Content Based Image Retrieval with Application of Clustering of Data and User Interaction

Aswathy AjithKumar
M.E (CSE),
CMS College of Engineering,
Namakkal, Tamilnadu, India.

Abstract - **The Content-based image retrieval uses the visual contents of an image such as color, shape, texture and spatial layout to represent and index the image. The feature vectors of the images in the database form a feature database. To retrieve images, users provide the retrieval system with example images or sketched figures. Retrieval systems have incorporated user's relevance feedback to modify the retrieval process in order to generate perceptually and semantically more meaningful retrieval results. However, existing relevance feedback-based CBIR methods usually request a number of iterative feedbacks to produce refined search results, especially in a large-scale image database. This is impractical and inefficient in real applications. In this paper, in terms of efficiency, the iterations of feedback are reduced substantially by using the navigation patterns discovered from the user query log.**

Keywords - **Content-based image retrieval, NRPF search technique, Query reweighting, Query point movement, Query expansion Feature extraction for clustering.**

1. INTRODUCTION

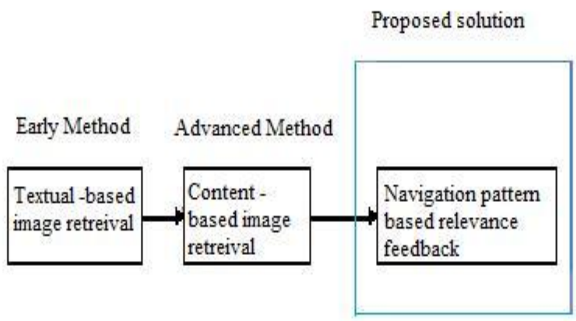
Content based image retrieval refine the query again and again by analyzing the specific relevant images picked up by the users. Especially for the compound and complex images, the users might go through a long series of feedbacks to obtain the desired images using current RF approaches. In fact, it is not practical in real applications like online image retrieval in a Large-scale image database. To solve this problem the iterations of feedback are reduced substantially by using the navigation patterns. In terms of effectiveness, our proposed search algorithm NPRF search makes use of the discovered navigation patterns and three kinds of query refinement strategies, Query Point Movement, Query Reweighting and Query Expansion to converge the search space toward the user's intention effectively. By using NPRF method, high quality of image retrieval on RF can be achieved in a small number of feedbacks.

Methods for Image Retrieval

Textual-based image retrieval has two problems: high-priced manual annotation and inappropriate automated annotation.

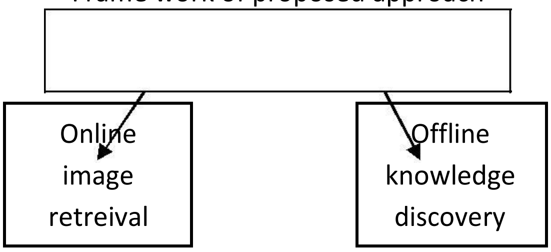
A) High-priced manual annotation: High-priced manual annotation cost is prohibitive in coping with a large-scale data set.

B) Inappropriate automated annotation: It yields the distorted results for semantic image retrieval.



In proposed approach it approximate an optimal solution to resolve the problems existing in current RF, such as redundant browsing and exploration convergence. To this end, the approximated solution takes advantage of exploited knowledge (navigation patterns) to assist the proposed search strategy in efficiently hunting the desired images. Generally, the task of the proposed approach can be divided into two major operations, namely offline knowledge discovery and online image retrieval. As depicted in the figure, each operational phase contains some critical components for completing the specific process. For online operation, once a query image is submitted to this system, the system first finds the most similar images without considering any search strategy, and then returns a set of the most similar images. The first query process is called initial feedback. Next, the good examples picked up by the user deliver the valuable information to the image search phase, including new feature weights, new query point, and the user's intention. Then, by using the navigation patterns, three search strategies, with respect to Query Point Movement (QPM), Query Reweighting (QR), & Query Expansion (QEX), are hybridized to find the desired images. Overall, at each feedback, the results are presented to the user and the related browsing information is stored in the log database. After accumulating long-term users' browsing behaviors, offline operation for knowledge discovery is triggered to perform navigation pattern mining and pattern indexing.

Frame work of proposed approach



II. ALGORITHM FOR NPRF SEARCH

The iterative search procedure can be decomposed into several steps as follows:

1. Generate a new query point by averaging the visual features of positive examples.

2. Find the matching navigation pattern trees by determining the nearest query seeds (root).

3. Find the nearest leaf nodes (terminations of a path) from the matching navigation pattern trees.

4. Find the top s relevant visual query points from the set of the nearest leaf nodes.

5. Finally, the top k relevant images are returned to the user.

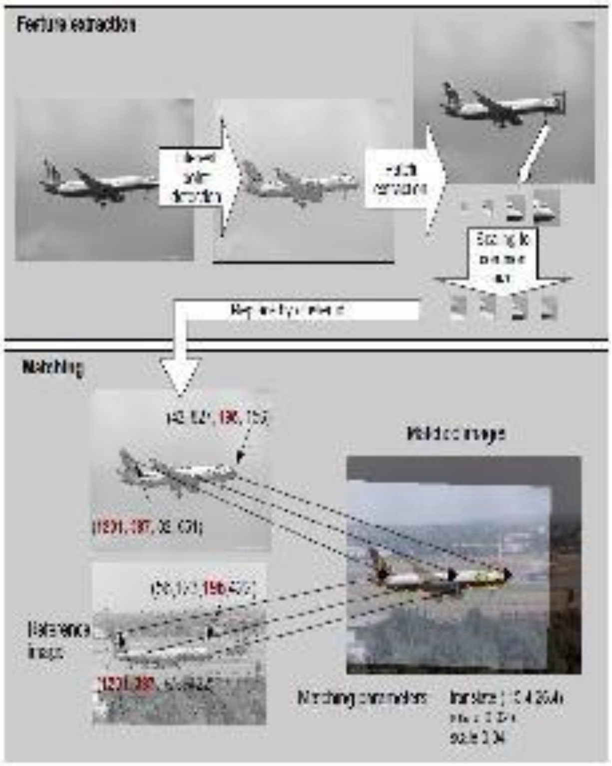
III. GEOMETRIC FEATURE EXTRACTION FOR CLUSTERING

By using the RAST algorithm, we are able to find the optimal matching for the equivalent of a fully connected patch-based model. Here match all given training images that contain the object of interest to the test image. This approach is analogous to nearest neighbour classification, using the RAST score as a similarity measure.

This procedure has the additional advantage that we determine the best-matching training image, which directly allows the use of the method in an object-based image retrieval scenario.

In the matching, we allow for a displacement of the patch positions by a predetermined number of pixels. The score describe the quality of a resulting matching is the number of patches that have been matched.

Fig 1 : Illustration of the presented approach: top box: detection of interest points; extraction of patches in multiple scales and scaling to a common size. Then, the extracted patches are replaced by the identifiers of their closest clusters. In the bottom box, the interest points, represented by vectors of cluster identifiers, are matched to interest points, represented equally, of a reference image. Corresponding cluster identifiers are printed in red, bold letters. The optimal matching and the according transformation parameters are obtained by applying the RAST algorithm. The final image shows the reference image overlaid on the best matching database image transformed according to the obtained transformation parameters.



Patch Extraction

To extract the image patches, first, all images are converted into gray scale and scaled to a common height of 225 pixels. The scaling is applied because in the database we use for evaluation, the Caltech database, the background images are smaller than the training images, which may aid some classifiers. Given an image, we extract patches of multiple sizes (7×7, 11×11, 21×21, 31×31 pixels) around up to 500 interest points obtained using the method proposed by Loupias. The use of patches of different sizes increases robustness to image scaling and allows to use visual clues that occur at different scales simultaneously. This procedure yields up to 2000 patches per image, 1730 on the average. The patches are allowed to extend beyond the image border, in which case the part of the patch falling outside the image is padded with zeroes. After the patches are extracted, they are scaled to a common size of 15×15 pixels to be able to determine a common code book for all extracted patches and to capture patch similarities across scale. A PCA dimensionality reduction is then applied to reduce the dimensionality of the data, keeping 40 coefficients. The first of these coefficients is discarded to achieve brightness normalization as it mainly encodes the overall image brightness. The patches from all training images are then jointly clustered with a Linde-Buzo-Gray algorithm using the Euclidean distance such that 2048 clusters are obtained. Then we discard all information for each patch except its closest corresponding cluster center identifier. It is possible to represent each extracted patch by scores to all cluster centers and thus reducing the amount of information loss by vector quantization, however this would incur much higher costs for finding corresponding points in the final matching algorithm and thus would lead to strongly increased runtimes while not expecting a big gain in accuracy.

Determining the Optimal Matching

We now outline the RAST algorithm that we use for the determination of the optimal matching of the patch sets obtained from two images. Assume as input the sets of patches R for the reference and S for the test image. Each patch p = (xp, yp, lp) is a triple of x-position, y-position, and label, where the label here consists of the vector quantizer output and the scale at which the patch was extracted. We are interested in finding the best transformation of the reference image to explain the patches observed in the test image. Here, we only consider the transformations translation, rotation, and scaling, although it is straightforward to use other sets of transformations. The transformations are characterized by a set of four parameters # 2 T, i.e. translation in x- and y-direction, rotation angle, and scale factor. Here, T is the set of all possible initial parameter combinations as detailed below. We find the maximizing set of parameters q(#, p, S) evaluates the goodness of fit for a given patch p and a set of parameters # to the patches in S by assigning a one in case of a match within a distance d0 that was set to d0 = 4 pixels in the experiments. The Euclidean distance between the position of patch p transformed using the parameters # and the position of patch p0 is denoted by d (#, p, p0) here. Note that other local quality functions that correspond e.g. to Gaussian distributions rather than to bounded error can easily be introduced into the algorithm. This maximization will be a complex task for most functional forms of Q. In many applications, such fits of parameters are carried out iteratively and heuristically, which involves the risk that the results found are only locally optimal solutions. Other methods include randomized approaches like e.g. random sample consensus . We employ a branch-and-bound technique to perform the maximization. This algorithm guarantees to find the globally optimal parameter set by recursively subdividing the parameter space and processing the resulting parameter hyper-rectangles in the order given by an upper bound on the total quality. Moreover, with small modifications, the algorithm allows us to efficiently determine the k best matches, not only the best match. We determine an upper bound on the quality of parameters in a hyper-rectangular region T.

We can now organize the search as follows:
1. Pick an initial region of parameter values T containing all the parameters that we are interested in. (For the experiments we used the following settings: x-translation ±200 pixels, y-translation ±100 pixels, angle ±0.1 radians, scale factor in [0.8,1.2].)

2. Maintain a priority queue of regions Ti, where we use as the priority the upper bound on the possible values of the global quality function Q for parameters # 2 Ti.
3.Remove a region Ti from the priority queue; if the upper bound of the quality function associated with the region is too small to be of interest, terminate. (When the upper bound of the quality is smaller than the value we are willing to accept as a match, we can be sure that no match that reaches this minimum quality can be re ached and can therefore end the algorithm.)

3. If the region is small enough to satisfy our accuracy requirements, accept it as a solution.

4. Otherwise, split region Ti along the dimension furthest from satisfying our accuracy constraints and insert the sub regions into the priority queue; continue at Step 3.

This algorithm will return the maximum quality match. To make the approach practical and avoid duplicate computations, we use a matchlist representation [16]. That is, with each region kept in the priority queue in the algorithm, we maintain a list (the matchlist) of all and only those patches that have the possibility to contribute with a positive local quality to the global quality. We maintain the list for each patch in the reference image. These matchlists will shrink quickly with decreasing size of the regions Ti. It is easy to see that the upper bound of a parameter space region Ti is also an upper bound for all subsets of Ti. When we split a region in Step 5, we therefore never have to reconsider patches in the children that have already failed to contribute to the quality computation in the parent and thus the match lists can be reused in the children. The running time of the algorithm is largely determined by two factors:

• The time necessary to determine max#2T Q(#,R, S). This time is bounded by the product of the sizes of the sets R and S and therefore linear in the number of patches in the model as mentioned above. Note that, due to the use of match lists as discussed above, the average number of comparisons is much smaller in each step. All other computations that are necessary in each subdivision step are much simpler and dominated by the determination of the upper bound.

• The number of times the initial region is split before a solution is reported. The interactions between the following variables influence this number:

– The dimensionality of the search space: the number of splits tends to grow approximately exponentially with the dimensionality. However, in the application presented here, this dimensionality is always fixed at four.
– The distribution of the patches in the images: the number of splits tends to decrease strongly if good matches are present.

– The number of matching labels between R and S: fewer matches allow to reduce the matchlists and to find the solution with fewer splits.

– The accuracy constraints imposed: if a more precise solution is needed, the number of splits increases.

IV. CONCLUSION

In the case of large database to make the system more profitable and efficient relevance feedback techniques were incorporated into CBIR such that more precise results can be obtained by taking user's feedbacks into account. The iterations of feedback are reduced substantially by using the navigation patterns discovered from the user query log.

Today, many successful approaches that address the problem of general object detection use a representation of the image objects by a collection of local descriptors of the image content. Commonly, SIFT features or just square sub images, called patches, are used to represent the parts. This paradigm has the advantage of being robust with respect to occlusions and background clutter in images. These models allow efficient determination of the maximum likelihood position of the object in the image.

Detection for a fully-connected part-based model has exponential complexity in the number of parts, while the method presented in this paper finds the optimal match of such a model in time linear in the number of parts considered. This is possible, because the search is organized over the transformation parameter space and simultaneously considers all parts. The search organization is only feasible because we implicitly factor the dependencies between the locations of the parts in the image into the four components x-translation, y-translation, rotation, and scale. If we wanted to include all general dependencies, the algorithm would effectively become exponential again, because of the exponential growth of the search space with the number of parameters.

REFERENCES

[1]D.H. Kim and C.W. Chung, "Qcluster: RelevanceFeedback Using Adaptive Clustering for Content-Based Image Retrieval," Proc.ACM SIGMOD, pp.599-610, 2003.
[2]J. Liu, Z. Li, M. Li, H. Lu, and S. Ma, "Human Behaviour Consistent Relevance Feedback Model for Image Retrieval,"Proc.15th Int'l Conf. Multimedia, pp. 269-272, Sept. 2007.
[3]P.Y. Yin, B. Bhanu, K.C. Chang, and A. Dong,"Integrating Relevance Feedback Techniques for Image Retrieval Using Reinforcement Learning," IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 27, no. 10, pp. 1536-1551, Oct. 2005.
[4]H. You, E. Chang, and B. Li, "NNEW: Nearest Neighbor Expansion by Weighting in Image Database Retrieval," Proc. IEEE Int'l Conf.Multimedia and Expo, pp. 245-248, Aug. 2001.
[5]G. Salton and C. Buckley, "Improving Retrieval Performance by Relevance Feedback," J. Am. Soc. Information Science, vol. 41, no. 4,pp. 288-297, 1990.
[6] H.T. Shen, S. Jiang, K.L. Tan, Z. Huang, and X. Zhou, "Speed up Interactive Image Retrieval," VLDB J., vol. 18, no. 1, pp. 329-343,Jan. 2009.
[7]V.S. Tseng, J.H. Su, B.W. Wang, and Y.M. Lin, "Web Image Annotation by Fusing Visual Features and Textual Information,"Proc. 22nd ACM Symp.Applied Computing, Mar. 2007.
[8]K. Vu, K.A. Hua, and N. Jiang, "Improving Image Retrieval Effectiveness in Query-by-Example Environment,"Proc. 2003 ACMSymp.Applied Computing, pp. 774-781, 2003.
[9]J.R. Smith and S.F. Chang, "VisualSEEK: A Fully Automated Content-Based Image Query System," Proc. ACM Multimedia Conf., Nov. 1996.
[10]L. Wu, C. Faloutsos, K. Sycara, and T.R. Payne, "FALCON: Feedback Adaptive Loop for Content-Based Retrieval," Proc. 26th Int'l Conf. Very Large Data Bases (VLDB), pp. 297-306, 2000.
[11]X.S. Zhou and T.S. Huang, "Relevance Feedback for Image Retrieval: A Comprehensive Review,"Multimedia Systems, vol. 8, no. 6, pp. 536-544, Apr.2003.
[12]Y. Rui, T. Huang, M. Ortega, and S. Mehrotra,"Relevance Feedback: A Power Tool for Interactive Content-Based Image Retrieval," IEEE Trans.Circuits and Systems for Video Technology,vol. 8, no.5, pp. 644-655, Sept. 1998.
[13]Y. Rui, T. Huang, and S. Mehrotra, "Content-Based Image Retrieval with Relevance Feedback in MAR S," Proc. IEEE Int'l Conf. Image Processing, pp.815-818, Oct. 1997.