Turk. J. Math. Comput. Sci.
9(2018) 103–116
© MatDer
http://dergipark.gov.tr/tjmcs
http://tjmcs.matder.org.tr

MATDER

A Comparison Study on Image Content Based Retrieval Systems

Saed Alqaraleh[a,*], Hersh Hama[a]

[a]Department of Computer Engineering, Hasan Kalyoncu University, Turkey.

Received: 19-02-2018 • Accepted: 21-10-2018

Abstract. In recent years, multimedia searching has become an important research field. Multimedia files are one of the most important materials on the internet. Unfortunately, even for the state-of-the-art methods and applications based on accessing multimedia on the internet, it is hard to find the required files. The main purpose of this study is to investigate the performance of well-known image content-based retrieval techniques, i.e., Fuzzy Color and Texture Histogram (FCTH), Edge Histogram Descriptor (EHD), Scalable Color Descriptor (SCD), Color Layout Descriptor (CLD), Color and Edge Directivity Descriptor (CEDD), and Speed-Up Robust Feature (SURF) combined with Fast Library Approximate Nearest Neighbor (FLANN). In general, the objective of using these techniques is to find the query's most relevant files and list them at the top of the retrieval list.
Several experiments have been conducted and it has been observed that FCTH and SCD outperform other studied techniques. On the other hand, for the SURF combined with FLANN approach, the results of most of the queries were below user expectations. In addition, extracting the feature vectors using this method requires massive amount of memory. Overall, none of the studied CBIR descriptors can be used individually to build a full image retrieval system. In our opinion, multiple descriptors can be used simultaneously to achieve a more robust system and accurate results.

2010 AMS Classification: 68P20, 68U10,68U35, 68M11.

Keywords: Content-based retrieval systems, image retrieval systems, multimedia searching, information retrieval.

1. Introduction

In past decades, videos and images have been developed with the help of social media, and the growth of digital devices. This development played a significant role in the ever-increasing use of multimedia data. Indeed, electronic communication has advanced rapidly in recent years, mostly due to the evolution of information technology. As a result, some multimedia search engines have been developed. In general, multimedia search engines can be categorized into: Firstly, Text Based Image Retrieval (TBIR), where users are required to write a query keyword (s) as a text to obtain images from database systems. Then, a search engine returns candidate relevant images in a ranked list, where the score of ranking is done according to a similar measurement between the search keyword and textual features of database images [5]. TBIR has been used widely in multimedia search engines and it is easily implemented. However, the main challenge for retrieving images using TBIR is that multimedia databases are built using the images surrounding text, such as filenames, metadata, link tags and content of the web pages that contain the indexed image. In some cases, multimedia surrounding text has no relation to the contents of the files itself. This might lead to obtaining irrelevant,

*Corresponding Author
Email addresses: saed.alqaraleh@hku.edu.tr (S. Alqaraleh), hersh.hama@uor.edu.krd (H. Hama)

duplicate, and undesired multimedia files. Secondly, content-based search engines (CBIR), which are used to find multimedia files on the Internet based on the files visual contents. CBIR employs visual content of images, such as color, texture, and shape [2], to retrieve the relevant images from the web and the databases system. Query-by-example (QBE) is a query technique that allows the user to search for documents based on an example. This technique can be used when a user has an image and is looking for similar files.

Mainly, multimedia features are categorized as: color, texture, and shape features and their details are summarized as: a) color feature [10, 19], which is widely used in image retrieval because of its simplicity and the fact that it does not rely on an image's orientation or size. However, the intensity of light and camera viewpoint plays an important role in the relevance of query results, b) texture feature [14], refers to visual patterns and plays a significant role in people's interpretations and visual perceptions and interpretation, by intuitively providing measurement of properties such as coarseness, regularity and smoothness, and c) shape feature, tries to refer to particular regions in images that contains all the geometrical visual information that could be sought out [1,9,26]. Figure 1. shows the main types of multimedia features and the categories of the studied methods, i.e. FCTH, EHD, SCD, CLD, CEDD, and SURF.

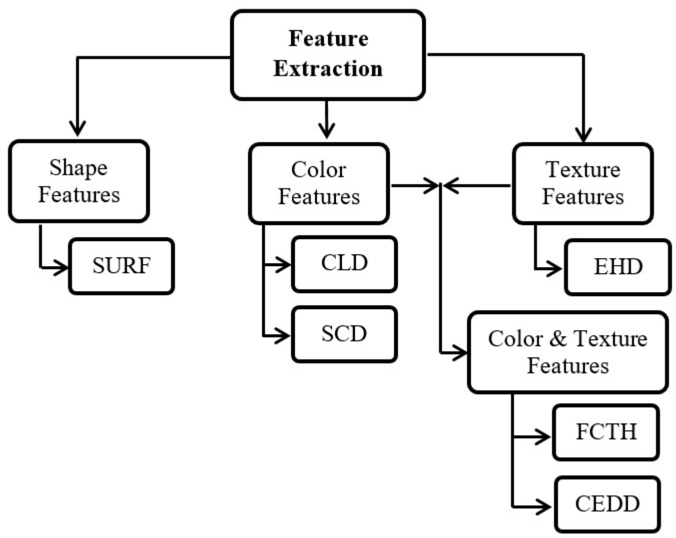


FIGURE 1. Main types of multimedia features

2. Related Work

Nowadays, multimedia retrieval systems are one of the important research fields. Many researchers are trying to improve the efficiency of getting multimedia files through the Internet. In this section, we have summarized recent developments related to image retrieval. In [7, 20, 22], query specific semantic signature, which is a unique image representative produced from the image features was introduced. The main advantages of semantic signature are: 1) the signatures are shorter than the image visual features, and 2) all duplicated images have the same signatures. Overall, this approach is efficient because the semantic signature is shorter than the original image feature vector, however the size of the retrieval image pool in this technique predefined to a fixed size (e.g., including 1000 images). In [17], an iterative approach was employed to refine the effectiveness of query ranked lists. This method assumes that contextual information provided by the ranked lists can be used for improving the incorrect scores, i.e., order of query results. In [16], the proposed approach utilizing query image to generate relevant description; such is acquired by computing the global similarity of a query image to a huge web collection of captioned images. Large scale database of pictures that are associated with descriptive text is one key requirement of this approach. In [24], the approach consists of online and offline stages. The offline part is done by learning the re-ranking model [24], using user label training data, which is constructed from the text-based search results. In the learning stage, a score vector is calculated for each image and it is a corresponding query. In the online stage, user should be submitting a text query into a search box engine. Then, the user will be able to obtain the initial ranked results. In [25], the multi-view hyper-graph-based learning technique

(MHL) adaptively associated click data and diverse visual features have been developed. There are three main steps in this approach. First step, achieving the query independent semantic representation, this process is done by classifying images as relevant and irrelevant to the query. This approach assumes that the queries have a strongly relevant relation with high click counts images and it assumes that the semantic similarities among these images are high. Second step, hyper-graph learning [25], is used to construct a group of manifolds for different visual features, where a set of vertices is connected by hyper-edge in a hyper-graph [25]. Third, the semantic of the manifold click data associated with multiple visual manifolds is calculated using the graph-based learning framework.

As mentioned before, even for the state-of-the-art methods and applications based on accessing multimedia on the internet, it is hard to find the required images. The main purpose of this study is to investigate the performance of well-known image content-based retrieval techniques and to find out if any of the studied content-based retrieval techniques can be used individually to build a full image retrieval system. In addition, the studied CBIR techniques are Fuzzy Color and Texture Histogram (FCTH) [4], Edge Histogram Descriptor (EHD) [23], Scalable Color Descriptor (SCD) [13], Color Layout Descriptor (CLD) [8], Color and Edge Directivity Descriptor (CEDD) [11], and Speed-Up Robust Feature (SURF) combined with Fast Library Approximate Nearest Neighbor (FLANN) [3].

3. Content Approaches

3.1. **Fuzzy Color and Texture Histogram (FCTH).** The FCTH descriptor extracts and combines color and texture attributes in one histogram. The size of FCTH is limited to 72 bytes per image, hence, it reduces the high dimensional vector of images and makes it more suitable to be used for large image databases. In addition, this method is appropriate for accurately retrieving images that have noise or distortion and can handle and be used for rotating images [4]. In FCTH, the image is divided into a number of blocks. Those image-blocks are then handled independently to extract their color information, through a 2 staged fuzzy histogram linking procedure that produces a 24-color histogram. In addition to this 24-bins histogram, and for the purpose of incorporating additional texture information, FCTH uses the high frequency bands of the Haar Wavelet Transform in a fuzzy system, to form 8 texture areas.

3.2. **Edge Histogram Descriptor (EHD).** Edges are a significant feature representing the content of the images [23]. In this descriptor, a histogram is used to represent edge features. The size of the histogram is fixed, i.e., only 80 bins. This makes it suitable to be used for huge databases and it is also more effective for retrieving images of different sizes and/or rotated ones. Initially, regardless of the original size of the image, it will be classified into 4×4 equal size local areas that are called sub-images. Next, for each sub-image, a histogram of edge distribution is generated. In this descriptor, the edges are categorized as follows: vertical edge, horizontal edge, 45-degree edge, 135-degree edge, and non-directional edge [23].

3.3. **Scalable Color Descriptor (SCD).** The Scalable Color Descriptor [13], is derived from a color histogram defined in the Hue-Saturation-Value (HSV) color space with fixed color space quantization, i.e., 256 bins. In addition, it uses a Haar transform coefficient encoding, allowing scalable representation of description, as well as complexity scalability of feature extraction and matching procedures. It is important to note that as the SCD uses color space, color quantization and histogram descriptors, this would allow the color histogram to be represented by variable number of bins based on the desired accuracy.

3.4. **Color Layout Descriptor (CLD).** This descriptor [8], represents the spatial distribution of color features in an image. It is based on generating a thumbnail (64 blocks) of an image. To extract color information, first, the image is divided into a number of blocks, and each block contains $8{\times}8$ pixels. Next, a single dominant color is selected as the representative color from each block and transformed to the Y/Cb/Cr color space. The average color of each block is used as the representative color. Finally, the CLD is yielded by applying $8{\times}8$ Discrete Cosine Transform (DCT) on each component of the color space. A set of frequency DCT components of each YCbCr are selected in zigzag scanned order quantized to form a CLD [2].

3.5. **Color and Edge Directivity Descriptor (CEDD).** The CEDD descriptor [11], extracts multiple attributes that include color and texture combined in one histogram. The size of CEDD is limited to 54-bytes per image, which reduces the high dimensional vector of images and makes it more suitable to be used for large image databases. In CEDD, the image is divided into a number of blocks. Those image-blocks are then handled independently to extract their color information, through a 2 staged fuzzy histogram linking procedure that produces a 24-color histogram. To extract texture information, it uses a fuzzy version of the five digital filters proposed by the MPEG-7 [11], forming 6 texture areas.

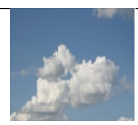
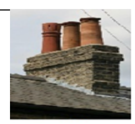
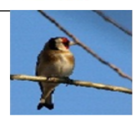
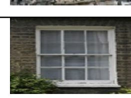
3.6. **Speed Up Robust Features (SURF) Combined with Fast Library Approximate Nearest Neighbor (FLANN).** SURF [3], is a visual descriptor that uses local feature detection to extract interest points, which are points that can be used to identify objects in the images. In addition, these points are the same for the rotated and scaled copy of the original file, and can provide reliable matching between different viewpoints of the same images. In general, the SURF descriptor is extracted by constructing a square region center aligned to a vertical and horizontal orientation. This region is classified into smaller 4×4 square sub-windows, for each sub-window, Haar wavelet responses are calculated. Then, the sum of the values of the responses are extracted for both vertical and horizontal orientation. Furthermore, the sum of the absolute values of both responses are extracted. Thus, each sub-window has a four-dimensional descriptor vector. The mechanism of SURF is summarized as follows; first, detecting the interest point: This approach focuses on blob-like structures [1] to detect the interest points in the image, which are allocated at junctions, speckles, and corners of objects and at locations where the determinant is maximal. The determinant of the Hessian matrix (DoH) includes various Gaussian filters at each point for scale selection to be convolved with the source image. Furthermore, to speed up the implementation of SURF, it uses an integral image, which is defined by the sum of all pixel values for all rectangular regions within the processed image. Second, Interest Point Description: The SURF describes the content of intensity distribution in the key-point neighborhood to provide a unique and robust description. The Haar wavelet is calculated for window regions within a circular area of the key-point's neighborhood, and the sum of both wavelet responses is computed. Next, the orientation of the processed window region is changed by 60°, and the Haar wavelet is re-computed.

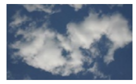
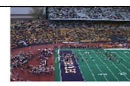
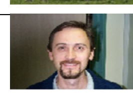
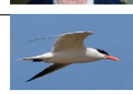
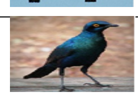
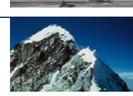
4. Experiments

In this section, the performance of the FCTH, EHD, SCD, CLD, CEDD and SURF combined with FLANN content-based methods have been investigated. To ensure the robustness of experiments results, multiple databases have been used in this study. These databases contain images that have been collected from different resources such as, Wang [12, 21], MIR Flickr [6], UCID [18], MSRCROID [15], etc. In addition, these images belong to a large variety of subjects and categories such as 'cloud', 'stadium', 'birds', 'sky', 'flowers', 'sea', etc., and have different sizes and extensions. Furthermore, in this study, C# language has been used to implement the studied methods, i.e. FCTH, EHD, SCD, CLD, CEDD, and SURF combined with FLANN. It is important to note that in the case of FLANN, which works based on SURF, the EmguCV has been integrated with C#. Postgresql 9.1, which is one of the most advanced open source database systems, and can be easily integrated with C#, has been used to create the needed databases. Last but not least, the used server has the following properties: Intel(R) Core (TM) i3, CPU M 350 at 2.27 GHz (4CPUs), 4 GB RAM, and Windows Seven 32-bit Operating System.
It is important to note that for the following five experiments, 10 query images were chosen randomly for each database, and the snapshots of the selected query is shown in Table 1.

Experiment #1: Using FCTH Method
In this experiment the performance of the FCTH method was investigated, and the number of relevant files for each query using the first 10, 20 and 30 results was found. The results of this experiment are shown in Table 1. For the first database, the number of relevant files using the first 10 files was at least 80%, and for the sixth query which belongs to the 'cloud' category, the percentage of relevant files was 100%. In addition, it has been observed that by increasing the numbers of images by using second and third databases, the numbers of irrelevant images have been increased for some queries. Hence, the performance of FCTH has been decreased. Furthermore, related to the third database, the followings have been observed: First, although, the percentage of relevant files was 100% for the first query of the category 'stadium', the number of irrelevant images was the highest for the seventh query of the category 'airplane', and most of the retrieved images belonged to 'birds', 'sky', 'balloons', and 'sea' categories. However, the retrieved images have the same background color in common with the query image. Second, most of the retrieved images for the ninth query 'mountain' category are not relevant, as most of the images in this category were taken in different seasons. For example, some images of snow-capped mountains were taken in the winter and some others were taken in the summer. Furthermore, some retrieval images contain buildings that look somehow similar to mountains.

Table 1: Number of relevant files using FCTH method for all databases (10.000, 50.000, and 100.000 Images)

Database #	Query #	Query Image	Number of Relevant files		
			First 10 files	First 20 files	First 30 files
First Database	Query #1		10	19	24
	Query #2		9	15	20
	Query #3		9	19	26
	Query #4		9	13	19
	Query #5		8	13	19
	Query #6		10	20	30
	Query #7		10	16	22
	Query #8		9	18	25
	Query #9		8	18	26
	Query #10		9	14	21
Second Database	Query #1		7	14	21
	Query #2		10	20	29
	Query #3		7	13	16

	Query #4		10	20	30
	Query #5		8	14	21
	Query #6		4	7	11
	Query #7		9	14	19
	Query #8		3	6	10
	Query #9		10	18	27
	Query #10		10	17	26
Third Database	Query #1		10	20	30
	Query #2		3	5	9
	Query #3		10	19	26
	Query #4		8	14	19
	Query #5		3	7	13
	Query #6		10	15	22
	Query #7		2	2	3
	Query #8		4	6	7
	Query #9		3	3	3
	Query #10		6	8	10

Experiment #2: Using EHD Method

In this experiment, the EHD descriptor was tested. Table 2 shows the number of relevant files for the 10 designated queries for each database. In general, successful retrieval systems need to extract the right features that represent the content of images and are sufficient in describing objects. It is very difficult to achieve the most relevant files using only a single feature type. Therefore, as this descriptor only describes and use edge based on local edge distribution, the results of most of the queries were below user's expectations, for instance, for the first database, first query, only two files out of ten were relevant, and for the second database, the fourth, eighth, and ninth queries produced zero relevant files. On the other hand, the performance of this method can be accepted, if the query image has few objects and the borders of these objects are clear, i.e. represented by enough number of pixels, for instance, as seen in the first database, fourth query of the category 'bicycle', and fifth and seventh queries of the category 'car', and in the second database fifth query of category 'car'. Last but not least, it has been observed that this descriptor cannot distinguish between images that contain objects that have approximately the same shape and borders, such as the sixth query and seventh query in the last database.

Table 2. Number of relevant files for queries using EHD method for all databases (10.000, 50.000, and 100.000 Images)

Database #	Query #	Number of Relevant files		
		First 10 files	First 20 files	First 30 files
First Database	Query #1	2	3	4
	Query #2	8	12	19
	Query #3	7	11	15
	Query #4	10	17	23
	Query #5	9	17	21
	Query #6	4	5	7
	Query #7	10	19	29
	Query #8	4	6	6
	Query #9	6	11	12
	Query #10	2	4	6
Second Database	Query #1	2	2	2
	Query #2	7	14	16
	Query #3	1	1	1
	Query #4	0	0	0
	Query #5	9	16	21
	Query #6	2	2	4
	Query #7	6	13	20
	Query #8	0	0	0
	Query #9	0	0	0
	Query #10	6	6	10
Third Database	Query #1	2	2	2
	Query #2	3	7	10
	Query #3	10	19	29
	Query #4	1	1	1
	Query #5	7	12	16
	Query #6	1	1	1
	Query #7	1	1	1
	Query #8	4	11	14
	Query #9	0	0	0
	Query #10	0	1	2

Experiment #3: Using SCD Method

In this experiment, the performance of the SCD method was investigated, and the number of relevant files for each query was found using the first 10, 20 and 30 results. The results of this experiment are shown in Table 3. The following have been observed:

(1) Using the first 10 results of the first database, the number of relevant files was at least 90%. In addition, for some queries such as the third and six queries, the percentage of relevant files was 100%.

(2) Using the second database, the overall performance was good, and this method was able to get 100% relevant files for some queries, such as the second query which belongs to the 'dinosaur' category, the third query of 'sport' category, fifth query of 'car' category, and seventh query of 'tree' category.
(3) Using the third database, it has been noticed that this descriptor is not accurate for huge databases, and the results for some queries were irrelevant. Hence, using the color feature only might not be sufficient to accurately describe the image's objects, for example most of the images retrieved for the seventh query contained different objects that are colored in blue. In addition, it is difficult to retrieve similar images for the eighth query of the 'bird' category based on the color feature only, as most of the query similar images were taken in different places that affect the content of the images. On the other hand, the percentages of the relevant images were 100%, for the first query of the category 'stadium' and the fourth query of the category 'sheep'. This is because of the fact that almost all the images in each of these categories have the same objects and colors.

Table 3. Number of relevant files for queries using SCD method for all databases (10.000, 50.000, and 100.000 Images)

Database #	Query #	Number of Relevant files		
		First 10 files	First 20 files	First 30 files
First Database	Query #1	10	19	27
	Query #2	10	20	29
	Query #3	10	20	30
	Query #4	9	17	24
	Query #5	10	19	28
	Query #6	10	20	30
	Query #7	10	18	25
	Query #8	10	18	21
	Query #9	10	17	24
	Query #10	10	20	24
Second Database	Query #1	9	18	25
	Query #2	10	17	25
	Query #3	10	16	24
	Query #4	10	20	29
	Query #5	10	18	27
	Query #6	5	10	11
	Query #7	10	17	27
	Query #8	7	7	7
	Query #9	6	12	16
	Query #10	9	16	25
Third Database	Query #1	10	20	30
	Query #2	10	17	25
	Query #3	10	20	29
	Query #4	10	20	30
	Query #5	2	4	7
	Query #6	8	16	23
	Query #7	2	4	5
	Query #8	3	5	5
	Query #9	2	3	3
	Query #10	9	16	21

Experiment #4: Using CLD Method

In general, CLD represents all images using the distribution of colors in that image, which means two images that contain totally different objects can be show as similar if its objects have the same colors. Table 4 shows the number of relevant files for the 10 used queries. Overall, as explained before, the number of irrelevant images can be high when objects have the same colors. On the other hand, the results of this method can be acceptable, whenever image's objects are represented by different colors, such as the first and ninth queries of the first database. Related to the second database, the overall performance of CLD has been improved by using a larger database as compared with the results of first database; for instance, the percentage of relevant files was 100%, for the first query of category 'window'. In addition, the number of relevant images is very high for some queries such as the seventh, ninth, and tenth queries and the results for some other queries are also good such as the second, fourth, and fifth queries. However, as mentioned before, this method might inadvertently retrieve some images that have the same colors of the query, even if these images actually have totally different objects. As shown in the results of the third database, this will increase the number of irrelevant retrieved images.

Table 4. Number of relevant files for queries using CLD method for all databases (10.000, 50.000, and 100.000 Images)

Database #	Query #	Number of Relevant files		
		First 10 files	First 20 files	First 30 files
First Database	Query #1	10	16	20
	Query #2	5	5	9
	Query #3	10	18	28
	Query #4	8	13	16
	Query #5	9	17	22
	Query #6	9	18	28
	Query #7	9	16	21
	Query #8	7	8	8
	Query #9	8	14	20
	Query #10	4	5	5
Second Database	Query #1	10	20	30
	Query #2	9	18	20
	Query #3	1	1	1
	Query #4	8	15	22
	Query #5	8	16	21
	Query #6	4	6	9
	Query #7	9	18	28
	Query #8	5	9	13
	Query #9	9	18	27
	Query #10	10	20	29
Third Database	Query #1	10	20	24
	Query #2	8	15	21
	Query #3	10	19	29
	Query #4	8	15	20
	Query #5	4	5	5
	Query #6	8	17	24
	Query #7	1	5	9
	Query #8	9	16	24
	Query #9	2	4	4
	Query #10	5	9	15

Experiment #5: Using CEDD Method

In this experiment, the CEDD descriptor was tested. Table 5 shows the number of relevant files for the 10 used queries for all databases. Using the first database, the overall number of relevant files is good. In addition, when checking the first 10 files this method was able to achieve 100% relevant files for most of the queries. Using the second database most of the results were acceptable. However, this descriptor cannot differentiate between two objects that have the same color and bounders, for instance the number of relevant files for the sixth query of category 'airplane' is low, and most the of retrieved images were from the category 'birds'. Using the third database, the results for some queries were highly relevant such as the first, third, fourth, and sixth queries; additionally, the results for the second and fifth queries are quite good. On the other hand, the performance of the texture part of this descriptor decreases when it is required to deal with objects that have unsymmetrical boundaries such as the ninth and tenth queries. Moreover, the number of irrelevant images is high for the seventh query of the category 'airplane', as most of the retrieved files were images of the 'birds' category that have edge and color similar to the query image.

Table 5. Number of relevant files for queries using CEDD method for all databases (10.000, 50.000, and 100.000 Images)

Database #	Query #	Number of Relevant files		
		First 10 files	First 20 files	First 30 files
First Database	Query #1	10	20	28
	Query #2	10	15	23
	Query #3	10	19	26
	Query #4	8	12	16
	Query #5	8	12	16
	Query #6	10	20	28
	Query #7	10	20	28
	Query #8	10	15	18
	Query #9	8	16	24
	Query #10	9	16	20
Second Database	Query #1	9	18	26
	Query #2	8	18	26
	Query #3	6	9	10
	Query #4	9	16	22
	Query #5	10	20	28
	Query #6	4	6	9
	Query #7	7	13	22
	Query #8	8	9	10
	Query #9	10	20	30
	Query #10	10	16	25
Third Database	Query #1	10	20	24
	Query #2	9	13	16
	Query #3	8	18	28
	Query #4	10	19	25
	Query #5	7	14	16
	Query #6	9	17	26
	Query #7	2	2	2
	Query #8	3	6	9
	Query #9	2	5	7
	Query #10	3	6	9

Experiment #6: Using SURF combing with FLANN

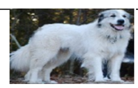
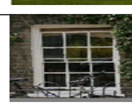
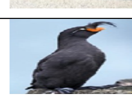
In this experiment, the performance of SURF combed with FLANN method was investigated using three databases that contained 1000, 2500, and 5000 images. In addition, for each database, 10 query images were chosen randomly, and the snapshots of these query images is shown in Table 6. Furthermore, for each query, the number of relevant files was found using the first ten, first twenty, and first thirty results. The results of this experiment are shown in Table 6. The results of this experiment can be summarized as follows.

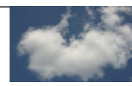
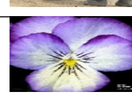
(1) Using the first database, it is clear that the number of the relevant images for the fifth query of the category 'tree' and the sixth query of the category 'window' were acceptable. However, the number of irrelevant images for some queries was high, as this approach neglects using the color, texture, and shape features individually, and instead it uses the interest points that are located at junctions or corners of the objects. Hence, the interest points are often not sufficient to find the similarities among image's objects that can be represented by multiple models and/or shapes. For instance, the number of relevant files for the eighth query of the category 'motorcycle' was very low.

(2) Using the second database, the relevant files for most of the queries were less than 50%. It is important to note that this method cannot find the exact interest point for the same objects that have been rotated, for instance, the ninth and tenth queries. In addition, as shown in the results of eighth query of the category 'cloud', where clouds can have different random shapes, which means that images of the category 'cloud' are represented by different interest points. Hence, this method cannot deal with objects that have multiple and different shapes.

(3) Using the third database, it has been observed that by increasing the size of the database, the number of irrelevant images also increases, and the numbers of relevant files of all queries were under user expectations.

Table 6: Number of relevant files for queries using SURF combing with FLANN for all databases (1000, 2500, and 5000 Images)

Database #	Query #	Query Image	Number of Relevant files		
			First 10 files	First 20 files	First 30 files
First Database	Query #1		2	4	6
	Query #2		7	12	15
	Query #3		1	1	1
	Query #4		3	3	3
	Query #5		8	17	26
	Query #6		10	20	28
	Query #7		5	9	11
	Query #8		0	2	3
	Query #9		5	7	10
	Query #10		6	9	11
Second Database	Query #1		2	3	4

	Query #2		2	4	5
	Query #3		8	12	17
	Query #4		8	13	16
	Query #5		10	16	18
	Query #6		3	8	12
	Query #7		4	11	13
	Query #8		2	3	3
	Query #9		5	7	9
	Query #10		2	3	3
Third Database	Query #1		1	1	1
	Query #2		2	4	5
	Query #3		0	0	1
	Query #4		4	7	8
	Query #5		3	7	8
	Query #6		1	1	1
	Query #7		1	2	2
	Query #8		0	1	1
	Query #9		1	3	5
	Query #10		3	7	8

5. Conclusions

Multimedia data has been growing rapidly in recent years, mostly due to the evolution of information technology. Nowadays, the image search engine is widely used with the proliferation of capture devices and the growth in digital images on the internet. The main role of this type of search engine is to retrieve and show the user relevant images. Text based image retrieval (TBIR), finds relevant files based on the query keyword (s). Finding images based on the text has some problems, for instance, sometimes the captions of images and/ or the corresponding link tagging may not actually be related to the content of the images itself. This might lead the users to obtain irrelevant, duplicate, and undesired images. To solve such problems, the CBIR system, which extracts features that describe the visual content of multimedia has been developed. In general, CBIR systems use color, texture, and shape features to describe the visual content of multimedia.

In this study, the performance of FCTH, EHD, SCD, CLD, CEDD, and SURF combined with FLANN content-based techniques have been investigated. The studied CBIR descriptors and its results are summarized as follows.

The FCTH descriptor combines color information and texture information in a single histogram, which is helpful to get good results. However, this technique cannot distinguish between objects that have almost the same properties such as shapes and borders.

The EHD method uses the edge feature to extract features that contain the object boundaries and sometimes they can work as shape techniques. The performance of this approach can be acceptable, if the borders of the object of the query image are regular, and clear, and/or if the query image has few objects. However, the overall performance of this method was below expectations. This is because of the fact that the EHD method divides the image into 16 sub-images and these sub-images are further divided into a number of image-blocks. This process may lead to inadvertently discarding important parts and features of the image objects. In addition, based on predefined thresholds, each image-block is represented by only one type of edge, however most times the image's blocks contain multiple types of edges. Hence, the performance of this descriptor decreases when it is required to deal with objects that have unsymmetrical boundaries.

The SCD descriptor uses only color information in image retrieval. Overall, the accuracy of this technique is good. However, it cannot find similar objects that have different colors. In addition, SCD has some difficulties in distinguishing between different objects that have the same color.

The CLD method represents any image using the distribution of colors in that image. The CLD descriptor uses only a color feature. This method can find the rotated and scaled files which are relevant to the query image. However, the number of irrelevant images can be high when the compared objects have the same colors.

The CEDD descriptor, extracts multiple attributes that include color and texture information which are combined in one histogram. Therefore, the results of this descriptor outperform other methods that use only a single feature type, such as EHD and CLD descriptors. On the other hand, as CEDD incorporates color and texture features in a single histogram, the drawback of this histogram is that it cannot precisely describe the image's complicated objects. Addition, it cannot differentiate between two different objects that have the same color and boundaries.

The SURF combined with FLANN uses the interest points that are located at junctions, speckles, and corners of the objects. These interest points can be used in CBIR systems, because they are not changed by the fluctuating rotation and possible differences in size. However, this method requires a massive memory and most times the interest points are not sufficient enough to find similarities among the image's objects that have and can be represented by multiple models and/or shapes.

Based on the results obtain using all the databases, FCTH and SCD descriptors were selected as the best descriptors. Hence, they have obtained the most relevant files compared to the other descriptors. On the other hand, for the SURF combined with FLANN approach, the results of most of the queries were below expectations. Overall, none of the studied CBIR descriptors can be used individually to build a full image retrieval system. In our opinion, multiple descriptors can be used to achieve a more robust system and accurate results.

In future work, the study presented in this work can be further improved by testing the performance of integrating multiple descriptors and building an image retrieval system which uses the best group of descriptors simultaneously.

References

[1] Amanatiadis, A., Kaburlasos, V.G., Gasteratos, A., Papadakis, S.E., *Evaluation of shape descriptors for shape-based image retrieval*, Image Processing, IET, **5(5)**(2011), 493–499. 1, 3.6

[2] Balasubramani, R., Kannan, D.V., *Efficient use of MPEG-7 color layout and edge histogram descriptors in CBIR systems*, Global Journal of Computer Science and Technology, **9(4)**(2009), 157–163. 3.4

[3] Bay, H., Ess, A., Tuytelaars, T., Van Gool, L., *Speeded-up robust features (SURF)*, Computer vision and image understanding, **110(3)**(2008), 346–359. 2, 3.6

[4] Chatzichristofis, S.A., Boutalis, Y.S., FCTH: Fuzzy color and texture histogram-a low level feature for accurate image retrieval, In Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08. Ninth International Workshop on IEEE, (2008), 191–196. 2, 3.1

[5] Duan, L., Li, W., Tsang, I.W.H., Xu, D., *Improving web image search by bag-based reranking, Image Processing*, IEEE Transactions on, **20(11)**(2011), 3280–3290. 1

[6] Huiskes, M.J., Thomee, B., Lew, M.S., New trends and ideas in visual concept detection: the MIR flickr retrieval evaluation initiative, In Proceedings of the international conference on Multimedia information retrieval, ACM, (2010), 527–536. 4

[7] Jain, H., Lodha, R., Deulkar, K., *Web-based Image Search using Image Re-Ranking Technique: A Review Paper*, International Journal of Current Engineering and Technology, **4(5)**(2014), 3300–3303. 2

[8] Jalab, H.A., Image retrieval system based on color layout descriptor and Gabor filters, In Open Systems (ICOS), 2011 IEEE Conference on IEEE, (2011), 32–36. 2, 3.4

[9] Kekre, D.H., Thepade, S.D., Mukherjee, P., Wadhwa, S., Kakaiya, M., Singh, S., *Image retrieval with shape features extracted using gradient operators and slope magnitude technique with BTC*, International Journal of Computer Applications (IJCA), **6(8)**(2010), 28–33. 1

[10] Khokher, A., Talwar, R., Content-based Image Retrieval: Feature Extraction Techniques and Applications, In Conference proceedings, Proceedings published in International Journal of Computer Applications (IJCA), (2012), 9–14. 1

[11] Kumar, P.P., Aparna, D.K., Rao, K.V., *Compact Descriptors for Accurate Image Indexing And Retrieval: Fcth And Cedd*, International Journal of Engineering Research and Technology (IJERT), **1(8)**(2012). 2, 3.5

[12] Li, J., Wang, J.Z., *Automatic linguistic indexing of pictures by a statistical modeling approach*, IEEE Transactions on pattern analysis and machine intelligence, **25(9)**(2003), 1075–1088. 4

[13] Manjunath, B.S., Ohm, J.R., Vasudevan, V.V., Yamada, A., *Color and texture descriptors*, IEEE Transactions on circuits and systems for video technology, **11(6)**(2001), 703–715. 2, 3.3

[14] M.Tech, V., Jagadeesan, J., Isaac, A., *Web Image Search Reranking Using CBIR*, International Journal of Computer Science & Engineering Technology (IJCSET), **5**(2014)(2229-3345), 348–360. 1

[15] Microsoft, Microsoft Research Cambridge Object Recognition Image Database, (2016) Retrieved from "https://www.microsoft.com/en-us/download/details.aspx?id=52644". 4

[16] Ordonez, V., Han, X., Kuznetsova, P., Kulkarni, G., Mitchell, M., Yamaguchi, K., Stratos, K., Goyal, A., Dodge, J., Mensch, A., Daumé III, H., *Large scale retrieval and generation of image descriptions*, International Journal of Computer Vision, **119(1)**(2015), 46–59. 2

[17] Pedronette, D.C.G., Torres, R.D.S., *Image re-ranking and rank aggregation based on similarity of ranked lists*, Pattern Recognition, **46(8)**(2013), 2350–2360. 2

[18] Schaefer, G., Stich, M., *UCID: An uncompressed color image database*, In Storage and Retrieval Methods and Applications for Multimedia, **5307**(2003), 472–481. 4

[19] Syed, H.S., Arif, I.U., Saeeda, N., Noor ul, A.Kh., Muhammad, I.R., AlHaqbani, B., *Content-Based Image Retrieval Using Texture Color Shape and Region*, International Journal of Advanced Computer Science and Applications (IJACSA), **7(1)**(2016), 418–426. 1

[20] Wang, X., Liu, K., Tang, X., Query-specific visual semantic spaces for web image re-ranking, In Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference, (2011), 857–864. 2

[21] Wang, J.Z., Li, J., Wiederhold, G., *SIMPLIcity: Semantics-sensitive integrated matching for picture libraries*, IEEE Transactions on pattern analysis and machine intelligence, **23(9)**(2001), 947–963. 4

[22] Wang, X., Qiu, S., Liu, K., Tang, X., *Web Image Re-Ranking Using Query-Specific Semantic Signatures*, IEEE Transactions on Pattern Analysis and Machine Intelligence, **36(4)**(2014), 810–823. 2

[23] Won, C.S., Park, D.K., Park, S.J., *Efficient use of MPEG-7 edge histogram descriptor*, ETRI journal, **24(1)**(2002), 23–30. 2, 3.2

[24] Yang, L., Hanjalic, A., *Prototype-based image search reranking, Multimedia*, IEEE Transactions on, **14(3)**(2012), 871–882. 2

[25] Yu, J., Rui, Y., Chen, B., *Exploiting click constraints and multi-view features for image re-ranking*, Multimedia, IEEE Transactions on, **16(1)**(2014), 159–168. 2

[26] Zhang, D., Lu, G., *Review of shape representation and description techniques*, Pattern recognition, **37(1)**(2004), 1-19. 1