IJARCCE

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

Reweighing Strategies for Relevance Feedback in CBIR

Renuka Devi SM

ECE Department, G. Narayanamma Institute of Technology and Science, Hyderabad, India

Abstract: CBIR relies on the searches purely is based on metadata obtained from the images like features and annotation information. The evaluation and the effectiveness of image search is more important and has been well-defined. Users of image databases often prefer to retrieve relevant images by categories. Unfortunately, images are usually indexed by low-level features like color, texture and shape, which often fail to capture high-level concepts well. To address this issue, relevance feedback has been extensively used to associate low-level image features with high-level concepts. Among all existing relevance feedback approaches, query movement and feature re-weighting have been proven to be suitable for large-scaled image databases with high dimensional image features. In this paper, we investigated different weight update schemes and compared the retrieval results. As far as feature re-weighting approaches are concerned, one of their common drawbacks is that the feature re-weighting process is prone to be trapped by suboptimal states. To overcome this problem, we introduce a disturbing factor, which is based on the Fisher criterion, to push the feature weights out of sub-optimum. Experimental results show that this method performances well compared to basic re-weighing methods

Keywords: Content based image retrieval, re-weighing features, Short term Learning, Long term learning.

I. INTRODUCTION

With the rapid development of internet technology, the transmission and access of image items have become easier and the volume of image repository is exploding. To facilitate the retrieval of image data, many content-based image retrieval (CBIR) systems have been developed. In text-based retrieval, images are indexed using keywords, subject headings or classification codes, which in turn are used as retrieval keys during search and retrieval. Text-based retrieval[1] is non-standardized because different users use different keywords for annotation. Text descriptions are sometimes subjective and incomplete because it cannot depict complicated image features very well. Examples are texture images that cannot be described by text. In text retrieval, humans are required to personally describe every image in the database, so for a large image database the technique is cumbersome, expensive and labour-intensive [2]. However in CBIR processing of query (image or graphics) involves extraction of visual features and/or segmentation and search in the visual feature space for similar images[3][4]. An appropriate feature representation and a similarity measure to rank pictures, given a query, are essential here.

There are two types of learning [5][6] as shown in figure 1. Firstly, Short-term learning (STL) (or intra-query learning) is learning within a single query session. Such learning is memory less and the acquiring of knowledge regarding the current query startsfrom the scratch. Secondly, if the interaction history of previous users over all past queries is potentially exploited to improve the retrieval performance for the current query it is termed as Long-term learning(LTL)(or inter-query learning).

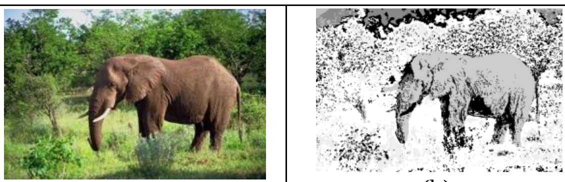
The broad categorization[my survey] of various STL and LTL techniques available in the literature are discussed below. The prolific activity of many comprehensive surveys[and short reviews on learning semantics in CBIR using Relevance feedback indicates the importance of the topic. The organisation of the paper is as follows. Section II discusses the visual descriptors used , Section III presents the various re-weighing methods. Section IV deals with experimental results and finally the section V gives the conclusions.

II. VISUAL DESCRIPTORS

The task of content-based image retrieval systems is to locate relevant images in image databases. To avoid the time consuming and subjective process of manual labelling of images, most image databases use content-based image retrieval techniques with low-level image features such as color, texture and shape. These databases represent each image feature using a feature vector, and retrieve images according to the distance (or similarity) between their feature vectors and those of the query. Although various effective low-level features have been proposed for content-based image retrieval, none of them can always capture high-level concepts successfully. To address this issue, relevance feedback has been used as a powerful tool to bridge the gaps between low-level features and high-level concepts.

Features are used to represent an image instead of using the original pixel values because of the significant simplification of image representation and the improved correlation with image semantics. No particular visual feature is most suitable for retrieval of all types of images. Colour visual feature is most suitable for describing and representing colour images. Texture is most suitable for describing and representing visual patterns, surface properties and scene depth. Shape is suitable for representing and describing boundaries of real world objects and edges. The color features include mean, variances, dominant color features [5]etc. These features are obtained by considering the HSV values of the image.

The 10 level quantized image is generated from the HSV image in order to extract the shape features and texture features. Initially the entire image pixels is set to all zeros then considering the top 10 most bins which has greater count of pixel values, each bin is assigned a grey level value from 1to10 such that the bin with more number of pixels gets the higher pixel value (say 10) now the 10 level grey scale image has the top 10 bins with 10 grey levels and rest of the bins with zeros which are of lower significance.



(a) (b)

Fig.2. 10 level quantized image generation

In figure 2 (b) is the 10 level quantized image obtained from the HSV image of (a). The shape features include occupancy ratio for pixel values. The texture features include entropy and contrast for the grey level co-occurrence matrix produced from the 10 level quantized image.

III. RELEVANCE FEEDBACK TECHNIQUES

The idea of Relevance Feedback was formulated by J. J. Rocchio in text retrieval [7] in 1971 and was adapted to image retrieval in 1997 in the work of Multimedia Analysis and Retrieval Systems(MARS) [9] and Mind Reader(MR) [8] as Query vector movement(QVM) or Query point movement(QPM). Most retrieval systems grant user the opportunity to give only positive feedback or positive, neutral and negative feedback or group similar images. Peng et al. [2] claim that with five relevance levels namely, highly relevant, relevant, no-opinion, non-relevant, and highly non-relevant, user's subjectivity is difficult to discriminate among non-relevant and highly non- relevant retrieved images. The author in [2] shows that a CBIR system with four levels i.e., excellent, fair, don't care and bad gives better performance than using five levels. Limitations of Relevance Feedback: RF has several limitations from its very nature such as[6]

a. Scarcity and imbalance of feedback examples:
b. High dimensionality of signatures:
c. Capability of capturing semantics:
d. Need measures that capture nonlinear distribution of relevant images:
e. Need fast scoring measures:

A typical CBIR system as shown in Figure 1 extracts visual attributes (colour, shape and texture) of each image in the database based on its pixel values and stores in a different database within the system called feature database, which is an offline process.The users usually formulate query image and present to the system. The system automatically extract the visual attributes of the query image in the same mode as it does for each database image, and then identifies images in the database whose feature vectors match those of the query image, and sorts the best similar objects according to their similarity value. If the user is not satisfied with the retrieval result, he/she can activate a Relevance Feedback process by identifying which retrieved images are relevant and which are no relevant. The system then updates the relevance information, such as the reformulated query vector, feature weights, and prior probabilities of relevance, to include as many user-desired images as possible in the next retrieval result. The process is repeated until the user is satisfied or the results cannot be further improved.

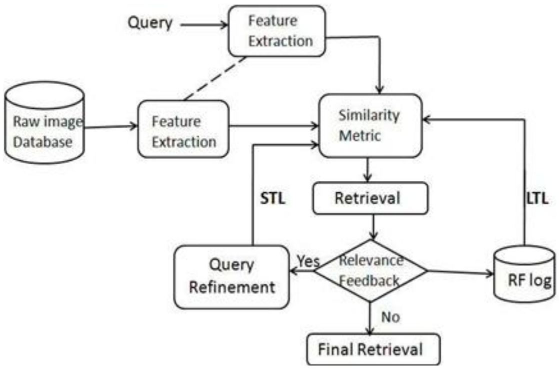


Fig. 1 Typical CBIR system with short-term and long-term learning facility[5]

Considering a CBIR system, a user submits a query image for retrieval purpose. Let the query image and a database image be represented by feature vectors $X = (x_1, x_2, \ldots, x_d)$ and$Y = (y_1, y_2, \ldots, y_d)$, respectively, where d is the number of selected features and $x_i$and y_i are the values of the i^{th} feature. The system derives the similarity between X and Y using the given similarity metric. The normalized Euclidean distance metricis generally used for this purpose [8].

$$\mathrm{Dist}(X, Y) = \sqrt{\sum_{i=1}^{d} \frac{(x_i - y_i)^2}{d}} \quad (1)$$

The database images that are the nearest neighbours of the query are then returned to the user. If the user is not satisfied with the retrieval result, he/she can activate an iterative RF process until satisfied. In the following subsections, the main existing RF techniques are presented.

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

A. Query vector modification

Let a user submit the i^{th} database image as the query and have experienced j RF iterations, and let $X_i^{(j)}$ denote the current query formulation. Also let the set of relevant images identified at the j^{th} iteration be R, the set of identified non relevant ima

The query vector modification (QVM) approach iteratively reformulates the query vector based on user's feedback in order to move the query toward a topological region of more relevant images and away from non relevant ones.

Ges be N [8]. For the $(j+1)^{th}$RF iteration, the method reformulates the query vector by

$$\left(X_i^{(j+1)}\right) = \alpha X_i^{(j)} + \beta \sum_{Y_k \in R} \frac{Y_k}{|R|} - \gamma \sum_{Y_k \in N} \frac{Y_k}{|N|}, (2)$$

Y_k are images that belong to region R or N, and$\alpha, \beta$and$\gamma$are the parameters controlling the relative weightof each component.

B. Feature relevance estimation

The feature relevance estimation (FRE)[3] approach also known as MARS(Multimedia Analysis and Retrieval System) assumes, for a given query, some specific features may be more important than others according to the users subjective[9]. Figure 3 shows the Query vector movement and Query expansion shown along with the iso-curves for each of the distance measures.

The relevance of each feature should be estimated before the similarity measure is derived. The most natural way of estimating the individual feature relevance is to assess the retrieval performance using each feature alone. First, the relevance of each feature is set equal to each other. To examine the retrieval ability of each feature, all the database images are projected onto the corresponding feature axis and the new closest images to the query are computed.

Then, the relevance of the feature is evaluated by counting how many of the newly retrieved images are identified as relevant. That is, the relevance weight w_i of feature i is proportional to $|R_i|$, where $|R_i|$denotes the number of relevant retrieved images obtained using feature ialone [8]. The larger the relevance weight, the better the retrieval ability of the tested feature, and thus the feature is more relevant to the query.

Finally, the feature relevance is used as a weight incorporated into the dissimilarity metric to express the degree of emphasis on the corresponding feature, viz,

$$Dist(X,Y) = \sqrt{\sum_{i=1}^{d} w_i (x_i - y_i,)^2 / \sum_{i=1}^{d} w_i} (3)$$

$$w_i = \frac{1}{\epsilon + \sigma_{rel.i}^k}, \epsilon = 0.0001 (4)$$

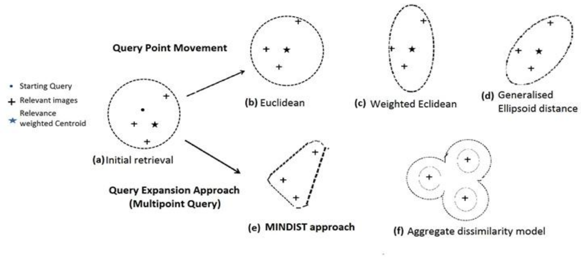


Fig. 3: Query vector movement and Query expansion shown along with the iso-curves for each of the distance measures

C. Re-weighting Type-1

When there is no RF, equal weight values are used for each feature component. With RF, these weights are updated using feedback samples. First, we used the following weight value:

$$w_i^{k+1} = \frac{\epsilon + \sigma_{N_r.i}^k}{\epsilon + \sigma_{rel.i}^k}, \epsilon = 0.0001 \qquad (5)$$

Here, $\sigma_{N_r,i}^k$ is standard deviation over the $N_r$retrieved images and $\sigma_{rel,i}^k$ is the standard deviation over the relevant images in k^{th}iteration. If a feature component has smaller variation over the relevant samples then it should get higher weight as this represents the relevant samples better in the feature space [1]. In the numerator of above eqn., we used standard deviation over $N_r$as the variation over the entire database remains unchanged with iteration and thus does not provide any extra information. However, in each iteration a new set of images is likely to be retrieved and a new $\sigma_{N_r,i}^k$ obtained. A small value of ϵ is used to avoid computational problem of $\sigma_{rel,i}^k$ being zero when no similar image (other than the query itself is retrieved) is retrieved. The value of ϵ is chosen to be 0.0001 so that it does not affect the weight values significantly.

D. Re-weighting Type-2

Wu and Zhang [3] used both relevant and non-relevant images to update weights. They used a discriminant ratio to determine the ability of a feature component in separating relevant images from the non-relevant ones.

Let the set of relevant and irrelevant images after the k^{th} relevance feedback iteration be R^k and U^k, respectively, where$I^k = R^k \cup U^k$. For all images inR^k, we stack their i^{th}feature component in to the set$F_i^{k,R}$. Similarly, we stack the i^{th} feature component from the irrelevant image set U^k in to the set$F_i^{k,U}$. The dominant rangeof R^k on the axis of the i^{th}feature component is defined as:

$$\Phi_i^k = [\phi_i^{k,1}, \phi_i^{k,2}], \qquad (6)$$

With $\phi_i^{k,1} = Min(F_i^{k,R})$ and $\phi_i^{k,2} = Max(F_i^{k,R})$,

ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

The confusion set of the i^{th} feature component after the k^{th} iteration is given by:

$$\Psi_i^{k,U} = \{\forall f_i^k \ / \ f_i^k \in \Phi_i^k and f_i^k \in F_i^{k,U}\} \qquad (7)$$

The confusion set is the subset of $F_i^{k,U}$ that falls into the dominant range Φ_i^k after the k^{th} iteration. The discriminant ratio of the i^{th} feature component is defined as:

$$\delta_i^k = 1 - \frac{\Sigma_{l=1}^k |\Psi_i^{l,U}|}{\Sigma_{l=1}^k |F_i^{l,U}|} \qquad (8)$$

The discriminant ratio indicates the ratio of irrelevant images located outside of the dominant range Φ_i^k over all irrelevant images, and it shows the ability of feature component i in separating irrelevant images from relevant ones.

The value of δ_i^k lies between 0 and 1. It is 0 when all non-relevant images are within the dominant range and thus, no weight should be given for that feature component. On the other hand, when there is not a single non-relevant image lying within the dominant range, maximum weight should be given to that feature component. [3]

$$w_i^{k+1} = \frac{\delta_i^k}{\epsilon + \sigma_{rel.i}^k} \qquad (9)$$

Unlike the Mars feature re-weighting approach which only uses the factor $\sigma_i^{k,R}$ learned from relevant images, we also integrate irrelevant images in our feature re-weighting method by introducing the factor δ_i^k , which indicates the distribution pattern of the irrelevant images on each feature component.

Re-weighting type-2 method and the Mars approach [14] are compared in Figure 3. As shown in Figure 3, the Mars approach assigns large weights to all feature components that allocate relevant images together.

However, many relevant images in the database may share the same irrelevant features such as the background. By inappropriately assigning large weights to irrelevant feature components as shown in the case 2 of Figure 3, the Mars approach may bring in more irrelevant images in the next iteration.

On the other hand, type-2 approach makes good use of irrelevant image as well as relevant ones to calculate the weights of each feature component. Hence, type-2 approach successfully distinguishes case 1 from case 2, and only assigns large weights to the former – which clusters all relevant images together and scatter the irrelevant images away from the relevant ones. For components whose distribution patterns are similar with case 3, both the Mars approach and type-2 method assign small weights to them.

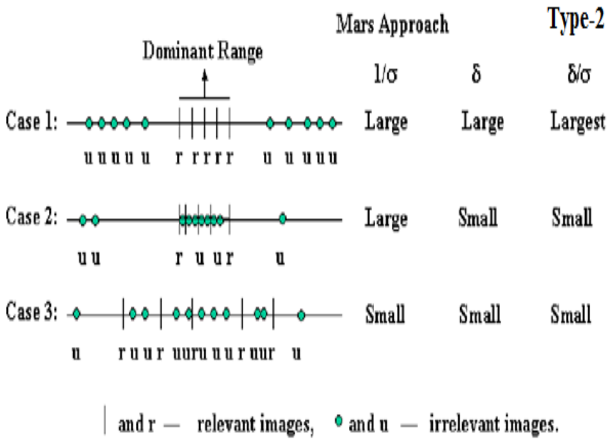


Fig.4. The weight value of a feature component in MARS and Type-2 approaches[3].

Re-weighting type-2 method and the Mars approach [14] are compared in Figure 4. As shown in Figure 3, the Mars approach assigns large weights to all feature components that allocate relevant images together. However, many relevant images in the database may share the same irrelevant features such as the background. By inappropriately assigning large weights to irrelevant feature components as shown in the case 2 of Figure 4, the Mars approach may bring in more irrelevant images in the next iteration.

On the other hand, type-2 approach makes good use of irrelevant image as well as relevant ones to calculate the weights of each feature component. Hence, type-2 approach successfully distinguishes case 1 from case 2, and only assigns large weights to the former – which clusters all relevant images together and scatter the irrelevant images away from the relevant ones. For components whose distribution patterns are similar with case 3, both the Mars approach and type-2 method assign small weights to them.

E. Re-weighting Type-3

In order to maximize the benefits in separating relevant images from the non-relevant ones, we introduced weight-type3 where we combined the above discriminant ratio with the weight factor. This resulted in weight-type 3 and our experimental results also demonstrated the synergy of the weight-type 1 and weight-type 2. [3].

$$w_i^{k+1} = \delta_i^k * \frac{\epsilon + \sigma_{N_r.i}^k}{\epsilon + \sigma_{rel.i}^k} \qquad (10)$$

F. Disturbing Factor

As far as feature re-weighting approaches are concerned, one of their common drawbacks is that the feature re-weighting process is prone to be trapped by suboptimal states during the relevance feedback. To address this

 DOI 10.17148/IJARCCE.2016.511118

problem, we introduce a disturbing factor, which is based on the Fisher criterion [3], to push the feature weights out of sub-optimum. The relevance feedback may converge with only few or even just one relevant image. This is because the feature weights are trapped in some suboptimal state, which can be detected by the following conditions:

- $|R^k| = 1$, for all $k \geq 1$.
- $R^k = R^{k-1}$ for all $k > 1$.

When either one of the above conditions holds, we think the feature weights is trapped in a suboptimal state. To push the feature weights out of the sub-optimum, we use a disturbing factor measured from the scatter of the classes of relevant images and irrelevant ones.

In reality, irrelevant images tend to be multi-model, but we simplify the situation by regarding them as one class, since we just wish to resume the feature re-weighting process when it's stuck.

For all images in R^k, we stack their s^{th}component of feature $m$in to the set$F_{m,s}^{k,R}$. Similarly, we stack the s^{th} component of feature m from the irrelevant image set U^k in to the set$F_{m,s}^{k,U}$. Denote the mean values of $F_{m,s}^{k,R}$ and $F_{m,s}^{k,U}$ by $\mu_{m,s}^{k,R}$ and$\mu_{m,s}^{k,U}$, respectively, and let the standard deviation of $F_{m,s}^{k,U}$ and$\sigma_{m,s}^{k,U}$. The disturbing factor is given by:

$$\lambda_{m,s}^{k} = \frac{\left(\mu_{m,s}^{k,R} - \mu_{m,s}^{k,U}\right)^2}{\left(\sigma_{m,s}^{k,R}\right)^2 + \left(\sigma_{m,s}^{k,U}\right)^2} \qquad (11)$$

The above formula is the Fisher criterion [6], [13], which has been extensively used in measuring the scatter between two classes. The weight $w_{m,s}$ is then updated by:

$$w_{m,s}^{k+1} = \lambda_{m,s}^{k} \times w_{m,s}^{k}. \qquad (12)$$

After each feedback iteration, the feature vectors of the query are set to the average values of all relevant feature vectors. Its optimality is proven by Ishikawa [15]. The overall distance between image $i_n^k \in I^k$ and query q^k is given by:

$$D_n^k = \vec{e^k} \times \left[d_{n,1}^k, \dots, d_{n,m}^k, \dots, d_{n,m}^k\right]^T \qquad (13)$$

Where $\vec{e^k} = \left[e_1^k, \dots, e_m^k, \dots, e_M^k\right]$and e_m^k is the importance of m^{th} feature, and $d_{n,m}^k$ is given by Eucledian distance. This distance scheme indicates that an image must be similar to the query in all features: color, texture and shape to be considered as relevant. As proposed by Rui et. al. [10], the weight of the m^{th} feature is calculated by:

$$e_m^k = \sum_{l=1}^{M} \frac{w_m^k}{w_l^k}, \qquad (14)$$

Where $w_l^k (l = 1, \dots, M)$ is the total weight for the feature l of query q^k and those of relevant images after the k^{th} iteration. M=3, since we have considered color, shape and texture features.

IV. PRECISION AND PR GRAPHS

We used precision as a measure of system performance which is given by the following formula:

$$\text{Precision} = \frac{\text{No. ofrelevantimages}}{\text{No. ofretrievedimages}}$$

Precision and recall are used to evaluate the performance of the proposed approach. Precision is the number of the retrieved relevant images over the total number of retrieved images, and recall is the number of the retrieved relevant images over the total number of relevant images in the database. To calculate precision and recall, only those retrieved images from the same semantic category as the query are counted as relevant.

The number of images returned to the user in each relevance feedback iteration is called scope. The precision and PR graphs for four retrieval techniques with scope=25 are given below.The figure 4 gives the analysis of relevance feedback techniques that we have used.

The number of images returned to the user in each relevance feedback iteration is called scope. The precision and PR graphs for four retrieval techniques with scope=25 are given below. The figure 4 gives the analysis of relevance feedback techniques that we have used. Improvement can be seen in the precision (%) from QVM method to Re-weighting type-3 with disturbing factor and maximum precision obtained is 77.9%. The table 2 gives the precision values in percentage for each technique in each iteration.

The number of images returned to the user in each relevance feedback iteration is called scope. The precision and PR graphs for four retrieval techniques with scope=25 are given below. The figure 4 gives the analysis of relevance feedback techniques that we have used. Improvement can be seen in the precision (%) from QVM method to Re-weighting type-3 with disturbing factor and maximum precision obtained is 77.9%. The table 2 gives the precision values in percentage for each technique in each iteration

Figure 5 is a PR(Precision-Recall) graph for all the five methods. Where recall helps to analyse how early the image is retrieved in a given method. Figure 6 to figure 10 shows the retrieval results for the query image database number 541 for various relevance feedback techniques for scope=25. From these figures we can observe that the number of relevant images increases from QVM to re-weighting type-3 technique.

DOI 10.17148/IJARCCE.2016.511118

IJARCCE
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940
International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016

TABLE 1 TABULATION OF ALL THE RELEVANCE FEEDBACK TECHNIQUES

Relevance feedback techniques	**Query for (k+1)th iteration ($X_i^{(k+1)}$)**	**Weight ($w_i^{(k+1)}$)**	**Distance Measure**
Query Vector Modification (QVM)	$\alpha X_i^{(k)} + \beta \sum_{Y_k \in R} \frac{Y_k}{\lvert R \rvert} - \gamma \sum_{Y_k \in N} \frac{Y_k}{\lvert N \rvert}$	Unity	Normalized Euclidean Metric
Feature Relevance Estimation (FRE)	$\text{mean}\left(\text{Relevant\_images}_i^{(k)}\right)$	$\frac{1}{\epsilon + \sigma^k_{rel,i}}$	Weighted Euclidean Metric
Re-weighting Type-1	$\text{mean}\left(\text{Relevant\_images}_i^{(k)}\right)$	$\frac{\epsilon + \sigma^k_{N_r,i}}{\epsilon + \sigma^k_{rel,i}}$	Weighted Euclidean Metric
Re-weighting Type-2	$\text{mean}\left(\text{Relevant\_images}_i^{(k)}\right)$	$\frac{\delta^k_i}{\epsilon + \sigma^k_{rel,i}}.$	Weighted Euclidean Metric
Re-weighting Type-3	$\text{mean}\left(\text{Relevant\_images}_i^{(k)}\right)$	$\delta^k_i * \frac{\epsilon + \sigma^k_{N_r,i}}{\epsilon + \sigma^k_{rel,i}}$	Weighted Euclidean Metric
Re-weighting Type-3 with Disturbing Factor	$\text{mean}\left(\text{Relevant\_images}_i^{(k)}\right)$	$\frac{\left(\mu^{k,R}_{m,s} - \mu^{k,U}_{m,s}\right)^2}{\left(\sigma^{k,R}_{m,s}\right)^2 + \left(\sigma^{k,U}_{m,s}\right)^2} \times w^k_{m,s}$	Weighted Euclidean Metric

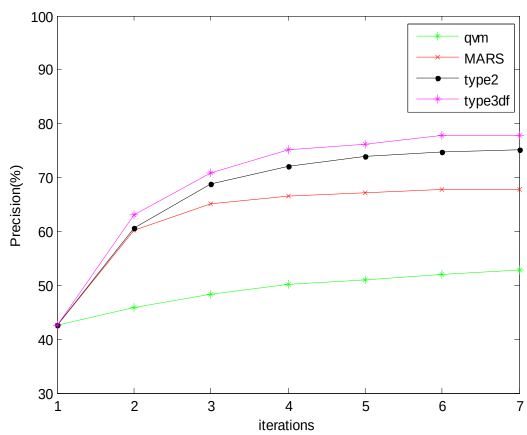


Fig. 4 Improvement in precision at scope=25

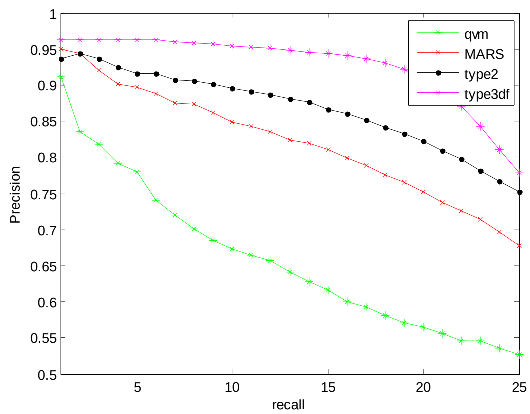


Fig. 5 Precision Vs Recall graph

TABLE 2
COMPARISON OF PRECISION (%), SCOPE = 25

Retrieval Technique	QVM	FRE	Type-2	Type-3 with disturbing factor
Iteration1	42.6	42.6	42.6	42.6
Iteration2	45.8	60.2	60.6	63
Iteration3	48.3	65	68.8	70.9
Iteration4	50.1	66.6	72	75
Iteration5	51	67.2	73.9	76.3
Iteration6	52.1	67.8	74.7	77.9
Iteration7	52.8	67.8	75.2	77.9

Query:541

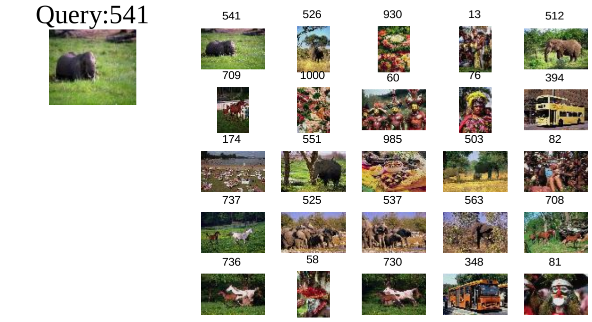
Relevant images:8

Fig. 6 Retrieved images using Euclidean distance for database image number 541

IJARCCE

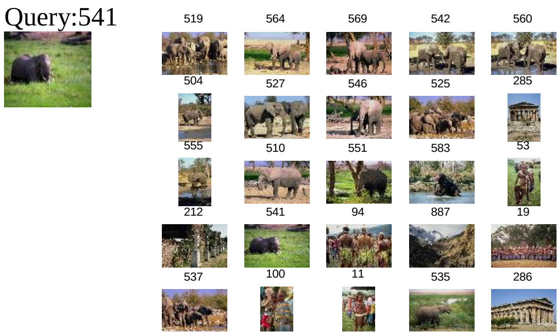
ISSN (Online) 2278-1021
ISSN (Print) 2319 5940

International Journal of Advanced Research in Computer and Communication Engineering
ISO 3297:2007 Certified
Vol. 5, Issue 11, November 2016



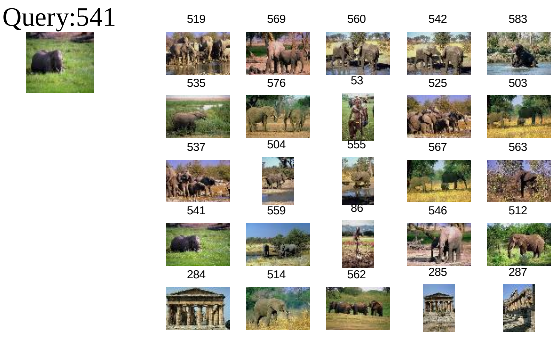
Relevant images:11

Fig.7 Retrieved images using QVM for database image number 541



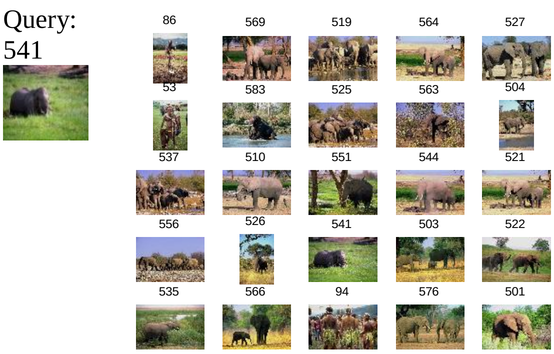
Relevant images:16

Fig.8 Retrieved images using FRE for database image number 541



Relevant images:20

Fig.9 Retrieved images using type-2 for database image number 541



Relevant images:22

Fig.10 Retrieved images using type-3 with DF for database image no.541

V CONCLUSIONS

Relevance feedback is very important to bridge the semantic gap between low-level-features and the high-level semantics , the humans perceive from the image. In this paper we have implemented four types of reweighing methods , Query vector movement, feature re-weighing,type-2and type-3 with distribution factor.The results are analysed over Wang database of 1000 images, and the retrieval performance and precision recall is plotted. It is observed that type-3 with distribution factor performs well because of fisher criterion

REFERENCES

[1] Michael EziashiOsadebey, "Integrated Content-based Image Retrieval using Texture, Shape and Spatial Information", Master Thesis Report in Media Signal Processing, February 2006.

[2] Peng-Yeng Yin, Shin-Huei Li, "Content-based image retrieval using association rule mining with soft relevance feedback", Journal of Visual Communication and Image Representation, April 2006.

[3] Yimin Wu and Aidong Zhang, "A Feature Re-Weighting Approach for Relevance Feedback in Image Retrieval, Rochester New York, September 2002.

[4] Gita Das, Sid Ray, and Campbell Wilson, "Feature Re-weighting in Content-Based Image Retrieval", New Zealand 2005 Conference (IVCNZ 2005), Dunedin New Zealand, November 2005.

[5] S. M. Renuka Devi and ChakravarthyBhagvati. 2011. A novel method for image retrieval using relevance feedback and unsupervised clustering. In Proceedings of the Fourth Annual ACM Bangalore Conference (COMPUTE '11). ACM, New York, NY, USA 4-6

[6] Renuka Devi SMlearning in cbir IJRET: International Journal of Research in Engineering and Technology

[7] J. J. Rocchio, Relevance feedback in information retrieval, The SMART Retrieval System–Experiments in Automatic Document Processing. Prentice Hall, New Jersey 313–323, 1971.

[8]]Liu, Ying, et al. "A survey of content-based image retrieval with high-level semantics." Pattern recognition 40.1 (2007): 262-282.

[9] Y. Ishikawa, R. Subramanyay, and C. Faloutsosz, "Mindreader: Querying databases through multiple examples," In Proc. 24rd International Conference on Very Large Data Bases, pp. 422–431, 1998.

[10] Y. Rui, T. S. Huang, and S. Mehrotra, "Content-based image retrieval with relevance feedback in MARS," in In Proceedings of International Conference on Image Processing, vol. 2, oct 1997, pp. 815–818.

[11] Liu, Guang-Hai, and Jing-Yu Yang. "Content-based image retrieval using color difference histogram." Pattern Recognition 46.1 (2013): 188-198.

[12] [12]Raju, U. S. N., et al. "Content Based Image Retrieval on Hadoop Framework." 2015 IEEE International Congress on Big Data. IEEE, 2015.

[13] Ji Wan, Dayong Wang, Steven Chu Hong Hoi, Pengcheng Wu, Jianke Zhu, Yongdong Zhang, and Jintao Li. Deep Learning for Content-Based Image Retrieval: A Comprehensive Study. In Proceedings of the 22nd ACM international conference on Multimedia (MM '14). ACM, New York, NY, USA, 157-166.