Automatic Categorization of Image Regions using Dominant Color based Vector Quantization

Md Monirul Islam, Dengsheng Zhang, Guojun Lu

Gippsland School of Information Technology, Monash University
Churchill VIC 3842, Australia
{md.monirul.islam,Dengsheng.zhang,Guojun.lu}@infotech.monash.edu.au

***Abstract*—This paper proposes a dominant color based vector quantization algorithm that automatically categorizes image regions. In contrast to the conventional vector quantization algorithm, the new algorithm effectively handles variable feature vectors like dominant color descriptors. Furthermore, the algorithm is guided by a novel splitting and stopping criterion which is specially designed for dominant color descriptors. This criterion helps the algorithm not only to learn the number of clusters, but also to avoid unnecessary over-fragmentations of region-clusters. Experimental result shows that the proposed approach categorizes image-regions with very high accuracy.**

Keywords: Dominant Color Descriptor, Vector Quantization, Content Based Image Retrieval.

I. Introduction

With decades of research in content based image retrieval (CBIR), it has been shown that semantic gap is one of the main reasons behind the poor performance in most CBIR systems. To reduce this semantic gap, recent focus in this area is to use automatic image classification and annotation. However, most of the efforts are either too broad, or too narrow, which forces images to certain categories [1] [2]. The problem with these approaches is due to they try to analyse image globally instead of analysing semantic components included in an image. Images normally contain multiple objects, as shown in Fig. 1. Intuitively it is more reasonable and also easier to learn the semantics of the components (regions) of an image than to learn the semantic of the image as a whole. Once the semantics at region level are learnt, image-semantics can be built up hierarchically. However, to learn region-semantics directly from low level features is not desirable because of poor discriminative power of low level feature representation which often results in different representations for similar regions and similar representation for different regions. There needs a mechanism that can effectively use these intra-class and inter-class classification errors to learn the semantics for regions. Machine learning approaches, like decision tree or decision rule, have the potential to do so [3] [4]. However the probability distributions required for these approaches is available only when regions are described with features from a finite set or dictionary. The success of semantic learning depends on how effectively a dictionary of representative features is created.

Fig. 1. Images with multiple concepts. (a) Image with concepts of 'tiger', 'water', 'sand', and 'grass'. (b) Image with concepts of with 'sky', 'water', 'sail', 'boat', and 'people'.

This paper presents a dictionary building method using dominant color based vector quantization algorithm. Basically a set of training regions are classified into different groups based on their visual similarities. A representative feature is then computed to represent each group. Fig. 2 shows the process. Thus each representative color feature in the dictionary identifies a group of visually similar regions.

In this work, regions are initially represented with MPEG-7 dominant color descriptors (DCD). A DCD consists of a few colors and the percentages of pixels belonging to those colors. We show that a DCD is *sufficient* as well as *necessary* to represent the color information of a region. Then a set of training DCDs is used to create a dictionary (or codebook) of representative DCDs.

The process is challenging due to variable feature dimensions of DCDs. Recently, Liu *et. al* [5] proposed to use a fixed number of color templates to find representative features. Each template is defined as the average color of a few manually selected sample regions belonging to same semantic concept. However it is not practical to assume the number of representative features in advance.

In contrast to the template approach proposed by Liu *et. al*

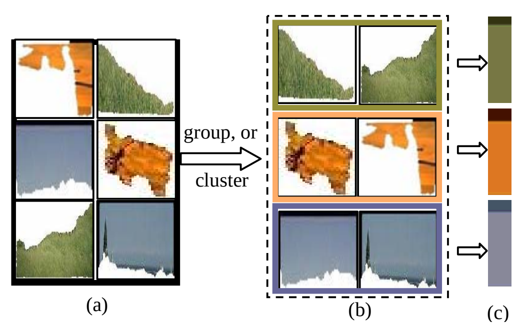

Fig. 2. Finding dictionary of representative features for regions. (a) Regions (non black) in database. (b) Visually similar regions grouped together by clustering method. (c) Each group of regions are represented with a DCD feature vector.

[5], Jing et. al [6] propose to use automatic clustering algorithm like Vector Quantization (VQ) to generate a set of representative features. The VQ algorithm groups similar objects (feature vectors) in same cluster and different objects in different clusters. The centroid of each cluster is regarded as a representative feature. They apply conventional VQ algorithm to form a codebook from region color features such as color moments. However, this approach assumes the number of codewords in advance and feature vectors have the same dimension. Their regions are presented with the first two color moments.

The template approach as well as the conventional VQ algorithm cannot be directly applied to features with variable dimensions, like dominant colors descriptors. First of all, both of these approaches find a representative feature vector by simply averaging the member feature vectors of the corresponding cluster which is not possible when features have variable dimensions. Furthermore, the conventional VQ algorithm causes over-fragmentation of clusters, because it splits each cluster in subsequent iteration even if a cluster already consists of similar objects and does not need further splitting. The reason behind this over-fragmentation is due to the predefined number of clusters.

This paper presents a modified VQ algorithm that addresses the above mentioned issues. The modified algorithm successfully finds a set of representative DCDs that can be used to represent all regions. As each representative feature identifies a group of regions, the effectiveness of these representative features is measured as to how accurately they can categorize image-regions. Therefore the accuracy of region categorization is used to measure the quality of the dictionary for region representations and to tell the accuracy of semantic learning for regions. The major contributions of this paper are:

- Application of VQ on *features with variable dimensions*
- Overcome the *over-fragmentation* problem of conventional VQ algorithm

The rest of the paper is organized as follows. Section II describes image segmentation and extraction of dominant color descriptors for regions. Dominant color based Vector Quantization is described in Section III. The performance of the dictionary in region categorization is presented in Section IV. The paper is concluded in Section V.

II. Image Segmentation and Dominant Color Features

Database images are segmented into regions using one of the state-of-the-art segmentation algorithms, JSEG [7]. The database consists of 5,100 images from Corel Photo gallery. The images are divided into 51 categories. Each category consists of 100 images and is dominated by a certain ground truth. The ground truths of these categories include Africa, ape, Australia, balloon, beach, bear, bird, butterfly, car, cactus, copter, deer, desert, dog, elephant, fighter plane, fireworks, flower, fox, heritage, horse, house, landscape, mountain, nature, ocean, rock, sailing, sunset, tiger, tree, etc. However, most of the images contain multiple objects as shown in Fig. 1. JSEG segmentation algorithm segments these images into

(a) (b) (c)

Fig. 3. DCD representation of regions. Regions in (a), (b) and (c) need 2, 3, and 4 colors

36,692 regions. As segmented regions normally have relatively homogeneous color and texture, there is no need of high dimensional feature vectors as needed in image representations. Fig. 3 shows examples of regions which can be represented by two to four colors. Later in this section, we show that only a small percentage of regions require more than four colors. Thus a few dominant colors are enough to represent a region. In contrast to the single color based representation in Liu *et. al* [5], Fig. 3 shows that some regions cannot be represented by only a single color. E.g., the tiger region in Fig. 3(c) cannot be represented with a single dominant color. Based on these arguments, a few dominant colors are *sufficient* but more than one color is *necessary*. In this work MPEG-7 dominant color descriptor (DCD) [8] is used to represent a region. The DCD for a region is defined as,

$$DCD = \{(c_i, p_i)\}, \quad (i = 1, 2, 3, \cdots, N)$$

where, N is the number of dominant colors in a region,

c_i is a dominant color represented as a vector in a color space, and

p_i is a fraction of pixels in the region belonging to color c_i.

p_i's are normalized into [0 1] and the sum of all p_i's in a DCD is 1. The distance between any two colors c_i and c_j in a DCD, is greater than a preset threshold, T_d. T_d is the maximum Euclidian distance for any two points in a color space to be considered similar color. The recommended value for T_d in CIELUV color space is any value between 10 and 20. T_d determines N, the number of colors in a region and the suggested value of N should be in the range of 1 to 8.

A. Dominant Color Extraction Procedure

The dominant colors descriptor for an image-region is

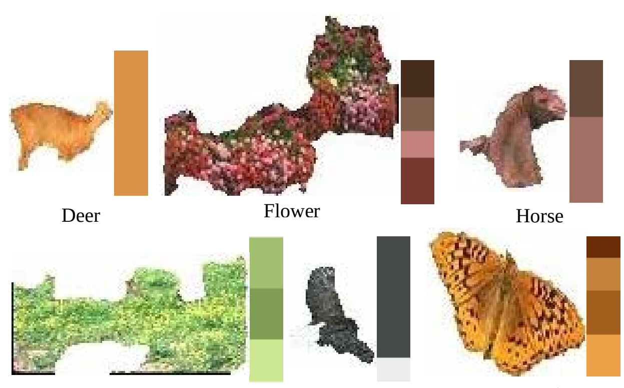

Fig. 4. Segmented regions and their dominant colors. Colors are shown according to their percentages.

extracted by using an iterative procedure, called Generalized Lloyd Algorithm (GLA) [9]. The objective is to divide the pixels among few clusters so that the distance between the centroids of any two clusters is greater than T_d. These centroids are regarded as the dominant colors for the region. Then the percentages of pixels in region belonging to different dominant colors are computed. The dominant colors along with their percentages form the dominant color descriptor for the region.

In this work, CIE LUV color space is used and the value of T_d is fixed at 20. Then 36,692 dominant color descriptors are extracted for 36,692 regions. Fig. 4 shows some sample regions and their DCDs.

B. *Define the Maximum Number of Dominant Colors for a Region*

MPEG-7 recommends that a region may have one to eight dominant colors. Thus a maximum of eight dominant colors is extracted for a region. However, the statistics of Fig. 5 shows that only 1.3% of the regions need more than 4 dominant colors. Therefore, without loss of significant information the feature extraction algorithm is then restricted to extract up to four dominant colors for a region.

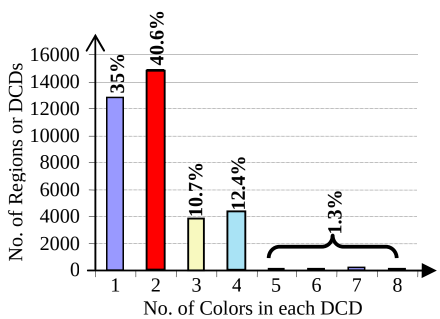

Fig. 5. Statistics of DCD features from 36692 regions. Only 1.3% regions need more than 4 colors.

C. *Clean up Unrelated Colors from a Region*

In many cases, regions contain some pixels from surrounding regions due to segmentation error, especially in the boundary area. Fig. 6(a) shows some examples of regions which includes such erroneous pixels. The colors of these pixels are significantly different from the colors of the rest of the corresponding regions. These pixels cause the number of dominant colors to be increased for these regions. If these pixels are removed, the region's color representation becomes more accurate. Although, it is impossible to remove these pixels during segmentation, it is possible to use a threshold to filter out unrelated colors. If the percentage, p_i of a color, c_i in DCD is less than the threshold value, we assume that this color comes from erroneous pixels and is discarded. In this work, this threshold is set at 0.1. The percentages of the remaining colors are normalized so that the sum of these fractions is 1. Fig. 6(b) shows the DCDs after discarding the unrelated colors.

(a) (b)

Fig. 6. Clean up DCDs. (a) Three regions and their dominant color descriptors. Each of them contains one or two dominant colors from erroneous pixels due to segmentation error. (b) The corresponding DCDs after discarding unrelated colors.

III. Dominant Color Based Vector Quantization Of Region Color Features

The previous section has described how DCDs are extracted and used to represent image-regions. Though DCDs are efficient to represent regions, it is impossible to directly use them for semantic learning because there are millions of DCDs. Therefore, a dictionary of DCDs is needed to describe the regions of a database. This section describes how a finite set of representative DCDs can be created using a modified dominant color based VQ algorithm.

The algorithm is presented in Fig. 7 and described as follows.

Step 1: Initialization. The algorithm starts with one cluster

VQ algorithm for dominant color descriptors
Input:
$x_1, x_2, x_3, \ldots, x_M$: M DCDs
Output:
$c_1, c_2, c_3, \ldots, c_N$: N centroid DCDs

1. initially N=1, and merge all DCDs to find initial centroid c_1 [*See description for centroid calculation*]
2.
   i. Select those centroids which have more than 4 colors.
   ii. Let, n = the number of selected centroids.
   iii. If n ==0 go to Step 5 **//Finished**
   iv. Split each of n selected centroids into two. [*See description for splitting process*]
   v. Set N=N +n
   vi. Let all available N centroids named as $c_1^{old}, c_2^{old}, c_3^{old}, \ldots, c_N^{old}$
3. Repeat Steps *i* to *iv*
   i. Let $cluster_i = \phi, \quad \forall_{1 \le i \le N}$
   ii. redistribute each DCD, x_m to its nearest cluster, $cluster_j$:
   set $cluster_j = cluster_j \cup x_m, \quad \forall_{1 \le m \le M}$, so that
   $$j = \arg\min_i(\text{distance}(x_m, c^{old}{}_i))$$ [*See description for distance*(x_m, c_i^{old})]
   iii. recalculate each centroid, $c^{new}{}_i, \quad \forall_{1 \le i \le N}$ [*See description for centroid calculation*]
   iv. if $c_i^{new} = c_i^{old}, \quad \forall_{1 \le i \le N}$ //check for change in centroid
   then
   $c_i = c_i^{new}, \quad \forall_{1 \le i \le N}$ //no change found in any centroid
   Go to Step 4
   else
   $c_i^{old} = c_i^{new}, \quad \forall_{1 \le i \le N}$ //change found in centroids
   Go to Step 3.i
4. repeat from Step 2
5. end

Fig. 7. The Modified VQ algorithm

which includes all the training regions represented with DCDs. The centroid DCD of this cluster is calculated from all member DCDs and thus it is a representative DCD feature vector for all regions belonging to the cluster. The centroid calculation is explained in Fig. 8.

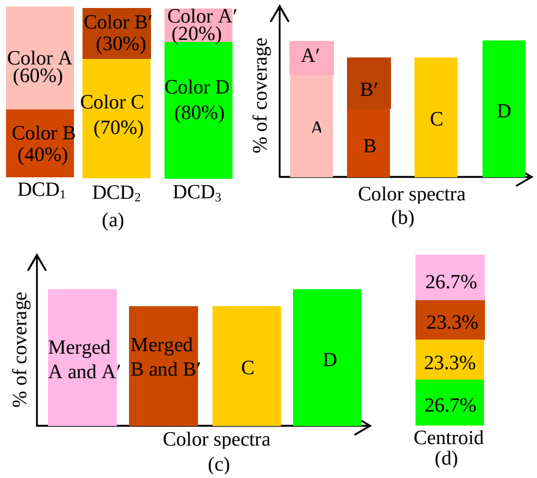

Fig. 8. Illustration of Centroid Calculation. Length of each color block is proportional to its percentage in the corresponding DCD. (a) A cluster with 3 DCDs. Numbers indicate the percentages of colors in corresponding DCDs. The distance between Colors A and A′ is 18.71 and the distance between color B and B′ is19.87. (b) Similar colors are grouped together in the color spectra. (c) Weighted average of similar colors. (d) All four colors of (c) are put together and then normalized to get the final centroid.

Centroid calculation: The principle in centroid calculation is to partition colors from all DCDs into several groups so that within a group the distance between any pair of colors is not greater than T_d. Then a single color is calculated from each group. Suppose there are three DCDs in a cluster, as shown in Fig. 8(a). Each DCD consists of one or more colors. The numerical values of different colors are given in Table I.

TABLE I
LUV VALUES FOR DIFFERENT COLORS

Color	**LUV values**
A	[255 150 25]
A′	[250 165 15]
B	[130 125 50]
B′	[115 112 51]
C	[255 150 75]
D	[255 -100 100]

The distance between colors A and A′ is 18.7083 and the distance between colors B and B′ is 19.8746 in CIE LUV color space. The distances between other pairs of colors are greater than 20. Therefore, the color A & A′ are merged together and so are B & B′, while color C and D make separate groups. Fig. 8(b) shows four groups of colors. For each group, a single representative color is calculated as the weighted average of its member colors. Table II shows the calculation of average colors of two groups. Fig. 8(c) shows the representative colors of all groups. Finally, the top eight colors are selected from the cluster and their percentages are normalized to form the centroid DCD which is shown in Fig. 8(d).

Step 2: Splitting Selected Clusters. In this step the algorithm selects certain types of clusters for splitting. Basically those clusters which consist of a very diverse range of colors should be further split. According to the statistics in Fig. 5, few regions have more than four colors. Therefore if the centroid of a cluster consists of more than four colors, the corresponding cluster is subject to further splitting. This selection criterion avoids the over-fragmentation of clusters because it selects only those clusters which need further splitting. Each selected cluster is split into two using the following splitting process.

Splitting process: The splitting process is explained in Fig. 9. Suppose Fig. 9(a) is the centroid of a cluster which is to be split. Fig. 9(b) shows the feature vector of the DCD in Fig. 9(a). A random perturbation vector is formed with the same dimension of the original vector. This random vector is added to and subtracted from the original vector to find two new centroids, as shown in Fig. 9(d). Next the percentages of colors in new centroids are normalized so that the sum of percentages of colors is 1 in each centroid. Fig. 9(e) shows the feature vectors of two new centroids after normalization and Fig. 9(f) shows their visual representations.

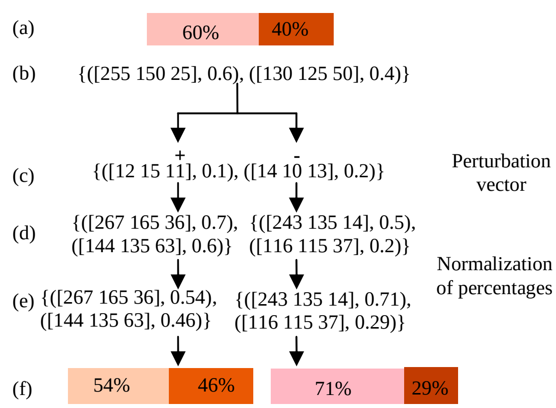

Fig. 9. Splitting Process. (a) A sample centroid. (b) Numerical Representation. (c) A random perturbation vector. (d) Two centroids after adding and subtracting the perturbation vector. (e) Normalized centroids. (f) Visual representation of two new centroids of (e).

Step 3: Updating Existing Clusters. In this step, the algorithm iteratively updates all clusters until all of them become stable. There are two main steps in the iterative

TABLE II
CALCULATION OF AVERAGE COLORS

Average of	**Calculation**	**LUV values**	**Relative amount**
A and A′	([255 150 25]*0.6 +[250 165 15]*0.2) / (0.6+0.2)	=[255 150 0]	60%+20%=80%
B and B′	([130 125 50]*0.4 +[115 112 51]*0.3) / (0.4+0.3)	=[124 119 50]	40%+30%=70%

process in Step 3:

- distributing each DCD to its nearest available cluster,
- calculating the centroid of each cluster

The calculation of centroid has already been described in Step 1. The distribution of a DCD to its closest cluster is determined by the distance calculation between the DCD and each available cluster centroid.

Distance calculation: As a centroid is as an ordinary DCD, so the distance between a DCD and a centroid is calculated in the same way as the distance calculation between two DCDs. Suppose, there are two DCDs, dcd_1 and dcd_2, defined as,

$dcd_1 = \{(c_{1i}, p_{1i})\}$, i=1,2,3,…,N_1, and

$dcd_2 = \{(c_{2i}, p_{2i})\}$, i=1,2,3,…, N_2

The distance, D, between these two DCDs is given by [10],

$$D^2(dcd_1, dcd_2) = \sum_{i=1}^{N_1} p_{1i}^2 + \sum_{j=1}^{N_2} p_{2j}^2 - \sum_{i=1}^{N_1} \sum_{j=1}^{N_2} 2a_{1i,2j} p_{1i} p_{2j}$$

where $a_{k,l}$ is the similarity between color c_k and c_l. It is defined as,

$$a_{k,l} = \begin{cases} 1 - d_{k,l} / d_{\max}, & d_{k,l} \le T_d \\ 0, & d_{k,l} > T_d \end{cases}$$

where, $d_{k,l}$ is the Euclidian distance between the colors c_k and c_l.

After the algorithm terminates, the centroids of all available clusters are the codewords of the codebook (or the dictionary). The codewords are the representative DCDs which will be used to describe all of the database regions.

IV. Performance Test For The Codebook

A. Test Datasets

The image database we use in this research is a set of 5,100 images from the widely used Corel photo gallery. As described in Section II, these images are segmented into 36,692 regions. Two datasets have been created from these 36,692 regions to measure the performance of the algorithm. The first one consists of 871 randomly selected regions. The concepts of these regions include ape, balloon, bear, bird, butterfly, car, copter, deer, elephant, fighter plane, fireworks, flower, fox, horse, plane, tiger, and tree. The selected regions mostly represent the subjects of these categories. In some cases, regions from same category vary in color. We divided all selected regions into a training set and a testing set. The training set consists of 442 regions and the testing set consists of 429 regions.

The second dataset consists of regions from all 51 image-categories and is divided into training and testing set. The regions of 10% randomly selected images from each of 51 categories build the training set. Similarly, the testing set consists of the regions of another 10% randomly selected images from each of 51 categories. Altogether the training and testing set of the second dataset consist of 3556 and 3684 regions, respectively.

For both datasets, training samples are used to generate codebooks, while both training and testing samples are used to measure the performance of their respective codebooks.

B. Performance of Codebook

As explained in Section I, the accuracy of semantic learning depends on the accuracy of codebook generation. Note that each codeword of the codebook is a representative feature. To measure the accuracy of the codebook, we test how accurately the codebook can categorize the testing regions. Each region is represented with the closest codeword from the codebook. Each codeword identifies a group of regions. If a group consists of visually similar regions, it indicates that the corresponding codeword correctly categorizes these regions. And this group is regarded as a homogeneous category. On the other hand, if a group consists of visually different regions, it is regarded as a non-homogeneous category. The percentage of homogeneous categories is an important performance measurement and it is calculated as,

$$\textit{Homogeneous Categories}\ (\%) = \frac{\textit{No of Homogeneous Categories}}{\textit{Total No of Categories}}$$

where the total number of categories is the total number of codewords in the corresponding codebook.

In all homogeneous categories, all regions are considered as correctly categorized. In case of a non-homogeneous category, regions belonging to similar colors are counted. The regions, which belong to the majority class within the non-homogeneous category, are considered representatives for that category. Other regions of that category are considered incorrectly categorized. Thus the total numbers of correctly and incorrectly categorized regions are obtained. The percentage of correctly categorized regions is another important performance measurement and it is calculated as,

$$\textit{Correctly Categorized Regions}\ (\%) = \frac{\textit{No of Correctly Categorized Regions}}{\textit{Total No of Regions}}$$

C. Experimental Result

After applying the modified VQ algorithm of Fig. 7, two training sets generate codebooks of size 99 and 718 codewords, respectively. The first training set consists of regions from selected image concepts, while the second set consists of regions from all available image concepts. Thus the second training set consists of more diverse types of regions, and so, a large number of codewords are needed to properly categorize these regions. Although the number of categories in the first training set is small, it generates 99 codewords after applying VQ. This happens because colors of regions belonging to same concept are not same. Furthermore, regions usually include pixels from background or other regions due to segmentation error. Though a thresholding criterion is applied to discard the unrelated colors, it does not guarantee 100% removal of those colors. However, as long as regions with similar color combination are grouped together, our objective is fulfilled. Table III shows the performance results for both datasets. The high accuracy in categorization shows the ability of the discrete set of codewords to represent regions. Thus, the probability distributions can be accurately calculated if these codewords are used to represent the color information of the regions.

TABLE III
EXPERIMENTAL RESULTS FOR BOTH DATASETS

	First Dataset		**Second Dataset**	
	Training set	**Testing set**	**Training set**	**Testing set**
No. of categories (codewords)	99		718	
% of homogeneous categories	97.98%	95.96%	99.72%	97.91%
% of correctly categorized regions	99.32%	97.67%	99.94%	99.07%

In this paper, the Corel dataset is used to generate the initial result about the performance of the modified VQ algorithm. For a general image database, a codebook from much larger dataset is needed to represent the regions, because a large codebook can represent a wider range of visually different regions. In fact, in order to learn region-semantics using decision tree or decision rules, regions have to be represented with features from multiple dictionaries, e.g., color, texture and shape dictionaries. By combining multiple dictionaries, accurate region-semantics can be learnt.

D. Comparison with Template based Approach

The performance of our VQ based dictionary is compared with the performance of the template based dictionary proposed in Liu *et. al*. [5]. The same methodology is applied to measure the performance of the template based dictionary. The first dataset is used for this purpose and the result from the test set of this data set is presented. Table IV shows the performance results.

TABLE IV
PERFORMANCE COMPARISON BETWEEN THE VQ BASED AND TEMPLATE BASED DICTIONARIES

	VQ based Dictionary	**Template based Dictionary**
% of homogeneous categories	97.98%	52.17%
% of correctly categorized regions	99.32%	59.50%

As explained in Section I, the template based approach uses a single dominant color to represent a region and defines a template as the average of colors from a few selected regions. Therefore, many regions lose important color information in this approach. As a result, visually different regions are represented with the same representative feature vector and the quality of the dictionary becomes poor which is reflected in Table IV. E.g., Fig. 10 shows few example regions which

Fig. 10. Regions which are described by the same representative feature from the template based dictionary.

are described by the same feature vector of the template based dictionary. This happens because the template based approach loses the blackish colors from the 'tiger' regions. Therefore, it can not differentiate these regions from the yellowish 'bear' regions and uses the same template to represent all of them.

V. CONCLUSIONS

This paper presents a region based vector quantization method using color features. The main objective is to find a finite set of representative dominant color descriptors to represent the regions of a real image database. The main contribution of this work is the extension of the conventional VQ to adapt to variable dimensional features and the proposal of a criterion to overcome the over-fragmentation issue.

Experimental results show that our VQ based dictionary classifies real image-regions with high accuracy. With this high performance VQ based dictionary, we will be able to learn higher level semantics for a large image database. In future, texture and shape dictionaries will be created. And machine learning algorithm will be applied to learn region-semantics using these three dictionaries.

REFERENCES

[1] A. Vailaya, M. A. T. Figueiredo, A. K. Jain, and H.-J. Zhang, "Image Classification for Content-Based Indexing," *IEEE Transactions on Image Processing*, vol. 10, No. 1, pp. 117-130, January 2001.

[2] M. Szummer and R.W. Picard, "Indoor-Outdoor Image Classification," *IEEE Intl Workshop on Content-based Access of Image and Video Databases*, Jan 1998.

[3] Y. Liu, D. Zhang, G. Lu, and W.-Y. Ma,"Deriving High-Level Concepts Using Fuzzy-ID3 Decision Tree for Image Retrieval," in the *Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol 2, pp. 501 – 504, March 18-23, 2005.

[4] S. D. MacArthur, C. E. Brodley, and C. Shyu, "Relevance Feedback Decision Trees in Content-Based Image Retrieval," in the *Proceedings of the IEEE Workshop on Content-Based Access of Image and Video Libraries*, Hilton Head, SC, June 2000.

[5] Y. Liu, D. Zhang, and G. Lu, "Region-Based Image Retrieval with High-Level Semantics Using Decision Tree Learning," *Pattern Recognition,* 41(8):2554-2570, 2008.

[6] F. Jing, M. Li, H. J. Zhang, and B. Zhang. "An Efficient and Effective Region-Based Image Retrieval Framework," *IEEE Transactions on Image Processing*, Vol. 13, No. 5, May 2004.

[7] Y. Deng, and B. S. Manjunath, "Unsupervised Segmentation of Color-Texture Regions in Images and Video," *IEEE Transaction on Pattern Analysis and Machine Learning* (2001), 23(8) 800-810.

[8] B. S. Manjunath , P. Salembier, and T. Sikora, *Introduction to MPEG-7: Multimedia Content Description Interface*, John Wiley & Sons, Inc., New York, NY, 2002.

[9] A. Gersho and R. M Gray, *Vector Quantization and Signal Compression*, Kluwer Academic Publishers, Norwell, Mass., 1993.

[10] N.-C. Yang, W.-H. Chang, C.–M. Kuo, and T.-H. Li, "A Fast MPEG-7 Dominant Color Extraction with New Similarity Measure for Image Retrieval" Journal of Visual Communication & Image Retrieval, vol. 19, pp. 92–105, 2008.