sjoberg,jorma.laaksonen,ville.viitaniemi}@hut.fi,
WWW home page: http://www.cis.hut.fi/picsom

Abstract. The content-based image retrieval (CBIR) system PicSOM uses a variety of low-level visual features for indexing an image database. In this paper we describe the implementation of segmentation into the PicSOM framework. That is, we have modified the system to use image segments as a supplement to entire images in order to improve the retrieval accuracy. In a series of experiments, we compare this new method to the baseline PicSOM system. The results confirm that using both segments and entire images together always increases the precision of retrieval.

1 Introduction

The importance of visual information has increased in recent years. Computer systems can today store huge amounts of image data, which has made automated image retrieval increasingly important. In many areas textual descriptions of images are not available or not sufficient to retrieve desired images from a database. Often the only solution is to consider the visual content of the images themselves: content-based image retrieval (CBIR) systems index the images by low-level visual properties either with or without prior image segmentation.

The general problem of image understanding is intrinsically linked to the problem of image segmentation. That is, if one understands an image, one can also tell what the distinct parts of it are. Segmentation thus seems to be a natural part of image understanding. But for an automatic system segmentation is not trivial and the results seldom correspond to the real objects in the image. But even so segmentation may be useful in CBIR, because different, visually homogeneous regions somehow characterise the objects and scenes in the image.

CBIR systems that have employed segmentation techniques include e.g. NetRa [1], VisualSEEk [2], BlobWorld [3], SIMPLIcity [4]. Additionally, such methods as Unified Feature Matching (UFM) [5] and the use of point configurations in the feature space [6] have been presented. The approaches differ mainly in

* This work was supported by the Academy of Finland in the projects *Neural methods in information retrieval based on automatic content analysis and relevance feedback* and *New information processing principles*, the latter being part of the Finnish Centre of Excellence Programme.

the fashion the segment-wise similarities are combined to form image-wise similarities used in the retrieval. In this work, we present a novel scheme for using segmentation in our PicSOM CBIR system and demonstrate that it indeed increases the accuracy of image retrieval.

2 PicSOM

The PicSOM CBIR system [7] uses the *Self-Organizing Map* (SOM) [8] algorithm to index and determine the similarity of images. PicSOM uses several SOMs in parallel to retrieve images relevant to a query. These parallel SOMs have been trained with separate data sets obtained using different feature extraction algorithms on the same images. So each SOM arranges the same images differently, according to its particular visual features.

For training the SOMs, PicSOM uses the *Tree-structured Self-Organizing Map* (TS-SOM) [9] algorithm. A TS-SOM has several layers of normal SOMs with increasing size. Each unit, except those in the lowest layer, has an area of child units in the larger SOM below. When the best-matching map unit (BMU) has been found in one layer, it is only necessary to search through its child units and their closest neighbours to find the BMU in the layer below. This scheme resembles the normal tree-search algorithm and reduces the complexity of the BMU search from $O(n)$ to $O(\log n)$.

The main principle used in PicSOM is *query by pictorial example* (QBPE) [10]. This means that the system shows the user a set of example images, which he then indicates as relevant or nonrelevant to the current query, i.e. close to or far from what he is searching for. Based on this *relevance feedback* information [11] PicSOM changes its configuration so that it will display better image examples in the next round. Relevance feedback is thus an iterative process used for refining the query and a form of supervised learning. PicSOM adapts to different query situations by weighting the influence of the parallel SOMs differently. By the use of relevance feedback after each iteration, PicSOM automatically adapts the weights to increase the influence of those SOMs that give the most valuable relevance and similarity information.

A detailed description of PicSOM can be found in [7]. The PicSOM WWW home page, with a list of publications on PicSOM and a working demonstration is located at `http://www.cis.hut.fi/picsom`.

3 Implementing Segmentation in PicSOM

3.1 Baseline Algorithm

In the QBPE process of PicSOM the user evaluates each shown image by marking it either as relevant or nonrelevant. This information is translated to the SOMs by locating the BMUs of the images. There can be several situations in any spatial neighbourhood on any particular SOM: a) many relevant images, b) only nonrelevant images, c) relevant and nonrelevant images mixed or d) no rated images. The first two cases, a) and b), indicate that the feature used in creating this map is good at separating relevant images from nonrelevant. Case c) on the other hand means that the feature is not very useful in this query.

After a query round, all relevant images get a positive weight inversely proportional to the total number of relevant images. The nonrelevant images get a negative weight inversely proportional to their total number. So the grand total of all weights is always zero. In each TS-SOM layer, these values are summed into the BMUs of the images resulting in a sparse value field on the maps. The value field is then low-pass filtered or "blurred" to spread the relevance information between neighbouring map units. This is because neighbouring map units have similar properties and it is probable that neighbours of relevant images are relevant too.

In this way all the units in the maps, and thus also the images mapped to the units, get a qualification value depending on the local density of relevant images. Maps with very mixed distributions of relevant and nonrelevant images get low qualification values as a result of the low-pass filtering and therefore they automatically get less influence in the image selection process.

In our implementation we first retrieve a fixed number of yet unseen images with the highest qualification values from each SOM. Then we remove duplicate images by summing their qualification values from all SOMs. The 20 images with the highest total qualification values are used as the new example images in the next query round.

3.2 Using Image Segments

The implementation of segmentation in PicSOM was done by generalising the original algorithm so that not only the entire images but also image segments are treated as objects in their own right. At the same time, the segments are also considered to be sub-objects of the images they are a part of.

The image segments were obtained from an automatic segmentation algorithm. Therefore we did not assume perfect correspondence between real-word entities and the segments. We extracted feature vectors from all the image segments by using the same algorithms that we had already used for the entire images. Separate TS-SOMs were trained from these vectors.

The relevance feedback process described in the previous section was modified so that when an image is marked as relevant all its sub-objects (i.e. segments) are likewise marked as relevant. Qualification values are then calculated for all the objects on all the TS-SOMs. The qualification values of all the sub-objects are added to the qualification values of their parent objects. Duplicate images from SOMs of different feature types are resolved by summing up the qualification values. Finally, 20 yet unseen images with the highest qualification values are, as before, chosen as the example images for the next query round.

In the new image selection process described above, the relevance values given to entire images are thus first given also to their contained image segments. In the last stage, the segment-wise qualification values are then summed again to produce image-wise qualification values used in the actual selection.

4 Experiments

To assess whether the introduction of segmentation into PicSOM gave any advantage we ran a series of experiments with hand-picked classes of images. We wanted to see how well the system could find members of a certain image class from a large database.

4.1 Performance Evaluation

Evaluating the performance of a CBIR system is never trivial. Even between humans the interpretation of the contents of an image might differ. In our experiments we have used a set of ground truth image classes that have been hand picked according to certain verbal criteria.

To evaluate the performance of the CBIR system we plot the initial portion of the curves showing *relative precision* against *recall*. Recall $\mathcal{R}$ expresses how large a portion of the relevant image class $\mathcal{C}$ has been shown after a total of t images:

$$\mathcal{R}(t) = \frac{\sum_{i=1}^{t} h_i}{N_{\mathcal{C}}} \in [0,1], \quad t = 1, 2, \ldots, N_T, \tag{1}$$

where $N_{\mathcal{C}}$ is the number of images belonging to class $\mathcal{C}$ and N_T is an upper limit for the number of images the user is supposed to be willing to retrieve. h_i gets the value one if the image retrieved with index i belongs to the desired image class and zero otherwise. Precision $\mathcal{P}$ indicates the accuracy of retrieval, i.e. how exclusively only relevant images have been retrieved:

$$\mathcal{P}(t) = \frac{\sum_{i=1}^{t} h_i}{t} \in [0,1], \quad t = 1, 2, \ldots, N_T. \tag{2}$$

Relative precision is obtained from the precision by dividing $\mathcal{P}$ with the *a priori* probability ρ_C of the class.

The recall–relative precision plot first shows, for small values of t, the initial accuracy of the CBIR system. After that the evolution of the curve indicates how well the relevance feedback mechanism works. With good use of relevance feedback, $\mathcal{P}(t)$ should initially rise and then turn to a slow decline when a sufficiently large portion of the relevant images has been shown.

4.2 Feature Extraction and Segmentation Methods Used

Two simple low-level visual features have been used. The 3-dimensional *average RGB colour* is calculated as the average of the red, green and blue colour components of the image pixels.

Texture neighbourhood is an 8-dimensional textural feature examining the Y-values (luminance) of the YIQ colour representation of the 8-neighbourhood of each pixel. The values of the feature vector are then the estimated probabilities $\bar{P}_i$ that the neighbour pixel in position i is brighter (higher Y-value) than the central pixel. When the Y-value of pixel k is y_k and its neighbour in position i has the Y-value $y_{k,i}$, the probability estimate $\bar{P}_i$ can be calculated as

$$\bar{P}_i = \frac{1}{n}\sum_{k=0}^{n-1} s(y_{k,i}, y_k), \text{ where } s(a,b) = \begin{cases} 1 & \text{if } a > b \\ 0 & \text{otherwise} \end{cases}, \tag{3}$$

where n is the number of image pixels. The feature vector for one image or image segment is then $[\bar{P}_0\ \bar{P}_1\ \dots\ \bar{P}_7]^T$.

For segmentation we used the *isodata* variant of the *k-means* algorithm combined with *region merging*. The initial *k-means* segmentation was based on the first three central moments of the colour distribution [12] in the HSV colour space. The moments were calculated separately for each of the colour space components and collected into a nine-component vector $\mathbf{c}$. For the estimation of the moments the image was divided into 3×3 tiles. In order to save computation time only a small fraction of image pixels was used in the clustering.

Similar colour moment features were used for region merging. The criterion for merging neighbouring regions was the Euclidean distance of colour moments, weighted with the size of the regions involved:

$$d(i,j) = (\mathbf{c}_i - \mathbf{c}_j)^T(\mathbf{c}_i - \mathbf{c}_j)[\sqrt{\min(s_i, s_j)} + b] , \tag{4}$$

where $s_i = |R_i| / \sum_j |R_j|$ is the relative size of region i and b is a constant.

4.3 Experiment Setting

We used a database of 59 995 colour photographs from the Corel Gallery 1 000 000 product converted to JPEG format using a Corel tool. The image sizes are 384×256 or 256×384 pixels. From this set we hand picked six sets of ground truth images:

- **faces,** 1115 images (a priori probability 1.85%), where the main target of the image has to be a human head which has both eyes visible and the head has to fill at least 1/9 of the image area.
- **cars,** 864 images (1.44%), where the main target of the image has to be a car, and at least one side of the car has to be completely shown in the image and its body to fill at least 1/9 of the image area.
- **sunsets,** 663 images (1.11%), where the image has to contain a sunset with the sun clearly visible in the image.
- **houses,** 526 images (0.88%), where the main target of the image has to be a single house, not severely obstructed, and it has to fill at least 1/16 of the image area.
- **horses,** 486 images (0.81%), where the main target of the image has to be one or more horses, shown completely in the image.
- **planes,** 292 images (0.49%), where all airplane images have been accepted.

We trained a total of four TS-SOMs. Two were created by using features calculated from the entire images and two by using features from the image segments. In each pair, one TS-SOM was trained with the *average RGB colour* features and the other with the *texture neighbourhood* features. The sizes of the TS-SOM layers were 4×4, 16×16, 64×64 and 256×256 units. Every object (entire image or image segment) was used 100 times to train each SOM layer.

The experiments were run on each of the ground truth classes in three different ways: 1) using images only, 2) using segments only, and 3) using both images and segments in parallel. Each image query was started by giving the system an initial image that was automatically selected from the correct class. After that we run 50 query rounds with 20 images retrieved at each iteration. The ground truth classes were used during the iteration to determine the relevancy of the images returned by the system. This information was given to the system as relevance feedback. The experiment was repeated so that each image in the ground truth classes was used once as the initialiser. After this the precision and recall results were averaged to produce one recall–relative precision curve for each class.

5 Results

In Fig. 1 we have plotted the recall–relative precision graphs for the six ground truth classes. The plots have been arranged according to their qualitative similarities. The plots in the first column show similar behaviour: for the faces, houses and horses classes the precision increases continuously and using only entire images results in better precision than using only image segments. In the second column, entire images are superior to image segments only in the planes class. In the sunsets and cars classes using only image segments produces better results than using only entire images.

The planes and sunsets classes differ substantially from the rest in that the relative precision of these classes starts to decrease during the query. At the same time, the values of the relative precision and recall for these classes are clearly higher than those for the other four classes. Therefore a substantial fraction of the relative images are found in the query process.

The most important observation is that in all the six cases combining features calculated from both image segments and entire images results in a precision clearly superior to that of either one feature type used separately. This finding can be regarded as an indication of the usefulness of automatic segmentation in a CBIR system.

6 Conclusions

In this paper we have studied the use of automatically generated image segmentations in the PicSOM CBIR system. The results of the performed series of experiments show that introducing segmentation into PicSOM increases the image retrieval accuracy. However, this was in general true only when the image

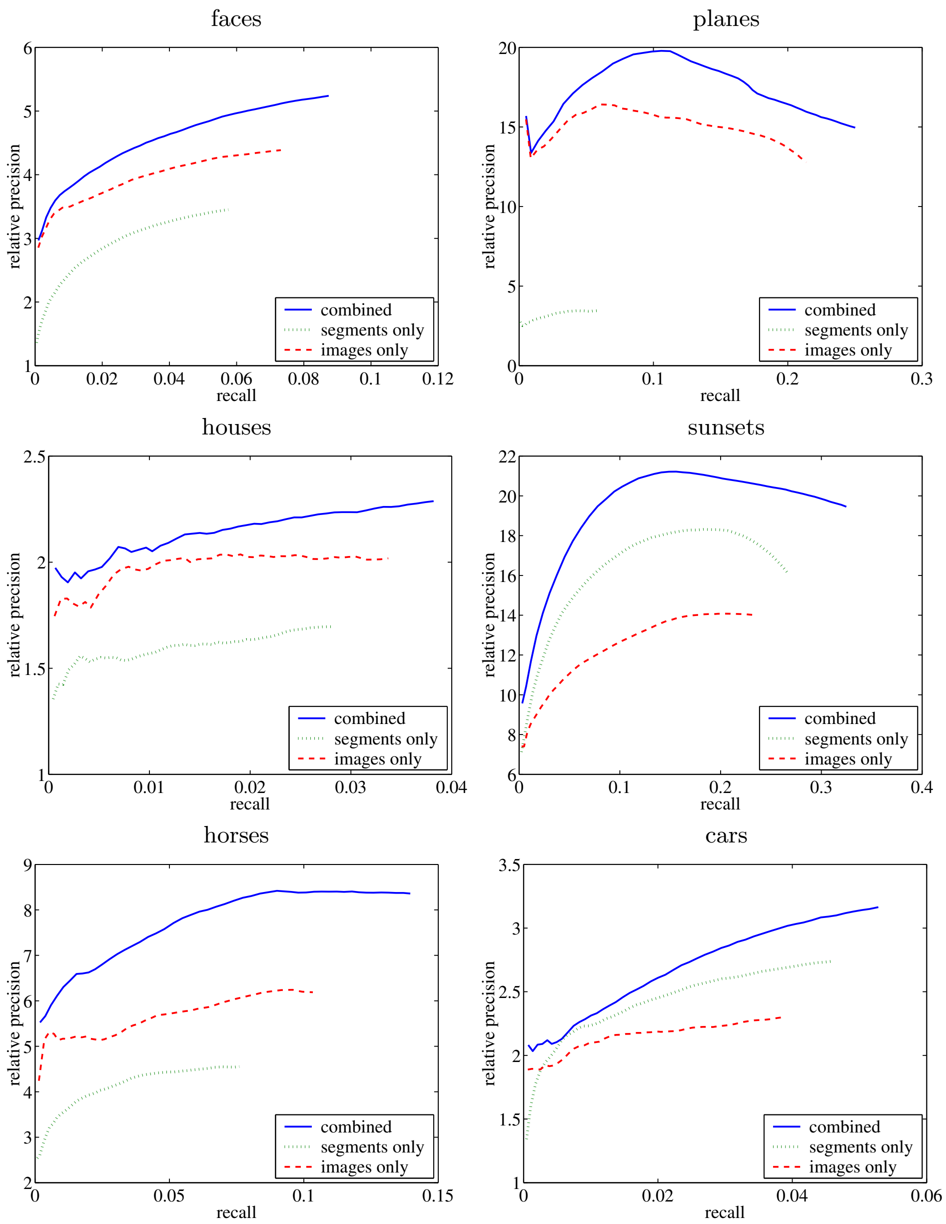

Fig. 1. Recall–relative precision graphs for all six ground truth classes. The plots have been arranged according to their qualitative similarities. It can be seen that using the combination of features from both the segments and entire images always results in the best precision.

segments were used together with the original images, not when they were used as the only form of visual data in the system.

The results are perhaps not generalisable to other CBIR systems, but at least they serve as a guideline to continue research on the use of automatic segmentation methods in CBIR. Even though the used segments do not always correspond to real objects in the image, features calculated from homogeneous image regions may characterise the image's content better than average feature values obtained from the entire heterogeneous image area.

Our experiments reported here were only initial tests and should be expanded by investigating other types of images and other segmentation and feature extraction methods, such as MPEG-7 descriptors. Also, as there are alternative ways of incorporating segmentation into PicSOM, these should be tried out as well and compared with the performance of other existing CBIR systems.

References

1. Ma, W.Y., Manjunath, B.S.: NeTra: a toolbox for navigating large image databases. Multimedia Systems **7** (1999) 184–198
2. Smith, J.R., Chang, S.F.: VisualSEEk: A fully automated content-based image query system. In: Proceedings of the 4th International ACM Multimedia Conference (ACM Multimedia '96), Boston, MA (1996) 87–98
3. Carson, C., Belongie, S., Greenspan, H., Malik, J.: Color- and texture-based image segmentation using EM and its application to content based image retrieval. IEEE Transactions on Pattern Analysis and Machine Intelligence **24** (2002) To appear in August 2002.
4. Wang, J.Z., Li, J., Wiederhold, G.: SIMPLIcity: Semantics-sensitive integrated matching for picture libraries. IEEE Transactions on Pattern Analysis and Machine Intelligence **23** (2001) 947–963
5. Chen, Y., Wang, J.Z.: Looking beyond region boundaries: Region-based image retrieval using fuzzy feature matching. In: Multimedia Content-Based Indexing and Retrieval Workshop, September 24-25, INRIA Rocquencourt, France (2001)
6. Dimai, A.: Invariant scene description based on salient regions for preattentive similarity assessment. In: 10th International Conference on Image Analysis and Processing (ICIAP), September 27-29, Venice, Italy (1999) 957–962
7. Laaksonen, J., Koskela, M., Oja, E.: PicSOM—Self-organizing image retrieval with MPEG-7 content descriptions. IEEE Transactions on Neural Networks, Special Issue on Intelligent Multimedia Processing **13** (2002) 841–853
8. Kohonen, T.: Self-Organizing Maps. Third edn. Volume 30 of Springer Series in Information Sciences. Springer-Verlag (2001)
9. Koikkalainen, P.: Progress with the tree-structured self-organizing map. In: 11th European Conference on Artificial Intelligence, European Committee for Artificial Intelligence (ECCAI) (1994)
10. Chang, N.S., Fu, K.S.: Query-by-Pictorial-Example. IEEE Transactions on Software Engineering **6** (1980) 519–524
11. Salton, G., McGill, M.J.: Introduction to Modern Information Retrieval. Computer Science Series. McGraw-Hill (1983)
12. Stricker, M., Orengo, M.: Similarity of color images. In: Storage and Retrieval for Image and Video Databases III (SPIE). Volume 2420 of SPIE Proceedings Series., San Jose, CA, USA (1995) 381–392