Chapter 6

Scheme I

This scheme presents an efficient spatial indexing technique based on Silhouette moments that makes the index robust subject to the three basic transformations for CBIR. Spatial index is generated based upon a fast and robust clustering technique, which can recognize color clusters of any shape. The new clustering technique has been found to be efficient in terms of time complexity and cluster quality than many of its counterparts. A matching engine has been devised to retrieve images from the image database, which has the capacity for global and regional similarity search.

6.1 Architecture of Scheme I

Scheme II works in four steps. In Step 1, it accepts the input query image, quantizes the image and produces the color clusters present in the image. In the 2nd step the user identifies the color clusters and the combination of color clusters that forms the objects in an interactive manner. In Step 3, translation, aspect ratio, scale and rotation invariant silhouette moments are calculated to generate the spatial cluster indices and the spatial object indices of the selected clusters and objects. In Step 4, query results are given based on inferences made by a matching engine.
6.2 Entropy Calculation

As discussed in Chapter 3 Section 3.3, entropy is calculated for the given input image and this entropy value is used for quantizing an image. An average entropy for a large collection of images (around 10,000 images downloaded from the web) has been calculated and plotted in Figure 6.2. From the entropy plot it has been found that the average entropy of images is mostly 2. Hence after exhaustive experimentation we can infer that those images whose entropy value is less than or equal to 2 contains less colors otherwise it contains more colors and needs quantization.
6.3 Quantization

Quantization of the color space is necessary to reduce the dimensionality of the index that characterizes an image at the cost of the quality of the index. The proposed scheme quantizes the color contents of the query image over HSV color space. It basically attempts to quantize the *Hue* component of each color pixel by balancing the visual fidelity and the dimensionality of the resulting quantization. The Human Visual System (HVS) discerns the changes in the *hue* component by much smaller gaps than changes in the *Saturation* and *Value*. Based on experimentation it has been observed that partitioning the buckets with an equidistant interval of 5 is more justified. Hence the quantization parameter $q=5$ is kept the same for all images. Thus the total number of buckets over the *hue*-axis is 20. The images that have entropy value greater than 2 are quantized with this partition.

After quantization of those images that needs it and those images that does not need quantization have to go through the process of cluster generation and subsequently spatial object index generation module. In this scheme we have used the power of BOO-Clustering algorithm for cluster generation.

6.4 Clustering

To retrieve the color clusters present in the input image both the BOO-Clustering algorithm as discussed in Chapter 5 and GDBSCAN [JSM00] (Generalized Density Based Spatial Clustering of Applications with Noise) has been employed. The only input parameter to BOO-Clustering algorithm is a distance parameter $d$ that defines the neighborhood distance of the template. Where as, the GDBSCAN algorithm takes two input parameters, neighborhood distance $d$ parameter and min-points, the minimum number of points to be present in the neighborhood defined by $d$ parameter. The clustering results of both the algorithms depend on the choice of the input parameters.

The value of the $d$ parameter was kept fixed for all images. But as the distribution of colored pixels vary from image to image, a fixed $d$ value may not produce distinct and meaningful color clusters for different images by BOO-Clustering algorithm. Hence to get distinct and meaningful color clusters both the BOO-Clustering and GDBSCAN algorithms are executed in a trial and error method for different input parameters.


CHAPTER 6. SCHEME I

6.5 Detection of color clusters and objects of interest in an image

The output of the BOO-Clustering algorithm and GDBSCAN algorithm is a set of color clusters of an image. An object is formed by a single color cluster or by combination of a set of color clusters. In an image the color clusters or some or all combination of color clusters that forms object within the image may not be of importance for index generation purpose. An interactive method is the only effective way for detection of proper color clusters and objects of interest of an image for index generation. Hence in this scheme, those clusters and objects of an image are chosen which are of importance for index generation for the image manually.

For example Figure 1.1 depicts an image having four different color clusters denoted by color$^1$ the leaf color cluster of the tree, color$^3$ the trunk color cluster of the tree, color$^1$ the background color cluster, and color$^4$ the soil color cluster. Each of these color clusters can be represented as objects. The other distinct object of the image is combination of the color$^1$ the leaf color cluster of the tree and color$^3$ the trunk color cluster of the tree. Thus there are five distinct objects in the image which are of importance for index generation for the image. The other combination of color clusters such as color$^3$ the background color cluster and color$^4$ the soil color cluster are not of very much importance for index generation. Hence to select the objects of interests in an image for index generation only an interactive method is an appropriate one at the cost of time complexity.

6.6 The Spatial Cluster Index and Object Index

As discussed in Chapter 3 the translation, aspect ratio, scale and rotation invariant silhouette moments [RMK98] of second order are given by

\[
\varphi_1 = \eta_{20} + \eta_{02} \quad (6-1)
\]

\[
\varphi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad (6-2)
\]

\[
\varphi_3 = \eta_{20}\eta_{02} - \eta_{11}^2 \quad (6-3)
\]

In this scheme equations (6-1), (6-2), and (6-3) are utilized for index generation of the color clusters and the objects generated. The indices for the color cluster or objects of interest consists of four parameters viz., $<\text{color of the cluster}, \varphi_1, \varphi_2, \varphi_3>$ termed as spatial cluster index and $<\text{average color of the clusters}, \varphi_1, \varphi_2, \varphi_3>$ is termed as spatial object index respectively. Hence, a search on spatial cluster index of all the clusters present in the input image represents a global similarity search of the image. A search on spatial object index, or one or a few spatial cluster indices of the input image represents regional similarity search of the image. The spatial cluster

94
indices and spatial object indices are stored in a spatial Cluster Object Tree as depicted in Figure 6.2. Here the \( \phi_1, \phi_2, \phi_3 \) are the translation, aspect ratio, scale and rotation invariant silhouette moments. For two or more similar clusters/objects with different colors will have the same \( \phi_1, \phi_2, \phi_3 \) values.

![Figure 6.2. Spatial cluster-object tree](image)

**6.7 Robustness & Compactness of the Index**

To establish an index to be robust, it has to be robust subject to the three basic transformations, i.e. translation, rotation, uniform and non-uniform scaling. The clusters produced by BOO-Clustering are more reasonable than GDBSCAN \([JSM00]\) due to the following reasons:

1. The time complexity of BOO-Clustering is \( \Theta(N) \) as compared to \( \Theta(N \times \log N) \) of GDBSCAN when applied to image data.

2. Both GDBSCAN and BOO-Clustering algorithms can identify color clusters of arbitrary shapes present in the image viz. concave and convex. But the drawback of GDBSCAN is that it depends on two parameters viz. \( d \) and min-points; where as BOO-Clustering depends only on one parameter viz. \( d \). Hence keeping \( d \) parameter to a fixed value we shall always get a fixed set of color clusters for a particular image in case of BOO-Clustering algorithm. But in case of GDBSCAN, keeping \( d \) parameter fixed but by altering the second parameter min-point we shall get different set of color clusters.
CHAPTER 6 SCHEME I

The three parameters viz., \( \varphi_1, \varphi_2, \varphi_3 \) are already been proved to be robust to the three basic transformations [RMK98]. The other parameter average color of the cluster or the object is also robust to the three basic transformations because the average color of the cluster or object of an image always remains the same even the image is tempered with the three basic transformations. Moreover if the color of the image is tempered, the other three components of the index will not change. Thus the four dimensional index is robust in all respect.

As regards compactness of the index, both BOO-Clustering and GDBSCAN algorithm produces color clusters present in the image eliminating the noisy clusters. From these color clusters objects of interest are extracted. For each color clusters and object of interest a four-tuple index is calculated. Hence the size of the index varies from image to image depending on the color clusters and objects of interest present in the image. But index generated by the George et al. [GPI03] method is more compact than ours because their method generates only one set of twelve chromatic moments for each image and which is fixed for any image. Keeping in view of the retrieval efficiency, ours is better than our counterparts. This is because our method can perform global search as well as region or object level search, which is a lacuna of George et al. method that it cannot search at object level.

6.8 Database Organization

The output of the BOO-Clustering algorithm and GDBSCAN is a set of color clusters from which the objects of interest are separated out from the image in an interactive manner. The spatial cluster indices and object indices are stored in a spatial cluster-object tree. The spatial cluster object tree is a variant of B-tree (Figure 6.2). The root node termed as Cluster_Object Root Node, of the tree points to \( k \) independent tree structures, where \( k \) is the number of parameters in the index. If a new parameter is to be accommodated (i.e., for a \( k+1 \) dimensional index), the root node has to be updated by insertion of a new pointer and accordingly an associated tree structure will have to be generated. Each of the parameter trees will maintain the parameter key value (i.e., say, Cluster/Object Color value, \( \varphi_1 \) value, \( \varphi_2 \) value, \( \varphi_3 \) value), along with a pointer to a list of Image IDs (i.e., PIDs).

6.9 Matching Engine for Scheme I

The matching engine has the facility for searching images in the image database based on the spatial cluster and object indices using the cluster-object tree (Figure 6.2). Figure 1.1 depicts a query image with four clusters having four different colors; color\(^1\), color\(^2\), color\(^3\), and color\(^4\). For
each cluster, the query image will have one cluster index. Let these cluster indices be denoted by 
\(<\text{color}^1, \varphi_1^1, \varphi_2^1, \varphi_3^1>, <\text{color}^2, \varphi_1^2, \varphi_2^2, \varphi_3^2>, <\text{color}^3, \varphi_1^3, \varphi_2^3, \varphi_3^3>\) and 
\(<\text{color}^4, \varphi_1^4, \varphi_2^4, \varphi_3^4>\).

For global search, the matching engine searches for all those images that have all the four cluster 
indices, as that of the query image, present in image database.

For object level search, at first the objects present in the query image are determined from the set 
of clusters present in the query image. The objects may consist of either a single color cluster or a combination of color clusters. For example, in Figure 1.1 the query image has four color clusters 
(1, 2, 3 and 4). Cluster 1 and 2 forms the object "tree"; color cluster 3 is the "background" object; color cluster 4 is the "soil" object. Also color cluster 1 is the "leaf" object of the tree and color 
cluster 2 is the "trunk" object of the tree. Thus there are all total five objects in the query image as 
depicted in Figure 1.1 and hence there will be five object indices in the query image. Let these 
object indices be denoted by 
\(<\text{Leaf}_\text{object}^1, \varphi_1^1, \varphi_2^1, \varphi_3^1>, <\text{Trunk}_\text{object}^2, \varphi_1^2, \varphi_2^2, \varphi_3^2>, <\text{Background}_\text{object}^3, \varphi_1^3, \varphi_2^3, \varphi_3^3>, <\text{Soil}_\text{object}^4, \varphi_1^4, \varphi_2^4, \varphi_3^4>\) and 
\(<\text{Tree}_\text{object}^5, \varphi_1^5, \varphi_2^5, \varphi_3^5>\). Here Tree_\text{object}^5 value is the average color value of Leaf_\text{object}^1 and Trunk_\text{object}^2.

For object level search, one of the query may be "Find all the images from the image database 
that have the object 'tree'". For this query the matching engine will use the index 
\(<\text{Tree}_\text{object}^5, \varphi_1^5, \varphi_2^5, \varphi_3^5>\). At first it will search in the color tree and will fetch those Picture Ids from the 
database that matches the stored color value with the Tree_\text{object}^5 color value. Next it will take 
up \(\varphi_1^5\) value and search will be done on the \(\varphi_1\) tree and will fetch another Picture Id list for matching stored \(\varphi_1\) value with \(\varphi_1^5\) value. Similarly the matching engine will fetch another two 
Picture Id lists for \(\varphi_2^5\) and \(\varphi_3^5\) value by searching in the \(\varphi_2\) and \(\varphi_3\) tree respectively. Next all the 
four Picture Id lists are concatenated to form a single Picture Id list, which is the final list of 
image retrieved.

The matching engine also has the facility for searching in the image database for color clusters 
and objects using both the conjunction 'AND' and disjunction 'OR'. For example, the query 
image as depicted in Figure 1.1 may have a query that "Find all images that have color cluster (1 
AND 2) OR 3; i.e. find all those images which have either both the (Leaf Objects 1 AND Trunk 
Object 2) OR Background Object 3". For this query, the indices 
\(<\text{Leaf}_\text{object}^1, \varphi_1^1, \varphi_2^1, \varphi_3^1>, <\text{Trunk}_\text{object}^2, \varphi_1^2, \varphi_2^2, \varphi_3^2>\) and 
\(<\text{Background}_\text{object}^3, \varphi_1^3, \varphi_2^3, \varphi_3^3>\) of the query image will 
be utilized and the search will initiate from Leaf object index. At first, it will search in the color
tree and will fetch those Picture Ids from the database that matches Leaf_object color value. Next it will take up \( \phi_1 \) value and search will be done in the \( \phi_1 \) tree and will fetch another Picture Id list that matches \( \phi_1 \) value. Similarly, the matching engine will fetch another two Picture Id lists for \( \phi_2 \) and \( \phi_3 \) value by searching in the \( \phi_2 \) and \( \phi_3 \) tree respectively. After that all the four Picture Id lists will be concatenated to form a single Picture Id List (PID) which matches the first Leaf Object index \(<\text{Leaf}_\text{Object}, \phi_1, \phi_2, \phi_3>\) of the query. Thus another two Picture Id lists (PID\(^2\) and PID\(^3\)) for the indices \(<\text{Trunk}_\text{Object}, \phi_1^2, \phi_2^2, \phi_3^2>\) and \(<\text{Background}_\text{Object}, \phi_1^3, \phi_2^3, \phi_3^3>\) will be generated. Now, according to the query the Picture Id lists PID\(^1\), PID\(^2\) and PID\(^3\) will be combined together according to (PID\(^1\) AND PID\(^2\)) OR PID\(^3\) to produce the final set of Picture Id list that corresponds to the images in the image database which satisfy the given query. This type of query can also be applied to object level search and combination of object and cluster level search.

The matching engine also has the facility of searching images in the database based on the parameters of interest and also on objects/clusters of interest. Both the conjunction AND and disjunction OR can be used both on objects and parameters. For example a query can be “Search for all images having any (color value) FOR (\( \phi_1 \) value AND \( \phi_2 \) value AND same \( \phi_3 \) value) for objects (Leaf Object AND Trunk Object) OR Background Object. i.e. Search for images that contains the Tree Object and Background Object of any color”. For this query, the indices \(<\text{Leaf}_\text{Object}, \phi_1, \phi_2, \phi_3>, <\text{Trunk}_\text{Object}, \phi_1^2, \phi_2^2, \phi_3^2>, <\text{Background}_\text{Object}, \phi_1^3, \phi_2^3, \phi_3^3>\) of the query image will be utilized and the search will initiate from Leaf object index. The matching engine will start the search operation from Leaf object. First it will fetch a list of images by searching in the \( \phi_1 \) tree. Similarly it will fetch two lists of images for \( \phi_2 \) and \( \phi_3 \) and finally these three lists of images will be merged to a single list (PID\(^1\)) according to the given criteria \( \phi_1 \) AND \( \phi_2 \) AND \( \phi_3 \). This list (PID\(^1\)) will contain all images that have the Tree object of any color and having similar shape. Similarly, another two lists (PID\(^2\)) and (PID\(^3\)) will be generated by the matching engine for the Trunk object and the Background object. Finally these three lists will be concatenated according to the given criteria; Leaf object AND Trunk object OR Background object; to form a single list. This final list of images is the desired set of images of the said query. The Matching Algorithm has already been discussed in Chapter 5.
CHAPTER 6. SCHEME I

6.10 Complexity Analysis and Comparison

There are two phases of computations involved in querying an image database. First, calculation of the index for the query image and second, comparison of the generated index with the stored indices of the images in the image database and subsequently retrieval of the similar images from the image database. We have compared this Scheme with the Paschos et. al. Chromatic Moment [GPI03] based technique.

6.10.1 Retrieval Time by Chromatic Moments:

1. Computation of Chromatic Moments is \( O(M \times Q^2 \times (2 \times \mu \times \alpha)) \)

   where \( M = \) No. of Chromatic Moments = 12,
   \( Q = \) No of histogram bins,
   \( \mu, \alpha = 2 \) multiplication and one addition per histogram bin.

2. Histogram computation time is \( O(N) \)

   where \( N = \) Total numbers of pixel in the image

3. Time taken by reading the chromaticity moments of each stored image is \( O(S \times M \times \rho) \)

   Where \( S = \) Total number of images in the image base,
   \( \rho = \) Time for reading the number from the disk

4. Computing distance to each stored image is \( O(S \times M \times (\alpha + d)) \)

   Where \( d = \) is the time taken to compute an absolute value

Hence the total retrieval time is \( O((1) + (2) + (3) + (4)) \)

   Which is \( O(M \times Q^2 \times (2 \times \mu \times \alpha)) + O(N) + O(S \times M \times \rho) + O(S \times M \times (\alpha + d)) \)

   Which is \( O(M \times Q^2 + N + S \times M + S \times M) \)

   Which is \( (M \times Q^2 + 2S + N) \)

6.10.2 Retrieval Time by BOO-Clustering and GDBSCAN techniques:

1. Time taken for Cluster generation by BOO-Clustering technique is \( O(N) \)

2. Time taken for Cluster generation by GDBSCAN technique is \( O(N \times \log N) \)

3. Time taken for Index calculation for an object having \( C \), no of pixels =

   Time taken for calculation of central moment for \( \mu_{20}, \mu_{20}, \mu_{11}, \mu_{11} \) + Time taken for calculation of aspect ratio invariant moment for \( \eta_{20}, \eta_{20}, \eta_{11} \) + Time taken for calculation of rotation, translation, scale and aspect ratio invariant moments \( \varphi_1, \varphi_2, \varphi_3 \)
CHAPTER 6. SCHEME I

is $O(4 \times C_i + \psi + v)$

4. Time taken for calculation for $O$ number of objects

is $O(4 \times \left( \sum_{i=1}^{O} C_i \right) + \psi + v)$

which is $O(4 \times \psi + v)$ is $O(4N)$

5. Retrieval time is $O \times \log S$ is $\log S$ [O is ignored because of its smaller size]

6. Total retrieval time taken by BOO-Clustering technique = (1) + (4) + (5)

is $N + 4N + \log S$

is $5N + \log S$

7. Total retrieval time taken by GDBSCAN technique = (2) + (4) + (5)

is $O(N \times \log N + 4N + \log S)$ is $O(N \log N + \log S)$

Here, $Q_2^2 < 5N$ and $Q_2^2 < N \times \log N + 4N$ but $2S \gg \log S$. If the number of images in the image database is smaller in size the Chromatic Moments index has time advantage over the BOO-Clustering and GDBSCAN techniques at the cost of precision and recall. As the number of images in the image database increases in size, both the BOO-Clustering and GDBSCAN techniques have time advantage over chromatic moments with a good precision and recall.

6.11 Experimental Results

To test the technique, we used a downloaded database ((a) Cohn-Kanade Facial Expression Database; http://www.pitt.edu/~jeffcohen, (b) ftp://ftp.eecs.umich.edu.groups/ai/dberwick/essbtml.zip and other images) consisting of 5000 real world and synthetic images divided into 100 similar groups such as facial expressions, scenery, animals, cars, flowers and space. Implementation was carried out for chromatic moment based technique along with the proposed cluster and object based techniques using BOO-Clustering method and GDBSCAN method in HSB color space. All the methods have been implemented in Java platform in an Intel Pentium IV machine. For any input image, indices are generated for the color clusters and objects of interest in an interactive manner for both the BOO-Clustering and GDBSCAN method. Similarity search is done on clusters of interest or objects of interest using a matching engine. We have performed a two way test to establish the supremacy of the BOO-Clustering method over the Chromatic Moment and GDBSCAN methods. They are:

1. Retrieval time comparison of BOO-Clustering method vs GDBSCAN method and Chromatic Moment method.
CHAPTER 6. SCHEME I

2. Retrieval effectiveness of the three methods for global and regional search.

6.11.1 Retrieval time comparison of BOO-Clustering vs GDBSCAN and Chromatic Moment methods

In this experiment we have tried to plot the average time taken by the three methods vs the number of images in the image database for global and regional search. In global search, all the clusters generated by the GDBSCAN and BOO-Clustering method are taken into account for the search operation; and in regional search, some significant clusters or objects produced by GDBSCAN and BOO-Clustering method are taken into account for the search operation. To test the retrieval time comparison, we have taken 100 images from the image database and the average retrieval time for the 100 images for both the global and regional search has been plotted for BOO-Clustering, GDBSCAN, Chromatic Moment method for different database size (Figure 7.3 and 7.4). It can be clearly seen that as the size of the image database increases the BOO-Clustering method has time advantage over the GDBSCAN and Chromatic Moment method.

6.11.2 Retrieval effectiveness of the three methods for Global and Regional Search

Retrieval effectiveness is calculated by the average precision recall (APR) curve. In our experiment, the APR was plotted for the best 200 queries out of the calculated 500 queries taken at random over the downloaded image database. Numerically, the area [RKS99] enclosed by an APR curve and the axes as a performance metric, called performance area, is defined as

$$\frac{1}{2} \sum_{i=1}^{N-1} (x_{i+1} - x_i)(y_{i+1} + y_i)$$

(5-5)

where \((x_i, y_i)\) is the (recall, precision) pair when the number of retrieved images is \(i\) and \(N\) is the total number of top matches.

The proposed method was compared with the Chromatic Moment based technique [RKS99] along with the GDBSCAN [JSM00] method for both global and regional search. In global search, all the clusters produced by BOO-Clustering and GDBSCAN for a query image are taken into account for the search operation for both the techniques. Figure 6.5 reflects the query result of global search of the three techniques. Based on experimental results, it is found that the performance areas for the curves are: 387 (BOO-Clustering based retrieval), 321 (GDBSCAN based retrieval) and 271 (Chromatic Moment based retrieval). Hence the improvement produced by the proposed BOO-Clustering based method over the GDBSCAN and Chromatic Moment based technique are: 20.56% and 42.8% respectively.
CHAPTER 6. SCHEME I

The proposed technique was also tested based on some specific clusters and objects (formed by combination of some selected clusters) of the query image along with its counterparts. For fine-tuning the search operation we have used AND/OR conjunction between the clusters and objects of some of the query images, i.e. if for a query image, five significant clusters (C1, C2, C3, C4, C5) are chosen for the search operation, then the search can be given as: “Search for all the images in the image database where cluster C1 AND C2 AND C3 OR C4 AND C5 matches with the query image”. Figure 7.6 reflects the query result of regional search of the two techniques. Based on experimental results, it is found that the performance areas for the curves are: 309 (BOO-Clustering based retrieval) and 258 (GDBSCAN based retrieval). Hence the improvement produced by the proposed BOO-Clustering based method over the GDBSCAN based technique for regional search is 19.77%.

6.12 Discussion

An improved content-based indexing scheme has been presented in this scheme. The scheme generates a compact, 4-dimensional transformation invariant index for the color clusters and objects of interest produced by a robust data clustering technique of any color image. The indices produced help in global and regional similarity search of images. The indices generated are stored in a spatial tree structure for faster retrieval of images. The proposed scheme can be found to be superior in comparison to its counterparts [GPI03]. The main drawback of the scheme is that, the clusters generated by BOO-Clustering and GDBSCAN algorithms are to be selected in an interactive manner for object detection and subsequently index generation for all images while creating the spatial cluster and object index database. Hence, the scheme suffers from execution time point of view while creating the spatial cluster index and object index database. To overcome it, next chapter presents an enhanced version of the scheme.
CHAPTER 6. SCHEME I

Retrieval Time Comparison for Regional Search

![Graph showing retrieval time comparison for Regional Search]

Figure 6.3. Retrieval time comparison for Regional Search

Retrieval Time Comparison for Global Search

![Graph showing retrieval time comparison for Global Search]

Figure 6.4. Retrieval time comparison for Global Search.
Figure 6.5. Precision Recall for BOO-Clustering vs GDBSCAN and Chromatic Moment based techniques for Global search.

Figure 6.6. Precision Recall for BOO-Clustering vs GDBSCAN based techniques for Regional search.
Figure 6.7. Some retrieval results of Scheme I.