Accuracy Measurement for Image Retrieval System

Dr. S. Thabasu Kannan[1], P.Kumaravel[2]

[1]Principal, Pannai College of Engineering & Technology, Sivagangai, Tamilnadu, INDIA

[2]Technical Associate 'B', Indian Institute of Astrophysics, Kodaikanal, INDIA

ABSTRACT

During the past decades we have been observing a permanent increase in image data, leading to huge repositories. Content-based image retrieval (CBIR) methods have tried to improve the access to image data. To date, numerous feature extraction methods have been proposed to improve the quality of CBIR and image classification systems.

In this paper, we are analyzing the technique of relevance feedback for the purpose of image retrieval system. The survey is used to study all the methods used for image retrieval system. The structure-based features its task as broad as texture image retrieval and/or classification. To develop a structure based feature extraction, we have to investigate CBIR and classification problems.

Digital image libraries are becoming more common and widely used as visual information is produced at a rapidly growing rate. Creating and storing digital images is nowadays easy and getting more affordable. As a result, the amount of data in visual form is increasing and there is a strong need for effective ways to manage and process it. We have studied support vector machines to learn the feature space distribution of our structure-based features for several images classes. .

CBIR contains three levels namely retrieval by primitive features, retrieval by logical features and retrieval by abstract attributes. It contains the problem of finding images relevant to the users' information needs from image databases, based principally on low-level visual features for which automatic extraction methods are available. Due to the inherently weak connection between the high-level semantic concepts and the low-level visual features the task of developing this kind of systems is very challenging.

A popular method to improve image retrieval performance is to shift from single-round queries to navigational queries. This kind of operation is commonly referred to as relevance feedback and can be considered as supervised learning to adjust the subsequent retrieval process by using information gathered from the user's feedback.

Here we also studied an image indexing method based on a Self-Organizing Maps (SOM). The SOM was interpreted as a combination of clustering and dimensionality reduction. It has the advantage of providing a natural ordering for the clusters due to the preserved topology. This way, the relevance information obtained from the user can be spread to neighboring image clusters. The dimensionality reduction aspect of the algorithm alleviates computational requirements of the algorithm.

It definitely contains the feature of novel relevance feedback technique. The relevance feedback technique is based on spreading the user responses to local self organizing maps neighborhoods. With some experiments, it will be confirmed that the efficiency of semantic image retrieval can be substantially increased by using these features in parallel with the standard low-level visual features.

The measurements like precision and recall were used to evaluate the performance. Precision-recall graph for the 1.000 and 10,000 image data-set and robustness analysis of the 1.000 image database for different brightness have taken.

*Keyword*s---- Relevance, Feedback, Precision, Recall, Accuracy, retrieval system.

I. RELEVANCE FEEDBACK (RF)

RF is a technique for text-based information retrieval to improve the performance of information access systems. The iterative and interactive refinement of the original formulation of a query is known as relevance feedback (RF) in information retrieval. The essence of RF is to move from one shot or batch mode queries to navigational queries, where one query consists of multiple rounds of interaction and the user becomes an inseparable part of the query process. During a round of RF, the user is presented with a list of retrieved items and is expected to evaluate their relevance, which information is then fed back to the retrieval system. The expected effect is that the new query round better represents the need of the user

as the query is steered toward the relevant items and away from the non-relevant ones.

Three strengths of RF are: (a) It shields the user from the inner details of the retrieval system (b) It brings down the retrieval task to small steps which are easier to grasp and (c) It provides a controlled setting to emphasize some features and de-emphasize others.

A. RF IN IMAGE RETRIEVAL

RF is popular todays because of *a)* more ambiguity arises in interpreting images than text, making user interaction more necessary, b) manual modification of the initial query formulation is much more difficult in CBIR than with textual queries. RF can be seen as a form of supervised learning to steer the subsequent query toward the relevant images by using the information gathered from the user's feedback. RF is to regard a system implementing it as one trying to gradually learn the optimal correspondence between the high-level concepts people use and the low-level features obtained from the images. The user does not need to explicitly specify priorities for different similarity assessments because they are formed implicitly by the system based on the user–system interaction. This is advantageous since the correspondence between concepts and features is temporal and case specific. This means that every image query is different from the others due to the hidden conceptions on the relevance of images and their mutual similarity and therefore using a static image may not be sufficient. On the other hand, the user feedback should be seen, instead of as filtering images based on some preexisting meaning, as a process of creating meaning through the interaction.

For implementing RF in a CBIR system, three minimum requirements need to be fulfilled. a) The system must show the user a series of images, remember what images have already been shown, and not display them again. So, the system will not end up in a loop and all images will eventually be displayed b) the user must somehow be able to indicate which images are to some extent relevant to the present query and which are not. Here, these images are denoted as positive and negative seen images. Clearly, this granularity of relevance assessments is only one possibility among others. The relevance scale may also be finer, e.g. containing options like "very relevant", "relevant", "somewhat relevant", and so on. Relevance feedback can also be in the form of direct manipulation of the query structure as with the dynamic visualization methods. c) The system must change its behavior depending on the relevance scores provided for the seen images. During the retrieval process more and more images are assessed and the system has increasing amount of data to use in retrieving the succeeding image sets.

Three characteristics of RF, which distinguishes it from many other applications of machine learning, are (a) Small number of training samples (typically $N_n < 30$).

(b) Asymmetry of the training data. (c) RF is used when the user is interacting with the system and thus waiting for the completion of the algorithm. An image query may take several rounds until the results are satisfactory, so fast response time is essential.

B. METHODS FROM TEXT-BASED IR
i) Relevance Feedback with VSMs

Here, each database item is represented as a point in K-dim space. Textual documents are commonly represented by the words. This information is encoded into a term-by-document matrix X. Similarly to the database items, the query is also represented as a point or vector. In order to do, the documents are ranked according to their similarity. In *text-based retrieval*, the standard similarity measure is the cosine measure (5.7). Dimensions are reduced in the preprocessing step by removing the most common terms. In this model, the following methods for query improvement exist.

a. *Query point movement.* The basic idea is to move the query point toward the part of vector space where the relevant documents are located. This can be used by a formula

$$q_{n+1} = \alpha q_n + \frac{\beta}{N^+(n)} \sum_{I_j \in D^+(n)} f_j - \frac{\gamma}{N^-(n)} \sum_{I_j \in D^-(n)} f_j$$

where q_n is the query pt on the nth round of the query and α, β and γ are weight parameters $(\alpha + \beta + \gamma = 1)$ controlling the relative importance of the previous query point, the average of relevant images, and non-relevant images, respectively. The relevant items concentrated on a specific area of the vector space whereas the non-relevant items are often more heterogeneous. Therefore, we should set the weights so that $\beta > \gamma$. Setting $\gamma = 0$ is also possible, resulting in purely positive feedback. Here the assumption is the distance to the query point increases does not generally capture high-level semantics well due to the semantic gap.

b. *Feature component re-weighting*. The basic idea is to increase the importance of the components of the used feature vectors, which is used to retrieve relevant images. Each component can be given a weight, which is used in calculating the distances between images. This can be easily done by augmenting the used distance measure with component-wise weights: the weight of kth component of the mth feature is denoted as w_{mk}. This can be used by a formula

$$d_{wL_\lambda}(\mathbf{f}_i, \mathbf{f}_j) = \left[\sum_{k=1}^{K} w_{mk} |f_i(k) - f_j(k)|^\lambda \right]^{\frac{1}{\lambda}}$$

Assuming that the feature components are independent, the case when the relevant items have similar values for f(k), i.e. the kth component of feature f, it can be assumed that f(k) captures something the

relevant items have in common and which corresponds to the user's information need.

$$w_{mk} = \frac{c}{\sigma_{mk}^+} \cdot$$

where c is a constant, so $\sum_{k=1}^{K} w_{mk} = 1$.

ii) RELEVANCE FEEDBACK WITH SOMs
a) *Generating the sparse value fields*

The user is expected to mark the relevant images as positive, and the unmarked images as negative. As all images in the database have been previously mapped in their best-matching SOM units at the time the SOMs were trained, and easy to locate the positive and negative images on each SOM in use. The map units are awarded a positive score for every positive image. Likewise, the negative images result in negative scores. These positive and negative scores are scaled so that the total sum of all scores on each map is equal to zero. If the total number of positive and negative images are N+(n) and N-(n) at query round n, the positive and negative scores are

$$x_+(n) = \frac{1}{N^+(n)} \quad \text{and} \quad x_-(n) = -\frac{1}{N^-(n)} \cdot$$

By assumption the neighboring SOM units are similar to it and the images mapped are relevant for the user.

b) *Location-dependent window functions*

Information on the distances between neighboring SOM model vectors in the feature space has earlier been used mainly in visualization. The average relative distance of a model vector to its neighbors can be color-coded with gray-level shades, resulting in a SOM visualization known as the *U-matrix*. Dark or dim shades are often used to visualize long distances whereas bright colors correspond to close similarity between neighboring model vectors. In this method, we relax the property of symmetry in the window function. Intuitively, if the relative distance of two SOM units in the original feature space is small, they can be regarded as belonging to the same cluster and, therefore, the relevance response should easily spread between the neighboring map units. Cluster borders, on the other hand, are characterized by large distances between map units and the spreading of responses should be less intensive.

For each neighboring pair of map units according to 4-neighborhood, say *i* and *j*, the distance in the original feature space is calculated. The distances are then scaled so that the average neighbor distance is equal to one. The normalized distances d_{ij} are then used for calculating location-dependent convolutions with two alternative methods, illustrated in the following figure.
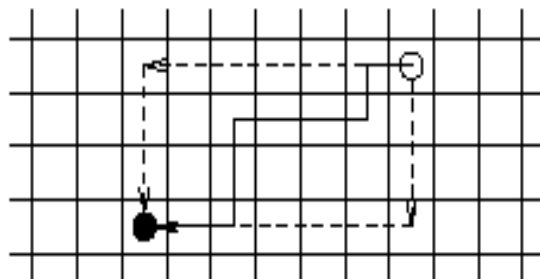

Fig 1: Location-dependent convolutions on the SOM Grid.

c. *Depth-first or Breadth-first search*

The policy of selecting the N_n best-scoring images as the ones to be shown to the user is valid when one or more areas of distinct positive response have been discovered. Concentrating the search on these areas, leading to a depth-first search on the SOM structure. Here the assumption is that the probability of finding more relevant images is high on the SOM grid. If this is not the case, it often is a better strategy to widen the scope of the search so that the user obtains a broader view of the database. For this purpose, we can use the mutual dissimilarity of the images as an alternative. This leads to a breadth-first type selection of images. BFS can be directly implemented using the SOMs and their image labels. The image label of a SOM unit is the image whose feature vector is nearest to the model vector of the map unit. So, it can be considered as a kind of average among the images mapped to that map unit. BFS can thus be implemented by returning only label images of map units on the intermediate SOM levels.

An important use for BFS is in the beginning of the queries if no initial reference images are available. In this mode of operation, it is important to return diverse images from the database. With SOM, a natural compromise is to use DFS on the bottommost SOM levels and BFS on the other levels. Upper SOM levels generally have sharper convolution masks, so the system tends to return images from the upper SOM levels in the early rounds of the image query. Later, as the convolutions begin to produce large positive values, the images on these levels are shown to the user. The images are thus gradually picked more and more from the lower map levels as the query is continued. The balance between BFS and DFS can be tuned with the selections of the size of the bottommost SOM level relative to the size of the image database and the window functions on different SOM levels. On the other hand, if one or more initial reference images are available, there is no need for an initial browsing phase. Instead, the retrieval can be initiated straight in the neighborhoods of the reference image on the bottommost SOM levels as they provide the most detailed resolution. The upper SOM hierarchy is

thus neglected and only the largest SOMs of each feature are used.

d. *Relations to other methods*

This method is based on clustering the images by using the well-known k-means. It yields better performance than the SOM. This is understandable as the SOM algorithm can be regarded as a trade-off between two objectives, namely clustering and topological ordering. This trade-off is dependent on the size of the SOM; the clustering property is dominant with relatively small SOMs whereas the topology of the map becomes more significant as the size of the SOM is increased.

The SOM attempts to represent the data with optimal accuracy in a lower-dimensional space of fixed grid of points, thus performing dimensionality reduction. This functionality is integral to our RF method as the computational complexity of measuring image similarity is drastically reduced by transforming the original high-dimensional space into a 2-D grid. This makes our method scale well to large databases.

II.   PERFORMANCE EVALUATION

For the retrieval, we use the global structure-based features Hglobal. For the current experiment we are interested in the performance of global structure features. So, we combine the global structure features with block-based features. The local method (BBF) computes pixel value distributions of equally sized blocks. Image matching is performed by comparing feature histograms out of a feature database. For evaluation, we use each image in both databases as the query image to compute the precision and recall. The images were obtained from the 1.000 image data-set and the results were retrieved from the 10.000 image collection. Images are ranked in decreasing similarities with the range between 0 and 1. Here 1 denoting an identical match with the query image.
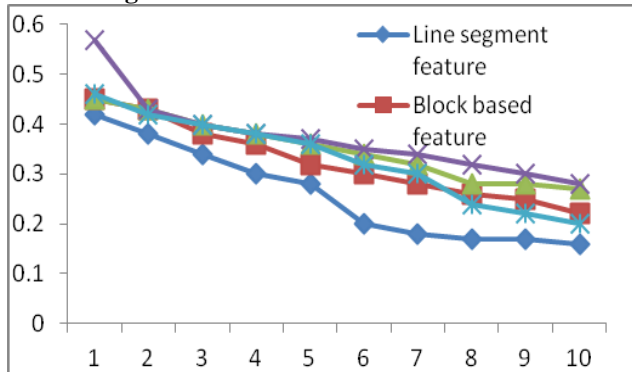
Color Image Retrieval

Fig 1Precision-recall graph for the 1.000 image data-set

The graph shows the average recall versus the number of retrieved images graph for 1.000 image database. It can be seen that the block-based features performs best for most of the classes, for the first 100

retrieved images. In terms of pixels it clearly shows that color dominates the information. Color is a stronger feature than structure. Though the curves of some classes look quite similar to each other, the overall precision recall graph proves the superior performance of our features. The figure shows that the combination of the local and global features reaches a precision of 100% until a recall of about 20 % is reached. An interesting observation can be found in the part between a recall of about 0.25 and 0.35.
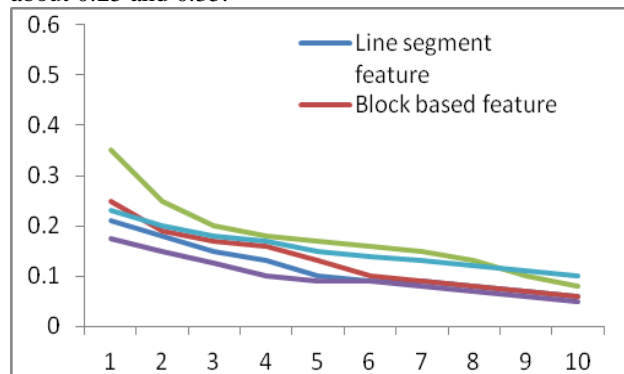

Fig 2 Precision-recall graph for the 10.000 image database

In the above figure we compute the precision recall curve for all other features. The combination of the global structure and block-based features performs better than the others methods.

Invariance Analysis

In an invariance analysis, we compute all features for each image for two newly created data-sets. To cope with these kinds of transformations, we have derived two data-sets from our 1.000 image database. In the *first* data-set, we increase for all images the brightness by 30% and decreased the saturation by 10%.
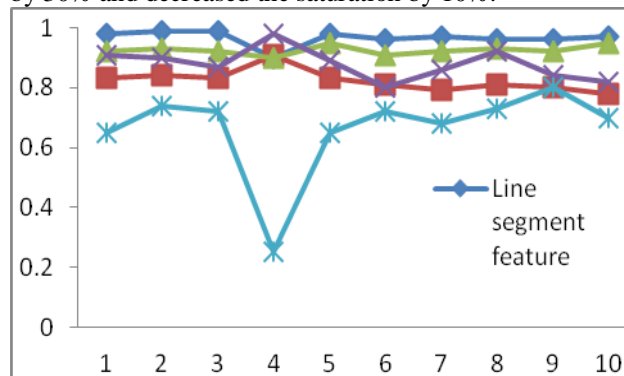

Fig 3 Robustness analysis of the 1.000 image database for different brightness.

The *second* data-set contains first with an additional rotation of all images by 90 degrees counter-clock wise. We take the histogram of each image from the original database and compute the similarity based on histograms obtained from the transformed image data-set,

where the brightness and saturation of every image was changed. In the second step, we computed the similarity of each image feature with respect to histograms obtained from the data-set.

The results of the comparison of two histograms are between [0, 1] with 1 indicating 100% invariance. The results are displayed in Figure 3, The curves show the variation of the features taken from the original image data and the data-set with a different brightness, saturation and rotation. A constant line at similarity equal to 1 indicates 100% invariance. From the figure we can conclude that the line segment feature performs best and the combination of block-based and line segment features second best and the color histogram the worst. The combination of our block-based and global line structure features is slightly worse than the invariant feature histograms. The comparison of the original image features with the features of just rotated images all features are equally robust to rotation.

Table 1: Averaged feature invariance representation.

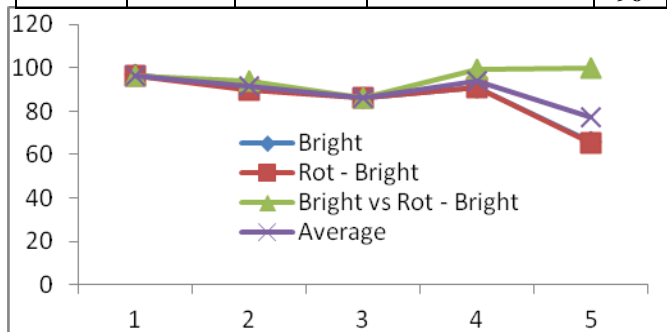| Features	Invariance of features under image transformations (%)			
	Bright	Rot - Bright	Bright vs Rot - Bright	Avg
LS	97.14	96.12	96.48	96.58
LS + BBF	90.99	89.67	93.92	91.53
BBF	86.25	86.25	86.25	86.25
IFH	91.05	90.83	99.49	93.79
Color	65.65	65.62	99.62	76.96

Fig 4 Averaged feature invariance representation

The precision recall graphs, averaged over the whole image database, reveal for both image data-sets that combined features perform better than the invariant feature histogram method and also outperforms the color histogram. It has to be said, that for some classes the features perform slightly worse, but averaged over all classes the precision recall graphs document a better performance.

III. CONCLUSION

The results show that structure is a prominent and highly discriminative feature. The features have been applied to various content-based image retrieval and classification problems. The first application was the classification and content-based image retrieval of watermark images, where the features have proven as powerful descriptors. The precision recall graphs have shown good results for various classes. Some classes performed worse due to the high intra-class variation. We have used support vector machines to learn the feature space distribution of our structure-based features for several images classes. For the classification we have used a multi-class SVM with a histogram intersection kernel. We have a proposal to extend the same by using feature selection and sophisticated weighting schemes that might improve the feature combination process, leading to better results.

REFERENCES

[1] Aksoy, S. and Haralick, R. M. (2011). Feature normalization and likelihood-based similarity measures for image retrieval, Pattern Recognition Letters 22(5): 563–582.

[2] Amsaleg, L. and Gros, P. (2009). Content-based retrieval using local descriptors: Problems and issues from a database perspective, Pattern Analysis & Applications 4: 108–124.

[3] Assfalg, J., Del Bimbo, A. and Pala, P. (2009). Using multiple examples for content-based image retrieval, Proceedings of the IEEE International Conference on Multimedia and Expo, Vol. 1, New York City, NY, USA, pp. 335–338.

[4] Beatty, M. and Manjunath, B. S. (2011). Relevance Feedback method for multidimensional scaling for content-based retrieval, Proceedings of IEEE International Conference on Image Processing, Vol. 2, Washington DC, USA, pp. 835–838.

[5] Brandt, S. (2009). Use of shape features in content-based image retrieval, Master's thesis, Laboratory of Computer and Information Science, Helsinki University of Technology.

[6] Csillaghy, A., Hinterberger, H. and Benz, A. O. (2008). Content-based image retrieval in astronomy, Information Retrieval 3(3): 229–241.

[7] Eakins, J. P. and Graham, M. E. (2009). Content-based image retrieval. Report to JISC technology applications programme, Technical report, Institute for Image Data Research, University of Northumbria at Newcastle. Available at:

[8] Fournier, J. and Cord, M. (2008). Long-term similarity learning in content-based image retrieval, Proceedings of

IEEE International Conference on Image Processing, Vol. 1, Rochester, NY, USA, pp. 441–444.

[9] Rui, Y., Huang, T. S. and Mehrotra, S. (2007). Content-based image retrieval with relevance feedback in MARS, Proceedings of IEEE International Conference on Image Processing, Santa Barbara, CA, USA, pp. 815–818.

[10] Sethi, I. K. and Coman, I. (2009). Image retrieval using hierarchical self-organizing feature maps, Pattern Recognition Letters 20(11): 1337–1345.

[11] Stricker, M. and Orengo, M. (2009). Similarity of color images, Storage and Retrieval for Image and Video Databases III, Vol. 2420 of SPIE Proceedings Series, San Jose, CA, USA, pp. 381–392.

[12] Sull, S., Oh, J., Oh, S., Song, S. M.-H. and Lee, S. W. (2008). Relevance graph-based image retrieval, Proceedings of the IEEE International Conference on Multimedia and Expo (ICME 2008), Vol. 2, New York City, NY, USA, pp. 713–716.

[13] R. Veltkamp and M. Tanase. Content-based image retrieval systems: A survey. In Oge Marques and Borko Furht, editors, In O. Marques and B. Furht (Eds.), Contentbased image and video retrieval, pages 47–101. Kluwer Academic Publishers, 2002.