Using Long-Term Learning to Improve Efficiency of Content-Based Image Retrieval

Markus Koskela and Jorma Laaksonen

Laboratory of Computer and Information Science, Helsinki University of Technology,
P.O.BOX 5400, 02015 HUT, Finland
{markus.koskela,jorma.laaksonen}@hut.fi

Abstract. Content-based image retrieval (CBIR) is an emerging research field, studying retrieval of images from unannotated databases. In CBIR, images are indexed on the basis of low-level statistical features that can be automatically derived from the images. Due to the gap between high-level semantic concepts and low-level visual features, the performance of CBIR applications often remains quite modest. One method for improving CBIR results is to try to learn the user's preferences with intra-query learning methods such as relevance feedback. However, relevance feedback provides user interaction information which can automatically be used also in long-term or inter-query learning. In this paper, a method for using long-term learning in our PicSOM system is presented. The performed experiments show that the system readily supports the presented user interaction feature and that the efficiency of the system can be substantially increased by using it in parallel with the MPEG-7 visual descriptors.
Keywords: content-based image retrieval, self-organizing map, relevance feedback, long-term learning

1 Introduction

Content-based image retrieval (CBIR) has received considerable research interest in the recent years. The field has matured into a distinct research discipline which differs substantially from text-based information retrieval. With low-level visual features it is not possible to base image queries on verbal terms like in text-based retrieval. Therefore, other query methods must be applied. One common approach to formulate queries in CBIR is *query by pictorial examples*, where the image queries are based on example images.

Depending on the image domain, the database in question, and the amount of *a priori* information available on the images, the CBIR problem exhibits a varying degree of difficulty. A difficult setting is encountered when the task is to retrieve images from a large database of miscellaneous images. Since very few assumptions about the images can be made, only representations of very general nature can be used and the general low-level features used in CBIR are insufficient to discriminate images well on a conceptual level. This creates a fundamental problem, namely the gap between the high-level semantic concepts

used by humans to understand image content and the low-level visual features used by a computer to index the images in a database.

A common method to improve CBIR performance is to use *intra-query* learning by relevance feedback. Improved retrieval results can be obtained if the image query can be turned into an iterative process towards the desired retrieval target. Relevance feedback can be seen as a form of supervised learning to adjust the subsequent query rounds by using the information gathered from the user's feedback. With relevance feedback, the learning takes place during one query instance and the results are erased when starting a new query.

Relevance feedback provides information which can also be used in an *inter-query* or *long-term* learning scheme. The relevance evaluations provided by the user during a query session partition the set of seen images into relevant and nonrelevant classes with respect to that particular query target. The fact that two images belong to the same class is a cue for similarities in their semantic content. While relevance feedback has achieved prevailing popularity in CBIR, less research has been focused on exploiting long-term learning. Still, some form of long-term learning has been incorporated into a number of CBIR systems. In *MetaSeek* [1], all user interactions were stored and used in later queries in selecting between a set of independent image search engines. The log files of the *Viper* CBIR system were used to adjust weights for different features in [2]. A Bayesian framework for both short-time and long-time learning was presented in [3]. In [4], latent semantic indexing (LSI) was applied to CBIR by considering the images as the vocabulary of the system and the classes of relevant images as documents whose words are the images. In [5], the feature-based similarities are first computed and the images are ranked correspondingly. Then, the images are re-ranked based on pair-wise semantic correlations obtained from existing relevance feedback information.

2 PicSOM

The PicSOM CBIR system is a framework for research on methods for content-based image retrieval (see [6] for a recent review; the PicSOM home page is at *http://www.cis.hut.fi/picsom*). The system is based on using several parallel Self-Organizing Maps (SOMs) [7] trained with separate feature data. The SOM defines an elastic, topology-preserving grid of points that is fitted to the input space. It attempts to represent all the available observations with an optimal accuracy by using a restricted set of models.

As a result of using multiple SOMs, the system inherently uses multiple features for image retrieval and generally benefits from using all available features as it automatically neglects the poorly-working ones. Features are usually comprised of statistical visual data such as the MPEG-7 [8] content descriptors. Additional vectorial data can, however, be used to train corresponding SOMs and thus be used in image retrieval. In this work, recorded user–system interaction from previous queries and existing keyword annotations for the images are used as additional features.

To reduce the complexity of training large SOMs, a special form of the algorithm, the Tree Structured Self-Organizing Map (TS-SOM) [9] is used. After training the SOMs with the TS-SOM algorithm, the map units are connected with the images of the database. This is done by locating the best-matching map unit (BMU) for each image. Also, among the images sharing a common BMU, the best-matching one is used as a visual label for that unit.

2.1 Relevance Feedback with Self-Organizing Maps

The relevance feedback mechanism of PicSOM, implemented by using several parallel SOMs, is a crucial element of the retrieval engine. Only a short overview is presented here, see [10] for a more comprehensive treatment. The basic assumption is that images which are similar according to a specific visual feature are located near each other on the corresponding SOM surface. Therefore, we are motivated to spread the relevance information given by the user also to the neighboring map units of the shown images. This is done as follows. All relevant images are first given equal positive weight inversely proportional to the number of relevant images. Likewise, nonrelevant images receive negative weights that are inversely proportional to their total number. The overall sum of these relevance values is thus zero. For each SOM, the values are then mapped from the images to the corresponding BMUs where they are summed.

The resulting sparse value fields on the SOM surfaces are low-pass filtered to produce qualification values for each SOM unit and its associated images. The low-pass filtering of sparse value fields can be performed by convolving the field with a tapered window function. The exact shape of the window function is not significant, eg. triangular or gaussian windows can be used, but the length of the window is important for both retrieval performance and computational complexity. The total qualification value for each image is finally obtained by summing the corresponding responses on all used SOMs. Content descriptors that fail to coincide with the user's conceptions mix positive and negative user responses in nearby map units. Therefore, they produce lower qualification values than those descriptors that match the user's expectations and impression of image similarity. As a consequence, the different content descriptors and the SOMs formed from them do not need to be explicitly weighted as the system automatically takes care of weighting their opinions.

3 Long-Term Learning Based on User Interaction

In the PicSOM system, the user evaluates the shown images and marks which ones are relevant to the query. For the lifetime of that query, the relevant images constitute the set of positive images whereas the remaining ones are implicitly regarded as negative ie. nonrelevant. This approach is just one possibility among others. In some systems negative examples must also be explicitly provided. The relevance scale may also be finer, eg. containing options like "very relevant", "relevant", "somewhat relevant" and so on. All these cases fit well to the method

presented below as we only require the representation of the relevance scale as scalar values. In our case, suitable values for relevant and nonrelevant images are $+1$ and -1, respectively. The value 0 is used for unseen images. On the other hand, the usefulness of negative feedback is questionable here as the set of rejected images during an image query might not share any common characteristics. Another possibility, used also in the experiments presented in this paper, is thus to use $+1$ for relevant images and 0 for all other images.

Latent semantic indexing (LSI) [11] is a common technique in text-based information retrieval. In LSI, the documents are projected into a space with "latent" semantic dimensions using singular value decomposition (SVD). The basis for LSI is the vector space model, ie. the m documents are represented by the words in them by using an $n \times m$ term-by-document matrix $\mathbf{X}$, where n is the number of different words. SVD is then applied as

$$\mathbf{X} = \mathbf{U}\,\mathbf{S}\,\mathbf{V}^T \tag{1}$$

where $\mathbf{U}$ and $\mathbf{V}$ are $n \times r$ and $m \times r$ orthonormal matrices and $\mathbf{S}$ is an $r \times r$ diagonal matrix containing the singular values of $\mathbf{X}$ on the diagonal and $r \leq \min(n, m)$ is the rank of $\mathbf{X}$. LSI is applied by considering only k ($k \leq r$) largest singular values of $\mathbf{S}$:

$$\widehat{\mathbf{X}} = \widehat{\mathbf{U}}\,\widehat{\mathbf{S}}\,\widehat{\mathbf{V}}^T \approx \mathbf{X} \tag{2}$$

and a representation of the originally m-dimensional data in k dimensions is obtained as the rows of $\mathbf{Y} = \widehat{\mathbf{U}}\,\widehat{\mathbf{S}}$. One can view LSI as a transform of the term space to an orthogonal "concept" space which locates clusters of co-occurring terms, thus finding meanings based on term co-occurrences.

In this work, LSI is applied as in [4]. That is, instead of considering images as documents as in the retrieval phase, here the user-provided relevance evaluations are considered as the documents and the images in the database as the words in the vocabulary. Term or image frequency weighting is unnecessary as each image may appear at most once in one relevance evaluation, but document frequency weighting, ie. weighting elements of $\mathbf{X}$ by the number of documents in which the corresponding term occurs in, can be applied. In our setting, LSI is primarily used to perform dimensionality reduction. This is needed as the dimensionality m of the data equals the number of image queries in the training data, which may well be in the order of hundreds or thousands and thus excessive for direct usage in SOM training.

The rows of matrix $\mathbf{Y}$, each corresponding to one image, are treated as a user interaction feature of dimensionality k and the corresponding SOM is trained and used in parallel and similarly as the SOMs trained with visual features. The result is illustrated in Figure 1, in which the 16×16-sized TS-SOM level of the user interaction feature is displayed. The sparsity of the map is a direct consequence of the sparsity of the data; images in the same relevance evaluation tend to get mapped into the same map unit. The shown images are the visual labels given to the SOM map units. It can be observed that images with similar semantic content have been mapped near each other on the map.

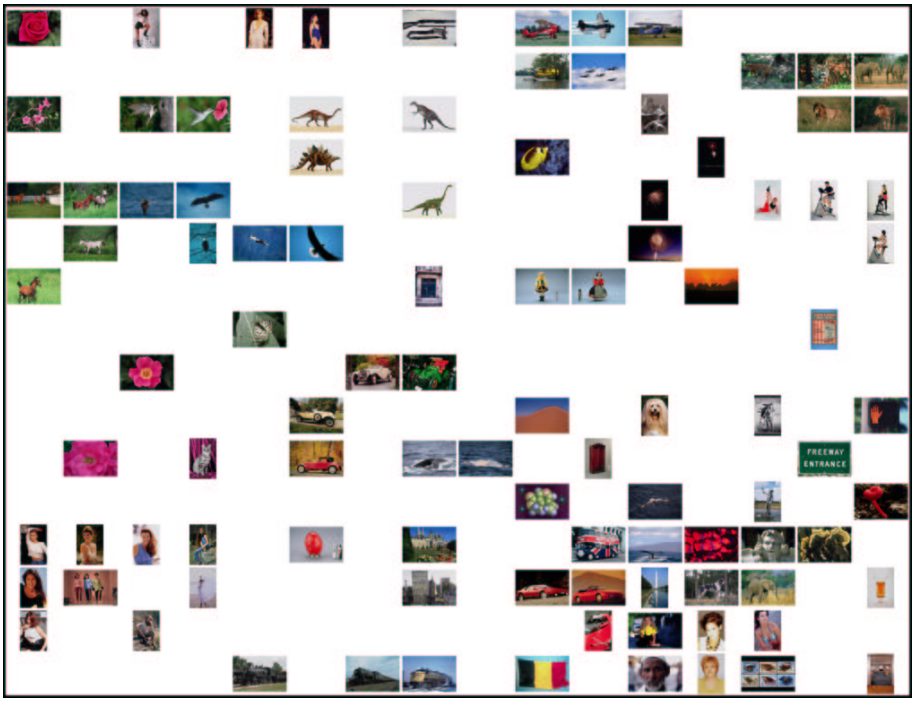

Fig. 1. The 16×16-sized TS-SOM level trained with the user interaction feature.

User-provided relevance evaluations are notably similar to hidden annotations [12]. In some occasions, the image database may contain elaborate manually-constructed captions or other annotations. Such annotated databases can be found eg. in commercial image libraries and medical databases. Implicit annotations can also be found, eg. from the text surrounding an image in the WWW. These annotations describe high-level semantic content of the image and often contain invaluable information for retrieval. Hidden annotations can be seen as high-quality user assessments. Images having a certain term in their annotations can be seen as the set of relevant images when the user was querying for images containing the concept corresponding to the term in question. Our technique can thus readily be utilized for keyword annotations.

4 Experiments

We used a database of 59 995 images from the Corel Gallery 1 000 000 product. To run automated tests, we created manually six ground truth image classes: **faces** (1115 images, *a priori* probability 1.85%), **cars** (864 images, 1.44%), **planes** (292 images, 0.49%), **sunsets**, (663 images, 1.11%), **horses**, (486 images, 0.81%), and **traffic signs**, (123 images, 0.21%). As visual features, we used a subset of

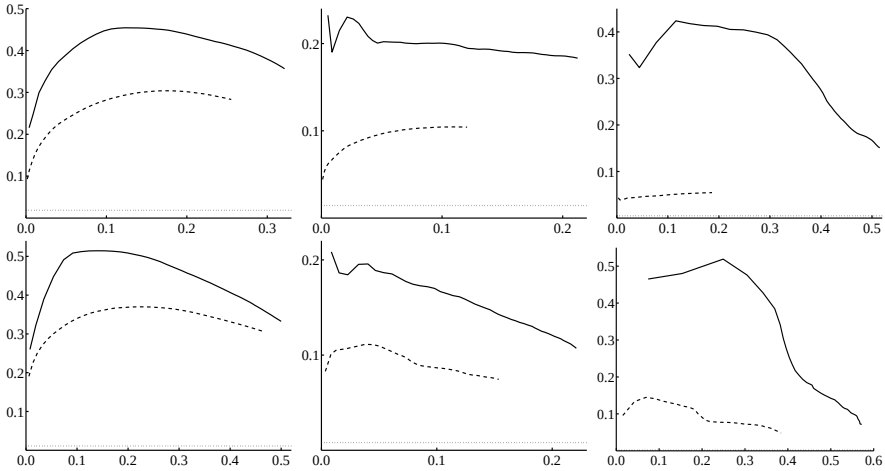

Fig. 2. Recall–precision plots (x-axis: recall; y-axis: precision) using MPEG-7 descriptors with (solid curve) and without (dashed curve) the user interaction feature. The *a priori* probability of the class is shown with a dotted line. Used classes were (top row, left-to-right) **faces**, **cars**, **planes**, (bottom row) **sunsets**, **horses**, and **traffic signs**.

MPEG-7 [8] descriptors, viz. *Scalable Color*, *Dominant Color*, *Color Structure*, *Color Layout*, *Edge Histogram*, *Homogeneous Texture*, and *Region Shape*.

If the size of the database, N, is large enough, we can assume that there is an upper limit N_T of images ($N_T \ll N$) the user is willing to browse during a single retrieval session. In our test setting, each image in the studied class is "shown" to the system one at a time as the initial reference image for category search. The system should then return images belonging to the same class, resulting in a leave-one-out type testing of the class. The system was set to return 20 images at each round, and with 50 rounds per query the total number of seen images was $N_T = 1000$ images, ie. 1.67% of the database size.

We chose to show the evolution of *precision* $\mathcal{P}(n)$ as a function of *recall* $\mathcal{R}(n)$ during the image retrieval process. Precision and recall are intuitive performance measures that suite also for the case of non-exhaustive browsing. First, the initial values of $\mathcal{P}(n)$ display the initial accuracy of the system. Then, the intermediate values show how the relevance feedback mechanism is able to adapt to the class and the final value $\mathcal{R}(N_T)$ – as well as $\mathcal{P}(N_T)$ – reflects the total number of relevant images found that far.

In our first experiment, the training data for the user interaction feature consisted of 317 saved query sessions recorded in our laboratory in which 6897 images (11.5% of the database) had been marked relevant at least once. The dimensionality k of the data was reduced to $k = 50$ with LSI. In our second experiment, we used the keywords provided for the images by Corel as keyword annotation data. It should be noted that the test ground truth classes were manually constructed using independently specified membership criteria, although

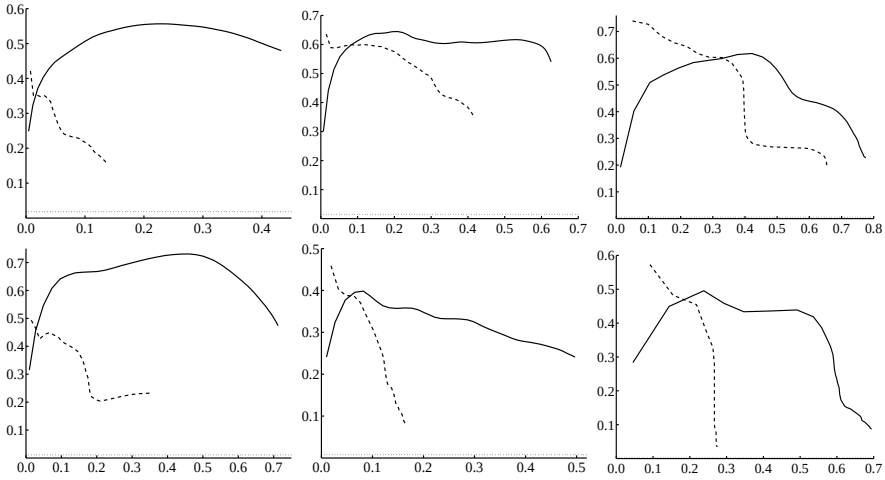

Fig. 3. Recall–precision plots (x-axis: recall; y-axis: precision) using the keyword feature with (solid curve) and without (dashed curve) the MPEG-7 descriptors. The *a priori* probability of the class is shown with a dotted line. Used classes were (top row, left-to-right) **faces**, **cars**, **planes**, (bottom row) **sunsets**, **horses**, and **traffic signs**.

obviously there are strong correlations between the hand-picked classes and semantically close keywords. All keywords associated with only three or fewer images were removed, resulting in 4538 keywords. A non-zero keyword feature vector was obtained for 57 864 images (96.4%). With LSI, the data was then reduced to $k = 150$ dimensions. We trained four-level TS-SOMs with level sizes 4×4, 16×16, 64×64, and 256×256 units for each visual feature and the keyword feature. Since the user interaction feature had non-zero vectors only for 6897 images, the corresponding TS-SOM was limited to three levels (64×64=4096 map units on the bottommost level). Only the bottommost SOM levels were used in the experiments as they provide the most detailed resolution. The used SOM sizes were thus 64×64 for the user interaction feature and 256×256 for others.

The resulting recall–precision plots for the user interaction experiment are shown in Figure 2. The MPEG-7 descriptors are used in all cases with (solid curves) and without (dashed curves) the user interaction feature. It can be seen that the user interaction feature considerably improves retrieval precision, even though only 11.5% of the images are included in the feature. The effect of using keyword annotations is illustrated in Figure 3. As can be expected, retrieval precision in this setting is distinctly higher. Here, the two curves are differentiated by the use of MPEG-7 descriptors; they are included in the solid curves and not used in the dashed ones. It can be observed that by adding the visual features, which perform much worse on their own (cf. Fig. 2), the overall results can be improved, although the initial precision degrades. This is due to PicSOM's ability to automatically weight features and focus on those providing the most useful information during relevance feedback.

5 Conclusions

With large databases of general images, the retrieval performance of low-level visual features alone often remains quite modest and additional feature types may be needed for acceptable performance. A method for improving the performance based on using automatically recorded user-provided relevance evaluations was presented in this paper. In the PicSOM framework, the user interaction data is treated similarly as statistical visual features and, after dimensionality reduction, a separate user interaction SOM is trained and used in retrieval. The experiments showed that the new feature greatly improves the precision of the system without any additional human labor required. The method can also be used for existing keyword annotations, which can result in greatly improved precision in category search, as was observed in the experiments. Still, even with the most efficient keyword feature, it was shown to be beneficial to incorporate also visual features into the query.

References

1. Benitez, A.B., Beigi, M., Chang, S.F.: Using relevance feedback in content-based image metasearch. IEEE Internet Computing (1998) 59–69
2. Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Long-term learning from user behavior in content-based image retrieval. Technical Report 00.04, Computer Vision Group, University of Geneva, Geneva, Switzerland (2000)
3. Vasconcelos, N., Lippman, A.: Learning over multiple temporal scales in image databases. In: Proceedings of Sixth European Conference on Computer Vision (ECCV'2000). Volume 1., Dublin, Ireland (2000) 33–47
4. Heisterkamp, D.R.: Building a latent semantic index of an image database from patterns of relevance feedback. In: Proceedings of 16th International Conference on Pattern Recognition (ICPR 2002). Vol. 4., Quebec, Canada (2002) 134–137
5. Li, M., Chen, Z., Zhang, H.J.: Statistical correlation analysis in image retrieval. Pattern Recognition **35** (2002) 2687–2693
6. 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
7. Kohonen, T.: Self-Organizing Maps. Springer-Verlag (2001)
8. MPEG: MPEG-7 overview (version 8.0) (2002) ISO/IEC JTC1/SC29/WG11.
9. Koikkalainen, P., Oja, E.: Self-organizing hierarchical feature maps. In: Proceedings of International Joint Conference on Neural Networks. Volume II., San Diego, CA (1990) 279–284
10. Laaksonen, J., Koskela, M., Laakso, S., Oja, E.: Self-organizing maps as a relevance feedback technique in content-based image retrieval. Pattern Analysis & Applications **4** (2001) 140–152
11. Deerwester, S., Dumais, S.T., Furnas, G.W., Landauer, T.K., Harshman, R.: Indexing by latent semantic analysis. Journal of the American Society for Information Science **41** (1990) 391–407
12. Cox, I.J., Miller, M.L., Omohundro, S.M., Yianilos, P.N.: Target testing and the PicHunter bayesian multimedia retrieval system. In: Proceedings of 3rd Forum on Research and Technology Advances in Digital Libraries (ADL'96), Washington, DC (1996) 66–75