Saliency-weighted graphs for efficient visual content description and their applications in real-time image retrieval systems

Saliency Weighted Graphs for Efficient Visual Content Description and Their Applications in Real-Time Image Retrieval Systems

[1]Jamil Ahmad, [2]Muhammad Sajjad, [1]Irfan Mehmood, [3]Seungmin Rho, [1,*]Sung Wook Baik

[1]College of Electronics and Information Engineering, Sejong University, Seoul, Republic of Korea

[2]Department of Computer Science, Islamia College, Peshawar, Pakistan

[3]Department of Multimedia, Sungkyul University, Anyang, Republic of Korea

jamilahmad@sju.ac.kr, muhammad.sajjad@icp.edu.pk, irfanmehmood@sju.ac.kr, smrho@sungkyul.edu, sbaik@sejong.ac.kr

Abstract

The exponential growth in the volume of digital image databases is making it increasingly difficult to retrieve relevant information from them. Efficient retrieval systems require distinctive features extracted from visually rich contents, represented semantically in a human perception oriented manner. This paper presents an efficient framework to model image contents as an undirected attributed relational graph, exploiting color, texture, layout, and saliency information. The proposed method encodes salient features into this rich representative model without requiring any segmentation or clustering procedures, reducing the computational complexity. In addition, an efficient graph matching procedure implemented on specialized hardware makes it more suitable for real-time retrieval applications. The proposed framework has been tested on three publicly available datasets, and the results prove its superiority in terms of both effectiveness and efficiency in comparison with other state-of-the-art schemes.

Keywords: attributed relational graph, image representation, content based image retrieval, saliency map, real-time retrieval

1. Introduction

Improvements in image acquisition technologies, availability of acquisition devices through smart phones, and the exponential increase in storage capacities have produced huge image databases. This huge visual data is unstructured and does not have a pre-defined model, making its understanding and management difficult using traditional data management systems. Effective and efficient retrieval of semantically relevant information from such huge data is a big challenge for multimedia researchers [1]. Highly focused research is underway around the world on the problem of pair-wise image matching, since this is the core component of content based image retrieval (CBIR) systems [2]. For this purpose, various models have been proposed for static visual content representation. Contents refer to the color, texture, and shapes of objects in images [3]. Every model tries to capture these characteristics of images in some way. It is usually desired to have a compact, distinctive, and effective representation of the visual contents that allows accurate comparison between images, ensuring high performance in retrieval applications. Numerous ways exist in the literature for image representation using both local and global features. Local features like scale invariant features transform (SIFT) [4], speeded-up robust features (SURF) [5], binary robust independent elementary features (BRIEF) [6] and the bag-of-words (BOW) model [7] have been widely used for image retrieval. However, these methods suffer from high computational cost and memory requirements due to their high dimensional nature.

[*] Corresponding author: sbaik@sejong.ac.kr

Global color descriptors like color histograms [8-10], color correlograms [11], MPEG-7 representations namely dominant color descriptor, color structure descriptor, color layout descriptor and scalable color descriptors [12] are used for color based image retrieval. However, they do not capture the texture and shape features in images and hence fail to adequately represent the rich visual contents. Similarly, texture descriptors such as texture browsing [13], edge histogram [14], homogeneous texture descriptors [15], local binary patterns (LBP) [16] and its variants [16-20] are exploited for texture based image retrieval. Shape contexts [21], moment descriptors, grid descriptors [22] and graph features [23] extracted from the object shapes allow images retrieval using shapes. In short, using only one characteristic is considered insufficient and such systems fail to cope with images of all types.

Aggregation of multiple features have also been investigated for CBIR systems. They include micro-structure descriptor (MSD), color difference histogram (CDH) [24], color and edge directivity descriptor (CEDD) [25], colorTon distribution descriptor [26], multi-texton histogram (MTH) [27] and fusion framework [28]. Each of these methods tends to capture multiple aspects of the image contents into spatially correlated histograms, representing colors, edges, and shape features to have a complete and compact representation. However, none of these methods are capable of capturing the characteristics in its entirety, hence reducing their performance and efficiency for large databases.

Theories related to human vision system can be helpful in CBIR by representing images in a form that is closer to human interpretation. For instance, visual saliency models can be incorporated in content representation frameworks for identifying perceptually significant regions, grabbing human attention [29]. This approach looks interesting and intuitive due to the fact that humans tend to look for specific objects in images while searching through the large databases. Hence, if incorporated correctly, saliency models can bring significant improvements to retrieval systems.

In this paper, we represent salient features of images as undirected attributed relational graphs (ARGs). Image features are represented as nodes, and the connecting edges signify their relationships. The proposed representation cohesively models salient low level features and their relationships that support convenient comparison between visual contents. Modeling visual saliency as ARGs can help in retrieval of semantically relevant images from large databases.

The main contributions of this paper are as follows:

(i) An effective method to model the salient low-level features using ARGs,
(ii) An efficient graph matching method to solve the pair-wise image matching problem, reducing computational complexity, and
(iii) A GPU based implementation of the proposed framework for real-time image retrieval systems.

The remainder of this paper is organized as the following: Section 2 presents an overview of the related works; the proposed method is explained in Section 3; Section 4 discusses experimental results of the proposed method over three datasets, and section 5 concludes the paper.

2. Related Work

The human vision system's sensitivity to colors and edge orientations have motivated researchers to exploit these characteristics into their frameworks for image representations. Numerous color and texture descriptors utilizing these attributes of the visual data has been developed. This section presents some of the notable contributions to the field of image retrieval over the recent past.

The CDH [24] exploits color and edge orientations along with perceptually uniform color differences to encode these features into a manner that is similar to the human vision system. It neither requires any clustering nor learning processes, making the feature extraction process very efficient. The CDH was tested on Corel dataset with significant improvements in retrieval performance. However, treating each color and edge equally affects its performance, because from perception point of view, they are not equally important. Mostafa and Mohsen [26] introduced the color ton distribution descriptor (CTDD) for CBIR systems. It extracts color distributions in the RGB color space through a co-occurrence matrix. They also used a self-organizing map as a classifier to detect the class of the query image and also performed segmentation for extracting features from various image regions. This method fails in the presence of intense color distributions. Secondly, image segmentation can also be a bottleneck in its performance. Liu et al. [30] presented a method called micro-structure descriptor (MSD) that tries to simulate early human vision processing by integrating low-level features into a compact representation. It is built using colors in micro-structures having the same edge orientations. The descriptor is low dimensional and performed efficiently on the Corel dataset [31]. Wang et al. [32] introduced a similar descriptor called structure element descriptor (SED) which describes image content in the HSV color space using 5 different structuring elements. Basically, it is a texture descriptor and detects texton in images to populate a histogram. The descriptor was tested on the Corel 10K dataset. Utilizing rich visual content in the most efficient and effective way is the key to develop a high performance CBIR system. In brief, the present techniques fail to encompass this richness into their frameworks and hence are unable to achieve acceptable performance.

Fusion based approaches have been developed with the idea that single feature representation cannot accurately represent the heterogeneous and complex structures in contents with sufficient discrimination. Hence, different methods are combined together by researchers to get improved performance. In [28], Ekta and Aman presented a fusion based approach by combining enhanced color difference histogram (ECDH) and local angular radial transform (L-ART) to represent images. It was observed that this scheme gained significant retrieval accuracy over the two descriptors alone. Similarly, Ahmad et al. [22] presented a weight based fusion scheme for shape based image retrieval. They developed a labeled-grid based approach for extracting a number of features from shapes. Then, they used a weighted ranking scheme to assign weights to different similarity measures derived from various features. Weights correspond to their importance in a particular query. Significant improvements were achieved during the experiments. Such fusion based approaches undoubtedly bring improvements but increase the computational burden, making the retrieval system infeasible for use with online systems.

In addition to the basic visual feature-set, saliency information is also being incorporated into content description frameworks to achieve such a representation that mimics the human visual system. Our brain and vision systems work together to efficiently locate the salient image regions. It limits the use of mental resources considerably by assigning very little visual processing resources to less salient regions, making the perception process highly efficient [33]. Although the schematics of human attention mechanism are not known exactly [34], yet computer vision researchers have devised theories for modeling the human visual perception system for detecting visual saliency. Using this saliency information in images can help in developing robust and effective image retrieval systems. Huang et al. [35] presented a method for representing images as undirected graphs exploiting saliency information. Intra-image similarity is measured by calculating the similarity of their graphs. The method produced excellent results on three publicly available datasets. However, the procedure encodes salient super pixels which require segmentation procedures, hence suffers from the same issues as the other segmentation based schemes. Also, it does not take into account the texture, edge and shape features, therefore, fails to effectively represent the principal image attributes.

The proposed framework captures low-level color and texture features along with saliency information to build a semantic representation of the visual contents using an undirected ARG. It is different from existing schemes in the sense that it does not require segmentation or clustering, improving features extraction. Furthermore, it supports convenient node-to-node and edge-to-edge mapping, allowing efficient matching of two ARGs. This improvement to pair-wise image matching eventually enhances performance of CBIR systems.

3. Proposed Method

Psychophysical and neurobiological studies revealed that the human vision system (HVS) has a higher sensitivity to color and edge orientations in images [36-38]. These characteristics play a significant role in visual content analysis and understanding. Therefore, we tend to represent these features as ARGs with the confidence that they have strong ability for pattern representation. Figure 1 shows the main steps involved in the proposed framework. The different phases are explained in the subsequent sections. A summary of the model parameters used in the proposed framework are provided in Table 1 for easy understanding.

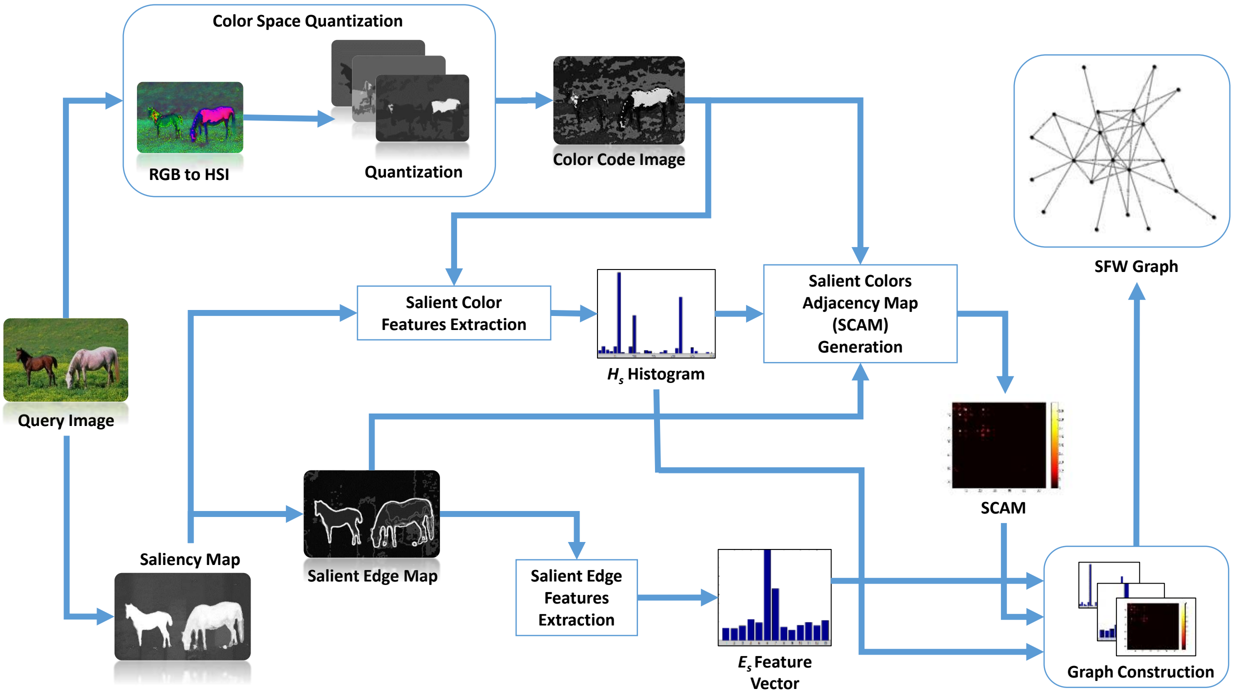


Figure 1: The proposed SFW graph formation framework

3.1 Color Space Quantization

Color perception is a core and fundamental component of primate vision, helping in salient object detection and recognition. HVS is capable of discerning thousands of colors but merely two dozen shades of gray [39]. This fact verifies the importance of colors in visual perception and understanding. In order to simplify the color perception modeling, only main representational colors are considered instead of the thousands of colors that an image contains [40]. Therefore, color quantization is employed in the proposed framework before extracting color information from images.

Color space selection and quantization are fundamental steps towards image representation. In the presence of so many colors models, selection of the appropriate model is a challenge. However, the choice of color model and the amount of quantization depends on the target application. Therefore, we conducted several experiments for selecting the appropriate color model and levels of quantization. We observed that the proposed approach works best with HSI (hue, saturation, intensity) color space and 2-bits quantization per color component. The HSI model is widely used in image processing applications because it represents color in a form that is natural and intuitive to humans. It decouples the chromatic and achromatic components and makes it consistent for the visual content to be easily interpreted by humans [39]. The levels of quantization typically limit the number of representative colors in images. Too many levels effectively destroy the whole purpose of quantization, whereas fewer levels in quantization fail to capture the necessary number of colors for effective representation. Hence, the tradeoff should be maintained in efficient and effective color quantization.

During the quantization phase, the input RGB image is first converted to HSI color space [39] and then the corresponding H, S and I components are quantized using the following equations with $\mathcal{Q}_{\text{bits}}$=2 bits/plane:

$$\mathcal{H}_q = \left\lfloor \frac{\mathcal{H}}{360} \times 2^{\mathcal{Q}_{\text{bits}}} \right\rfloor, \; \mathcal{S}_q = \left\lfloor \frac{\mathcal{S}}{255} \times 2^{\mathcal{Q}_{\text{bits}}} \right\rfloor \text{and} \quad \mathcal{I}_q = \left\lfloor \frac{\mathcal{I}}{255} \times 2^{\mathcal{Q}_{\text{bits}}} \right\rfloor \tag{1}$$

where $\mathcal{H}_{\text{q}}$, $\mathcal{S}_{\text{q}}$ and $\mathcal{I}_{\text{q}}$ are the quantized hue, saturation and intensity components, respectively. Each of these components are composed of values in the range $[0 - 2^{\text{qbits-1}}]$. Uniform quantization was used for all these components, because it performed the best with our approach. After quantization, color code image $\mathcal{C}$ is generated which represents each quantized color with a unique integer value c. Color codes are generated by concatenating binary codes of corresponding quantized color components as:

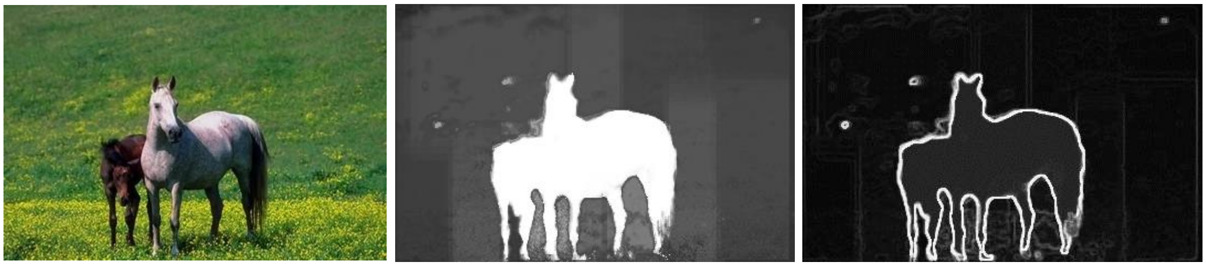
$$\beta = [\text{bits}(\mathcal{H}_q), \text{ bits}(\mathcal{S}_q), \text{ bits}(\mathcal{I}_q)] \tag{2}$$

where bits(…) is a function that maps bits to a decimal value and $\beta \in \mathcal{R}^{\text{n}}$ is a matrix constructed by concatenating bits from all three component images. Converting β into decimal values for all the pixels give us the color code image $\mathcal{C}$. The color code value for a pixel p is calculated as the decimal value represented by all the bits in β as :

$$\mathcal{C}_{\mathcal{P}} = \sum_{i=0}^{3\times\mathcal{Q}_{\text{bits}}-1} \beta(i) \times 2^{3\times\mathcal{Q}_{\text{bits}}-1-i} \tag{3}$$

3.2 Saliency Map Generation

The HVS is reliably efficient in locating the visually salient parts of a scene [41]. Exposure to the salient regions in the visual field helps us in efficiently utilizing the limited perceptual resources during visual activities. Therefore, computer vision based methods for finding salient regions are utilized during image analysis. In this work, we have employed saliency measure as weights for color and edge features. Higher weights are given to features located in the salient part of the image, whereas lower weights are assigned to the rest of the image. A sample image and its saliency map are depicted in figures 2 (a) and (b) respectively.



(a) (b) (c)

Figure 2: (a) Image, (b) saliency map using method in [42] and (c) salient edge map

In order to detect salient regions, several existing saliency detection methods were evaluated including [42-44]. For our framework, we needed a method that is efficient and effective. The method described in [42] was found to be the most suitable one. In their method, the authors have formulated saliency as a statistical framework with local feature contrast in motion, color and illumination to construct an initial map. A conditional random field (CRF) model then uses this saliency map for energy minimization based segmentation in an attempt to recover salient objects. The final saliency map S_{map} is generated having higher values for the visually salient parts than the rest of the scene. These saliency values S_{xy} are used in the proposed framework for assigning weights to the extracted color and edge features which are represented as node and edge attributes, respectively.

3.3 Edge Features Extraction

From the image S_{map}, an edge map is extracted using canny edge detection algorithm referred to as the salient edge map $S\mathcal{E}_{\mathrm{map}}$. This edge map contains salient edges among the salient colors in the image under analysis. A sample salient edge map is given in figure 2(c). In our approach, we have ignored the relatively shorter edges as they are considered to be of little importance.

For pixels in this edge map with value greater than some edge threshold $\tau_{\mathbf{e}}$, a salient edge feature $\mathcal{E}_{\mathbf{s}}$ is computed. It takes into account the color adjacency to determine edge lengths between them and then uses the saliency value of the edge to derive $\mathcal{E}_{\mathbf{s}}$ as:

$$\mathcal{E}_{\mathrm{s}}(c_1, c_2) = S_{\mathrm{xy}} \times \sum p \in \{\, S\mathcal{E}_{\mathrm{map}}(c_1, c_2) > \tau_e \,\} \tag{4}$$

$$\mathcal{E}_{\mathrm{s}}(c_1, c_2) = \frac{\mathcal{E}_{\mathrm{s}}(c_1, c_2)}{\max(\mathcal{E}_{\mathrm{s}})} \tag{5}$$

where c_1 and c_2 are the two adjacent colors, p is the edge pixel between these two colors and $S_{\mathbf{xy}}$ is the saliency value of the corresponding edge. $\mathcal{E}_{\mathbf{s}}$ is the salient edge feature value for the two colors normalized to the maximum which measures the saliency weighted relative edge length in the underlying image. Longer edges are believed to attract more attention than the smaller ones and therefore, it is used as a factor for indicating perceptual significance [41].

3.4 Color Features Extraction

The saliency map $S_{\mathbf{map}}$ and color code image C are used to extract the salient color features by populating a weighted normalized histogram for the color code image. The saliency values $S_{\mathbf{xy}}$ are incorporated into the saliency weighted normalized color histogram $\mathcal{H}_{\mathbf{s}}$ as:

$$\mathcal{H}_{\mathrm{s}}(cc) = n \times S_{\mathrm{xy}} \tag{6}$$

$$\mathcal{H}_{\mathrm{s}}(c) = \frac{\mathcal{H}_{\mathrm{s}}(c)}{\max(\mathcal{H}_{\mathrm{s}})} \tag{7}$$

where n is the number of pixels with color code c and S_{xy} is the saliency value at the position of the pixel in the image $S_{\mathbf{map}}$. Colors with greater contribution to the perception field and saliency are considered more important than the colors with less saliency and less proportion in the image. Thus, values of this histogram indicate the relative importance of a color in a particular image. These weights help in graph matching when two nodes representing colors are compared. The values are normalized to maximum in order to allow a fair comparison between two colors and to avoid the effect of image size on it.

Table 1: Summary of input and output parameters in the proposed framework

Description of model parameters			
$\mathcal{H}, \mathcal{S}, \mathcal{I}$	The Hue, Saturation and Intensity components of image	$\mathcal{H}_s$	Normalized histogram
$\mathcal{H}_q, \mathcal{S}_q, \mathcal{I}_q$	The quantized hue, saturation and intensity components respectively	n	Number of pixels having same color code (i.e. frequency of a color)
$\mathcal{Q}_{bits}$	No of bits used in quantization	$\mathcal{W}_i$	Weight of the node *i* in the SFW graph
β	An array of quantized hue, saturation and intensity bit values	$\mathcal{E}_{i,j}$	Weight of the edge between nodes *i* and *j*
C	Color code image having unique numeric IDs for each quantized color (HSI)	$\mathcal{G}$	The salient features weighted (SFW) graph
C_p	Color code for a single pixel *p*	$\mathcal{N}, \mathcal{E}, \mathcal{A}$	Set of nodes, edges and attributes in the SFW graph
S_{map}	Saliency map obtained from [42]	σ, ε	Color and edge dissimilarity weights
$S\mathcal{E}_{map}$	Salient edge map obtained from S_{map}	$\mathcal{D}_s$	Graph Dissimilarity score
τ_e	Edge threshold for eliminating trivial edges	$\mathcal{P}, \mathcal{R}$	Precision and Recall measures
S_{xy}	Saliency value at pixel location (x,y), obtained from S_{map}	$\mathcal{AP}$	Average Precision
$\mathcal{E}_s$	Edge saliency value	$\mathcal{MAP}$	Mean Average Precision

3.5 Colors Adjacency Map Generation

A color adjacency map is generated for the image, representing adjacency among colors. The values in this map correspond to the saliency weighted relative length of the edge between two colors. Longer and salient edges are represented as larger values and smaller, less salient edges are represented as smaller values. Zeros represent no adjacency between colors. The salient colors adjacency map (SCAM) is basically a symmetric matrix, so only the upper triangular part is considered. Rows correspond to one color, and the columns represent the other color. The values at (x,y) correspond to the $\mathcal{E}_s$, calculated previously in equations 4 and 5.

3.6 Salient Features Weighted (SFW) Graph

In an undirected ARG, the salient colors and edges in an image are modeled as nodes and edges, respectively. Each node in the SFW graph contains two attributes, i.e., n_{id} is the color code C, kept for node to node mapping while graph matching, and $\mathcal{W}$ is the weight of the node taken from the $\mathcal{H}_s(C)$. These weights are actually compared to measure dissimilarity among graph nodes after mapping on the basis of nodes IDs. A perfect mapping will report minimum dissimilarity, and a no-match will result in maximum dissimilarity. During graph construction, nodes are identified from the SCAM. Only those colors are represented as nodes which have weights higher than some threshold t_c. We have an edge $e_{i,j}$ in the graph only when there is a salient edge between colors i and j, i.e. $\mathcal{E}_s$ values greater than a threshold τ_e. The longest edge has a value of 1.0 and smaller edges have relatively smaller values. These edge weights are considered when measuring similarity between two edges. Hence, our SFW graph is a triple, $\mathcal{G} = (\mathcal{N}, \mathcal{E}, \mathcal{A})$ where $\mathcal{N}$ is the set of nodes, $\mathcal{E}$ is the set of edges between the nodes and $\mathcal{A}$ is the set of attributes that these nodes and edges carry. The node attributes n_{id} and $\mathcal{W}_i$ are attached to node $n_i \in \mathcal{N}$, and $\mathcal{E}_{i,j}$ is the weight of the edge between nodes i and j, $e_{i,j} \in \mathcal{E}$.

The number of nodes in the graph depends upon quantization, the number of salient colors in an image and the threshold value τ_e. The threshold τ_e is important, because it controls the graph density. If lower τ_e is chosen, there will be a large number of edges resulting in a denser graph with a greater chance of having less important edges among potentially less important colors. Hence, carefully chosen τ_e is important for effective representation. Several experiments were conducted for selecting the optimal value for this threshold.

The proposed SFW graph is less likely to contain any isolated nodes or double edges for normal images. A constant image with just a single color is the only case when an image is represented with a single-node graph. In normal circumstances, the SFW graph is a rich representation of the visual contents. The presence of node and edge attributes make them easy to map and match. These attributes are carefully chosen to achieve geometric transformation invariance, making it a robust descriptor. The information regarding nodes and edges can be represented as sparse arrays while storing for indexing. The SFW graphs for a sample color image, and the mapping derived between two graphs is shown in figures 3 and 4, respectively.

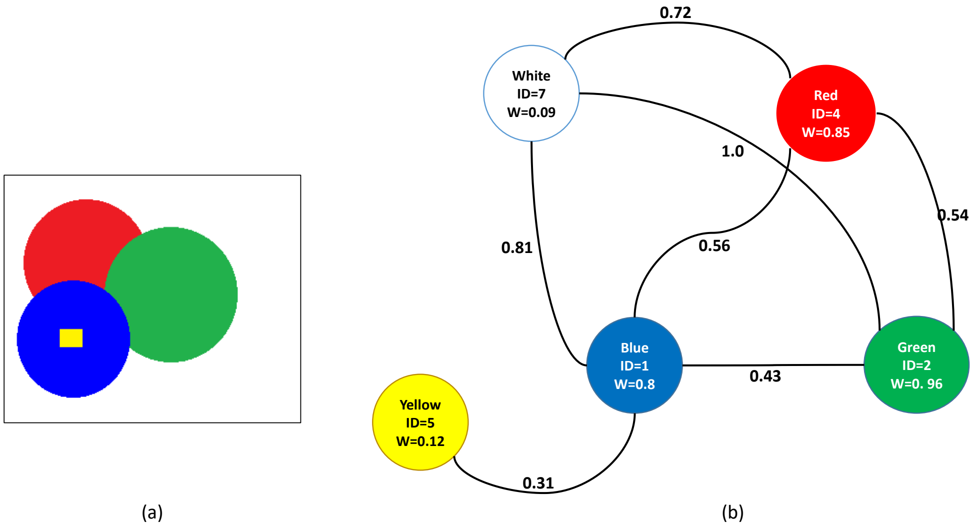


Figure 3: (a) Image and (b) its SFW graph

It can be seen from the graph in figure 3 that colors are represented as nodes and the edges between them are represented as edges. Node attributes indicate node ids and weights. Similarly, edge weights represent their relative lengths in the image. Longer edges have higher weights whereas shorter ones have low weights. The colors red, green, blue and yellow are the salient ones carrying greater saliency values than the white background. Although the color, white is in greater proportion, it carries the lowest weight. Green color is salient and also in greater proportion in the image, so it carries the largest weight. Yellow color is in smaller proportion and hence has a smaller weight. The longest edge is between the colors white and green, so it is indicated as 1, whereas the smallest edge is between blue and yellow, hence its relative weight is 0.31.

3.7 Graph Matching

For query image q and image $\mathbf{y}$ taken from the database $\mathcal{DB}$, the similarity is derived on the basis of how well their graphs $\mathcal{G}_{\mathbf{q}}$ and $\mathcal{G}_{\mathbf{y}}$ are matched. For the two graphs, a mapping is derived between them by taking into account the $n_{\mathbf{id}}$ attribute. Two nodes with similar $n_{\mathbf{id}}$ can be mapped for deriving a dissimilarity value between them. Two images having a node representing similar colors will always have the same $n_{\mathbf{id}}$, making it easy to derive a mapping between nodes, as shown in figure 4. In the proposed scheme, mapping between graphs is represented as $\mathcal{G}_{\text{map}}=(\mathcal{N}_{\text{map}}, \mathcal{E}_{\text{map}})$, where $\mathcal{N}_{\text{map}}$ is the node mapping i.e. a set of node pairs being mapped in the two graphs and $\mathcal{E}_{\text{map}}$ is the edge mapping, the set of mapped edge pairs. The dissimilarity score $\mathcal{D}_s$ between two mapped graphs is computed by considering their node-to-node mapping $\mathcal{N}_{\text{map}}$ and edge-to-edge mapping $\mathcal{E}_{\text{map}}$ as:

$$\mathcal{D}_s = \left[\sigma\sum_{i=1}^{m}\Delta\mathcal{N}_i + \varepsilon\sum_{j=1}^{n}\Delta\mathcal{E}_s^j\right] + \left[\sigma\sum_{k=1}^{s}\mathcal{W}_k + \varepsilon\sum_{\ell=1}^{t}\mathcal{E}_s^{\ell}\right] \qquad (8)$$

where m is the number of mapped node pairs in $\mathcal{N}_{\mathbf{map}}$, n shows the number of mapped edge pairs in the graphs, $\Delta\mathcal{N}_{\text{i}}$ correspond to the dissimilarity between the mapped nodes and ΔE_s^j represents the dissimilarity between mapped edge pairs. s is the number of unmapped nodes; $\mathcal{W}_{\mathbf{k}}$ are their weights; **t** is the number of unmapped edge pair; and, E_s^l are their weights. σ and ε are the weights that a user may assign to the color or edge features when retrieving similar images. It allow users to have a degree of freedom to set a balance between color and edge features while executing a query. The dissimilarity scores between mapped nodes and edge pairs are computed as:

$$\Delta\mathcal{N}_i = \left|\text{Deg}(n_{i1}) - \text{Deg}(n_{i2})\right| + \left|\mathcal{W}_{n1} - \mathcal{W}_{n2}\right| \text{ and } \Delta\mathcal{E}_j = \left|\mathcal{E}_s^{e1} - \mathcal{E}_s^{e2}\right| \qquad (9)$$

Here, Deg(n) returns the degree of the node $n_{\mathbf{i}}$ and $\mathcal{W}_{\mathbf{n}}$ is the weight of the nth node. A low $\mathcal{D}_{\mathbf{s}}$ indicates a better match whereas a high value specifies a weaker match.

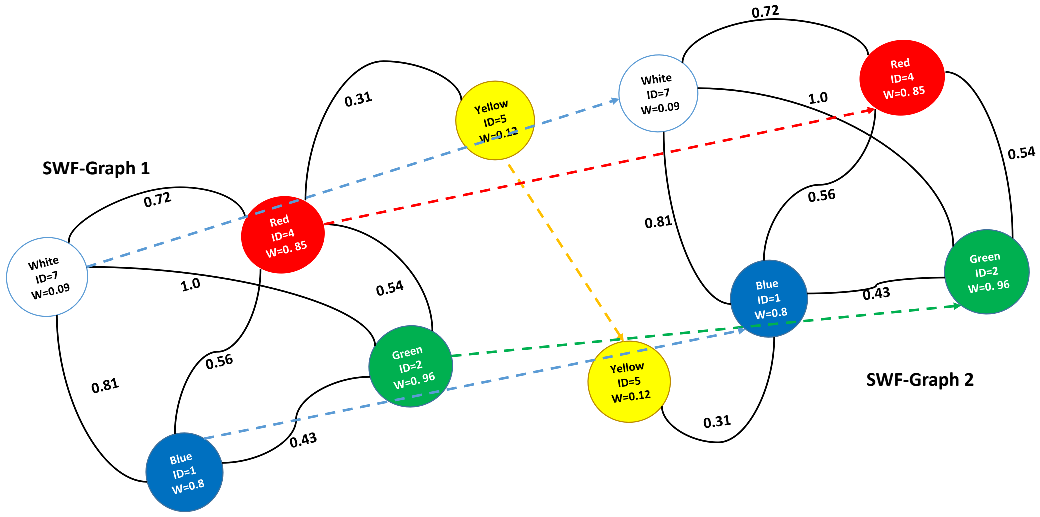


Figure 4: Graph mapping between two graphs, nodes are represented as color coded circles, edges are represented as solid black lines, dotted lines show node-to-node mapping

We formulate the problem of retrieving relevant images from a large dataset by representing them as ARGs and then retrieving the perceptually relevant images through graph matching ranked by the matching scores among the query and other images. We believe that if a discriminative feature-set is used to construct the graph, it will substantially enhance the performance of retrieval applications.

4. Experiments and Results

The performance of the proposed scheme was evaluated in image retrieval and retrieval of visually similar frames from videos. This evaluation was conducted on three publicly available image datasets including Corel dataset [45], Zurich-Buildings dataset (ZB) [46] and UKBench (UKB) [47]. The reason behind the selection of these datasets is that all of them have been widely used in the evaluation of many state-of-the-art methods. These datasets provide sufficient variety and complexity of visual contents for effective evaluation. In addition to this, we also tested its performance in retrieval of similar frames from several YouTube videos. Highly satisfactory results were obtained across all tests. Details of the experimental setup, datasets, evaluation metrics used and results are explained in the following subsections.

4.1 Experimental Setup

All experiments were conducted on a Windows 7 desktop PC having Intel Core i5-4670 CPU @ 3.40 GHz and 8.0 GB RAM. MATLAB 2014a was used to implement and evaluate the algorithm. For implementation on the graphic processing unit (GPU), we used an NVIDIA GeForce GTX 650 GPU [48] having compute capability 3.0 with Parallel Computing Toolbox (PCT) provided with MATLAB 2014a.

The feature extraction, graph construction and graph matching modules of the proposed system were tested on the GPU. It was observed that the feature computation and graph construction module were costly in terms of computation time. Therefore, the computationally expensive operations were implemented on the GPU, in order to take advantage of their parallel processing capabilities. In a typical image retrieval application, the retrieval time mainly depends upon the time required for feature computation of the query image, followed by the time required for matching the features of the query image with all other images in the dataset. In order to enable the framework to be adopted for online retrieval, it is necessary to reduce the feature matching time because this is the dominant computation when a query is performed by the CBIR system. In the proposed framework, the feature matching is computationally efficient and the GPU based implementation makes it even faster.

Modern GPUs are extremely powerful multi-core processors for graphics rendering in particular and usual data processing in general. It contains a large number of stream multiprocessors (SMs) with the capability of executing threads concurrently [49]. Each SM is equipped with many stream processors (SPs) and some low capacity, low latency shared memory. Global memory is shared by both CPU and GPU.

In order to enable execution on a GPU, the sequential procedure should first be decomposed into sequential and parallel code allowing it to be run on a CPU and GPU simultaneously. The PCT allows code to run directly from MATLAB on CUDA-enabled GPUs [50].

4.2 Datasets

Three famous datasets were used in the experiments, namely Corel dataset [45], Zurich-Buildings dataset (ZB) [46] and UKBench [47]. Corel is a large dataset containing several thousand images taken from the Corel Gallery Magic 2×10^4. Mainly two subsets are used in image retrieval performance evaluation scenarios i.e. Corel-5K dataset consisting of 5000 images organized into 50 categories of 100 images each. The second dataset is referred to as Corel-10K. It contains 10^4 images from 100 categories having 100 images in each category. The images are of similar sizes i.e. 192 × 128 or 128 × 192. We used Corel 10-K dataset for our experiments. ZB dataset consists of 1005 images of 201 buildings viewed from 5 different angles. The size of each image is 640 × 480. The buildings are randomly selected in Zurich, Switzerland. This dataset is chosen for evaluation because there is a mix of color and texture in its images. UKBench dataset consists of 6376 images in total with 4 images per object taken under different conditions. For each query, there are 4 similar images in the entire dataset. The objective is to retrieve the remaining three relevant images in the top retrieved images. The purpose of using this dataset is to evaluate the proposed algorithm for object based image retrieval.

4.3 Performance Evaluation Metrics

We used standard performance metrics widely used by the information retrieval community. These include: precision ($\mathcal{P}$), recall ($\mathcal{R}$) and mean average precision ($\mathcal{MAP}$) [51,52]. Precision measures the retrieval accuracy whereas recall measures the robustness of the retrieval algorithm. Higher average values for these metrics indicate better retrieval performance. These can be computed as:

$$\mathcal{P} = \frac{\text{Number of Relevant images retrieved}}{\text{Total number of images retrieved}} \tag{10}$$

$$\mathcal{R} = \frac{\text{Number of Relevant images retrieved}}{\text{Total Number of relevant images in the dataset}} \tag{11}$$

It is desired for the retrieval algorithm to be precise and be able to recall as much information as desired. Often, a precision-recall curve is shown to measure the performance of a retrieval algorithm. This curve provides an overall view of the accuracy and robustness of the approach in comparison with other approaches. Thus, the whole story of retrieval performance is depicted by its trend.

In addition to precision and recall, another widely used metric for evaluating the performance of CBIR systems is $\mathcal{MAP}$ which can be computed as:

$$\mathcal{MAP} = \frac{1}{\mathcal{Q}} \sum_{i=1}^{\mathcal{Q}} \mathcal{AP}_i \tag{12}$$

Where $\mathcal{Q}$ represents the number of queries run on the CBIR system, $\mathcal{AP}$ represents average precision, denoting the mean precision values of the relevant images and can be computed as:

$$\mathcal{AP} = \frac{1}{r} \sum_{i=1}^{r} \mathcal{P}_i \tag{13}$$

where $\mathcal{P}$i is the precision value for all the relevant images r. The $\mathcal{MAP}$ returns values in the range [0-1] with higher values representing good retrieval ranks and lower values indicating bad rankings.

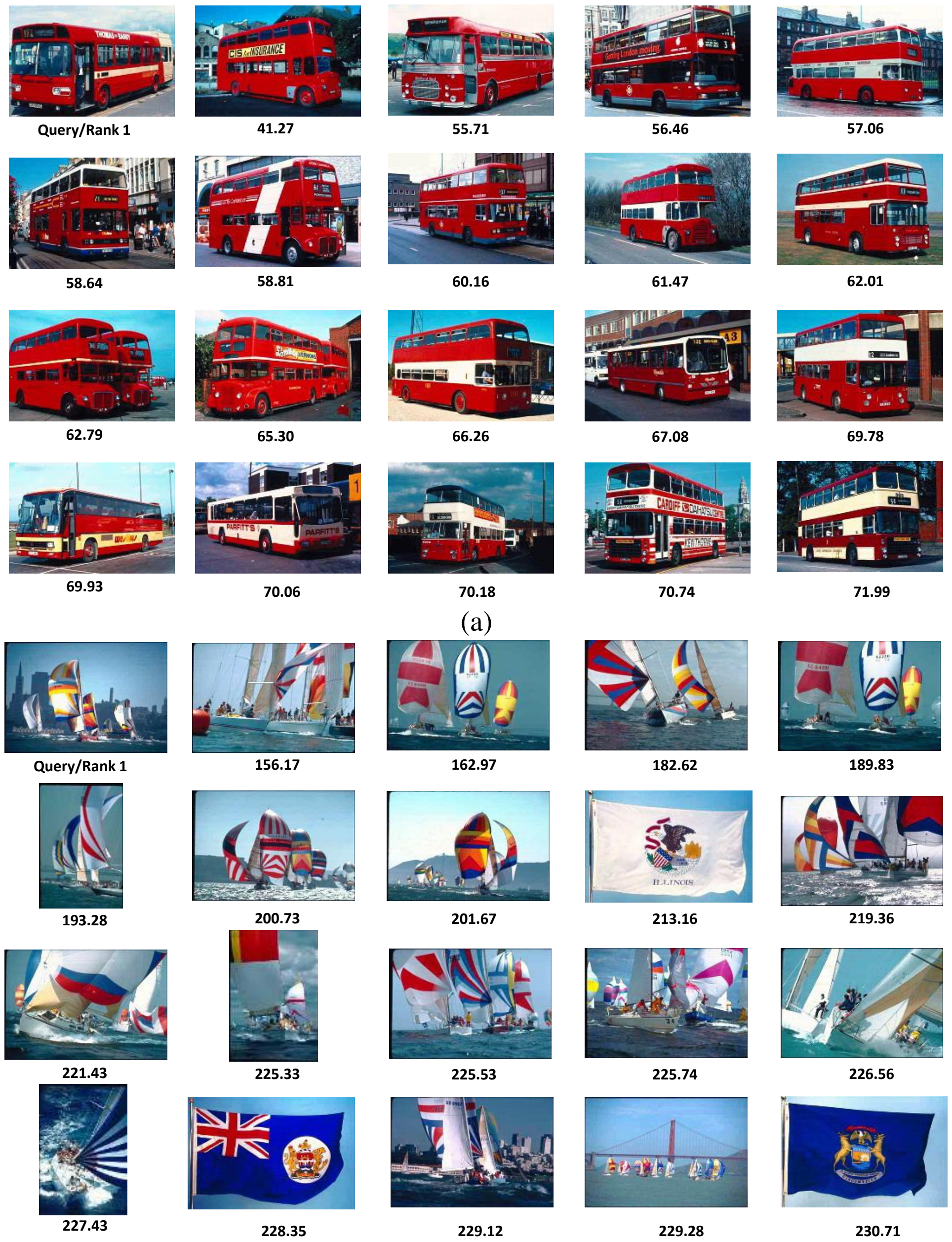
4.4 Retrieval Performance

Random queries were selected from within each class of the datasets to ensure a fair comparison. Performance of the proposed scheme evaluated on the datasets used is explained in the subsequent subsections.

4.4.1 Retrieval Performance on the Corel Dataset

Corel dataset is one of the most widely used datasets for evaluating the performance of retrieval algorithms. It is a challenging dataset because it contains a wide variety of images with rich color and texture contents. Visual results for some of the queries executed on the proposed system are shown in Figure 5. The first image in each of the result sets (a, b) is the query image. It can be seen from the visual results that relevant images are retrieved in the top ranks. And the irrelevant images that are retrieved in the lower ranks have visual similarity to the query image. For instance, images of flags are retrieved at ranks 9, 17 and 20 for a query of boat image (fig 5: b). The scores displayed below each image in figure 5 is the graph dissimilarity score. A lower score means a good match whereas a higher value represents a weak match. The images are arranged in increasing order of this score.

We evaluated the performance of our proposed scheme using a precision-recall curve and compared the results with other state-of-the-art techniques. Random queries were chosen from the entire dataset to retrieve top-n images where n was set at 20, 40, 60, 80 and 100 during the experiments. The retrieval results were very satisfactory with low recall. However, graceful degradation in the precision was noticed when recall was increased. Nevertheless, our results were much better than the other methods. We compared our results with MSD [30], ECDH [28], ART [28], FF [28], SED [32] and CTDD [26]. Our algorithm has shown performance improvements of 5.6% over CTDD, 13.87% for FF, 22% for ECDH, 24.4% for ART, 27.2% for SED and 31.6% for MSD. From the precision-recall curve, it can be seen that the retrieval performance was improved over the entire recall range. Hence, it is evident from these results that the proposed algorithm outperforms existing techniques by a significant margin.



(a)

(b)

Figure 5: Comparison of average precision-recall curve using different image retrieval methods for Corel-10K dataset

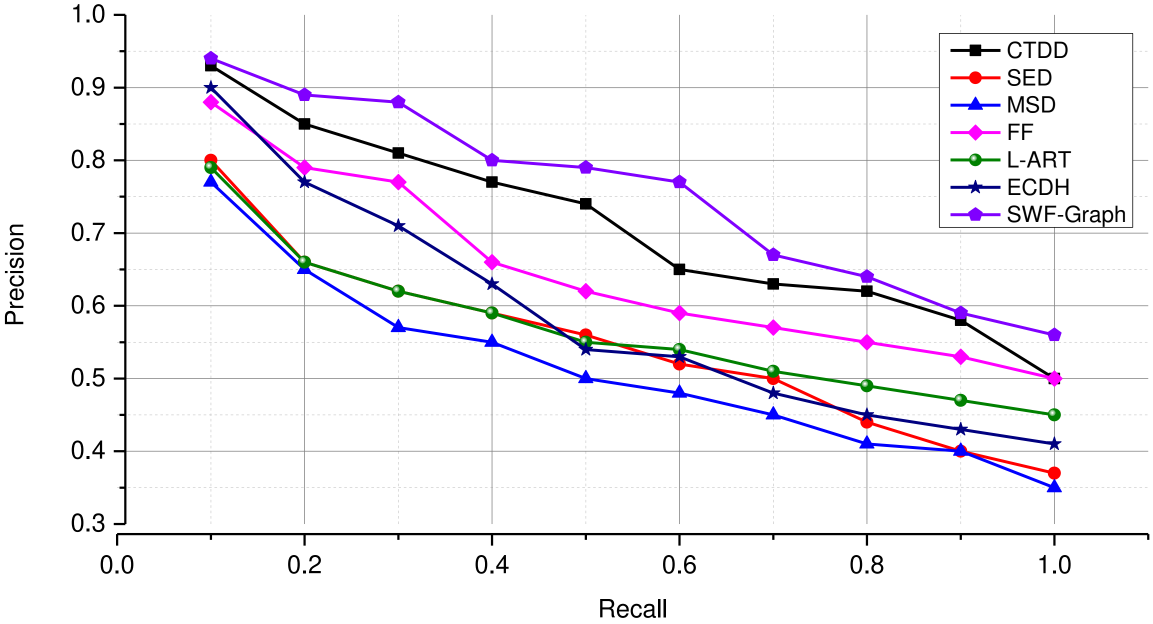


Figure 6: Precision-Recall curve comparison for Corel 10K dataset

4.4.2 Retrieval Performance on ZB Dataset

The performance of the proposed method with other state-of-the-art methods on this dataset is presented in Table 2. The values indicate the percentage $\mathcal{MAP}$ retrieval accuracy of all five relevant images over randomly selected query images. Our algorithm achieves an average performance gain of 15% on other similar approaches for recognizing buildings.

Table 2: Performance comparison of SFW-Graph with state-of-the-art methods on ZB dataset

Approach	Performance (MAP %)
MSD [30]	74.52
ECDH [28]	76.10
ART [28]	76.84
FF [28]	77.95
SED [32]	81.66
CTDD [26]	82.74
SFW-Graph	**90.10**

Some visual results are given in Figure 7(a) and Figure 7(b). The dissimilarity scores are shown below each image. The leftmost image is the query image. Since the images in this dataset have a very limited color palette with similar texture defined by the windows, walls, sky, clouds and trees, it becomes difficult for retrieval algorithms to retrieve relevant images in such scenarios. Representation techniques that possess significant discrimination power are able to perform well under such circumstances. Our graph representations are scale and orientation invariant, and hence images of the same buildings captured at various angles are retrieved successfully at top ranks. In certain queries, our algorithm failed to retrieve all the relevant images in the top-5 images. High illumination variations in some images caused a major performance hit in this dataset. However in those cases, most of the relevant images were found in the top-10 images.

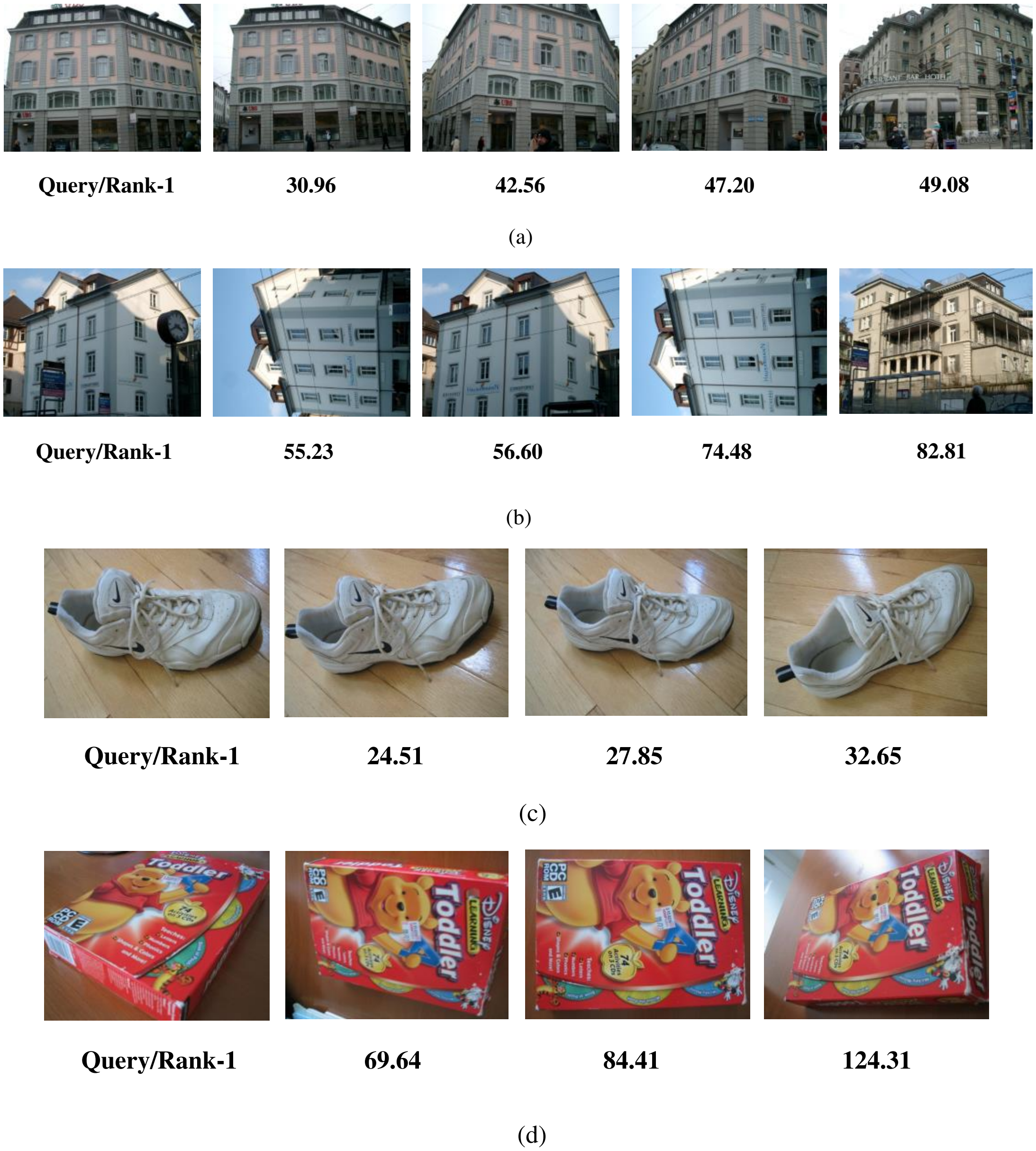


Figure 7: Sample retrieval results for two queries each over ZB (a,b) and UKBench (c,d)

4.4.3 Retrieval Performance on UKBench Dataset

The proposed method also performed well on the UKBench dataset. An average performance gain of 14% was achieved. The color palette in this dataset is far wider than the ZB dataset; hence, the proposed descriptor was able to achieve better performance in this large dataset. Since our graph representation takes into account both color and texture in the salient regions of the image, the effect of background gets reduced. Similarly, the slight illumination changes caused by changing the camera positions during image capture do not put any significant effect on its graph which shows the robustness of the proposed approach. Table 3 shows the performance comparison of the SFW-graph with other methods.

Table 3: Performance comparison of SFW-Graph with existing methods on UKBench dataset

Approach	Performance (MAP %)
MSD [30]	68.75
ECDH [28]	70.69
ART [28]	73.12
FF [28]	76.57
SED [32]	80.12
CTDD [26]	84.05
SFW-Graph	**86.25**

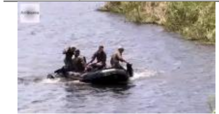
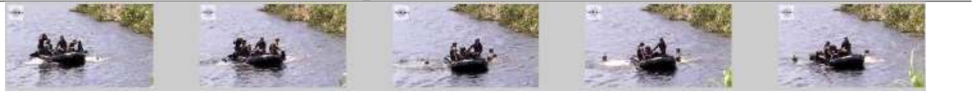
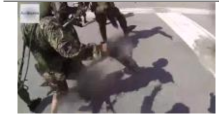
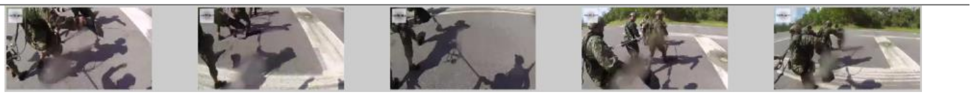
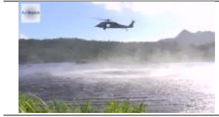
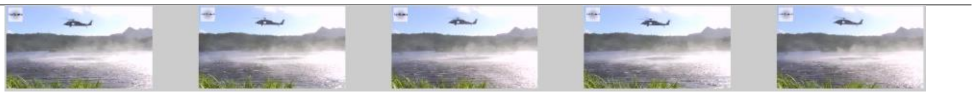
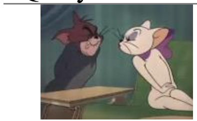
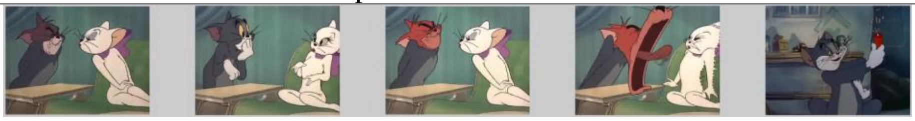
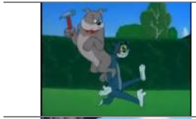
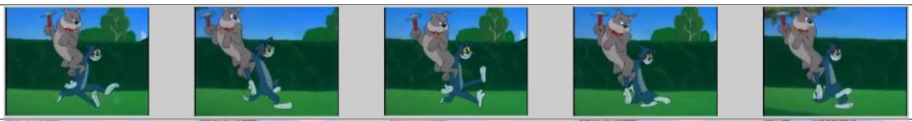
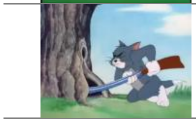
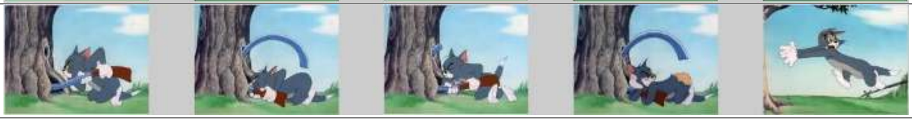
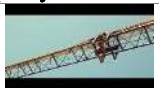
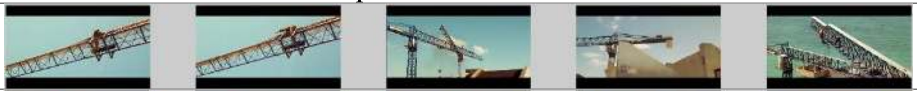
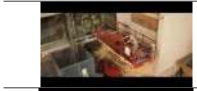
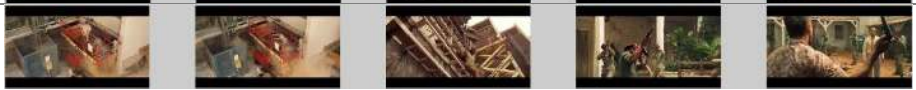
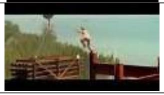
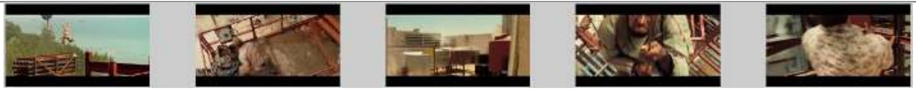
The retrieval results in Figure 7(c) and Figure 7(d) show slight variations of illumination and reflections in images, caused by the changing camera positions during image capture. Objects in the images also get affected by rotation, position, size and other geometric transformations including affine, making object based image retrieval more challenging. In spite of these challenges, most of the objects were successfully retrieved by our method. This success is attributed to the use of relative features when they were derived from the image during graph construction. It was noticed that, most of those queries where all the relevant images were not retrieved in the top-4 ranks, the false positives were mostly found in the lower ranks. The first image is the query image, and the remaining images are the top retrieved images with score displayed on top of each image. It can be seen that images are accurately ranked on the basis of the degree of similarity that they have with the query image. Images with greater visual similarity have scores closer to each other and different than the rest.

4.5 Retrieving Similar Frames from Video Sequences

In order to evaluate the pair-wise image matching performance of the proposed descriptor in locating the position of particular frames in videos, a case study is being presented. The purpose of this case study is to provide an insight into the representative strength of the proposed ARG by utilizing it as a video searching and browsing tool. For evaluation, several videos were acquired from YouTube including military documentary clips, short movie clips, cartoons, and music videos. Each of these videos was at least 4-5 minutes long consisting of 9000 frames on average. Random frames were selected as queries from each video and the proposed method was used to retrieve similar looking frames from those videos. Since video contains highly redundant visual data, certain frames were skipped while looking for similar frames. Hence in most of the cases, the query frame was skipped during the search. For all these experiments, 20 frames were skipped to cope with redundancy issue during the retrieval process. Some details about videos used along with the visual results of top-5 retrieved frames are given in Table 4.

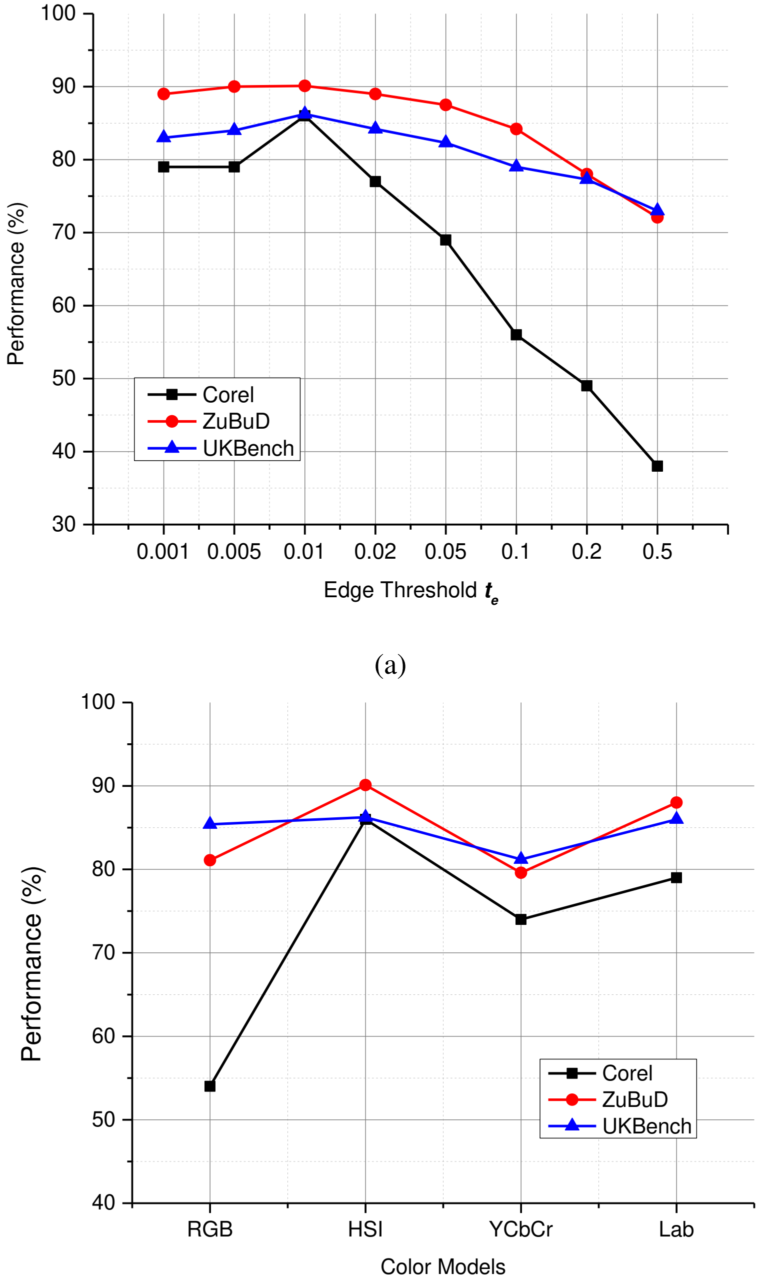
The frame skip setting allows our system to compare the query frame with just a single frame per second. Therefore, the desirable output is just to get to the right frame in the video. The results revealed that more than 95% of the time, our proposed method was able to retrieve the most relevant images in the top most frames. The failure in some cases was caused by lesser number of similar frames in the video and the larger frame-skip. However, we were able to retrieve those frames successfully when we reduced the frame skip to 12. These experiments reveal that the proposed method can be effectively applied to this application. We also hope that this technique can be deployed with a higher degree of success to cluster visually similar frames inside video sequences as well as extract keyframes later on. However this study is out of the scope of the present discussion and is therefore left for a future study. Nevertheless, the results of this case study supported our claim that the proposed method is a powerful representation model for visual data that effectively preserves its richness. We strongly hope that our proposed framework will allow successful pairwise image matching efficiently wherever applied. Table 5 lists the quantitative results of the experiments conducted on 30 YouTube videos.

Table 4: Retrieval results of frames from video sequences

Video Title	**U.S. Marines Maritime Raid Force - MH-60S Helicopter Casting and SPIE**
Description	A training video of the U.S. marines and sailors (Duration = 4:52) No of Frames = 8789, Frames checked during retrieval = 600
Query Frame	Top-5 retrieved frames
	
	
	
Video Title	**Tom and Jerry - Funny Best Moment's**
Description	A short video clip compiled from the best funny moments (Duration = 6:12) No of Frames = 11173, Frames checked during retrieval = 750
Query Frame	Top-5 retrieved frames
	
	
	
Video Title	**Casino Royale Movie CLIP - Parkour Chase (2006) HD**
Description	A short movie clip where James Bond is chasing a person (Duration = 7:51) No of Frames = 14134, Frames checked during retrieval = 950
Query Frame	Top-5 retrieved frames
	
	
	

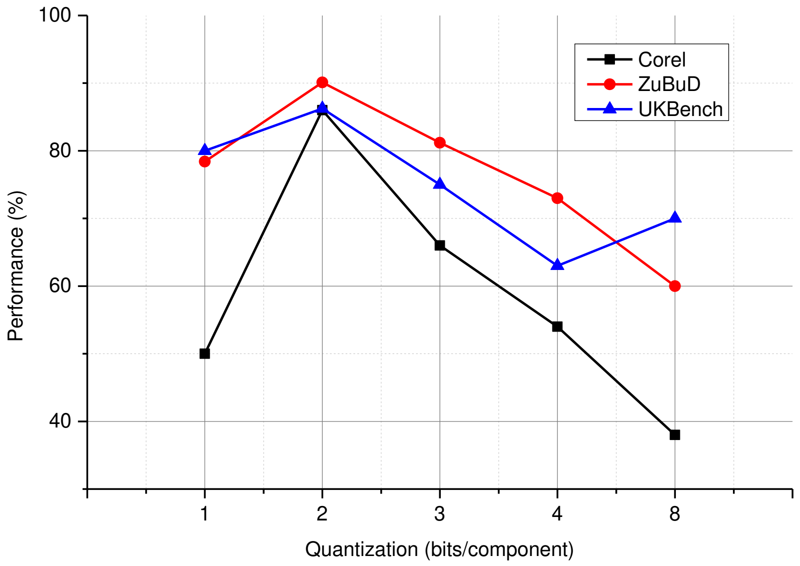
4.6 Effect of Edge Threshold, Color Models and Quantization on Retrieval Performance

There are several parameters that govern the performance of our graph based image retrieval approach. The main parameters include the edge threshold τ_e which defines the graph density, the choice of color models and the amount of quantization. All these parameters are evaluated to test the performance of the proposed approach with different configurations on all three datasets. Figure 8(a) shows the performance of all datasets on varying values of τ_e. Setting high values for τ_e misses out most of the information and models only the longer edges in the ARG. It was observed that less dense graphs resulting with higher τ_e values fail to sufficiently model the texture defined by edges, and hence this significant drop is noticed in performance. The performance hit on the Corel dataset is the most intense due to the fact that this dataset consists of images with very rich colors and textures. Additionally, images belonging to the same class vary greatly in colors and textures. For the value 0.01 our scheme performs the best across all datasets. Lower values cause very small and potentially insignificant edges in images to be modeled in the ARG. Therefore, performance drops slightly.



(a)

(b)



(c)

Figure 8: (a) Effect of threshold value τ_e on performance (b) Performance with various color models and (c) Performance comparison with varying quantization settings

Table 5: Quantitative evaluation of frame retrieval in video sequences

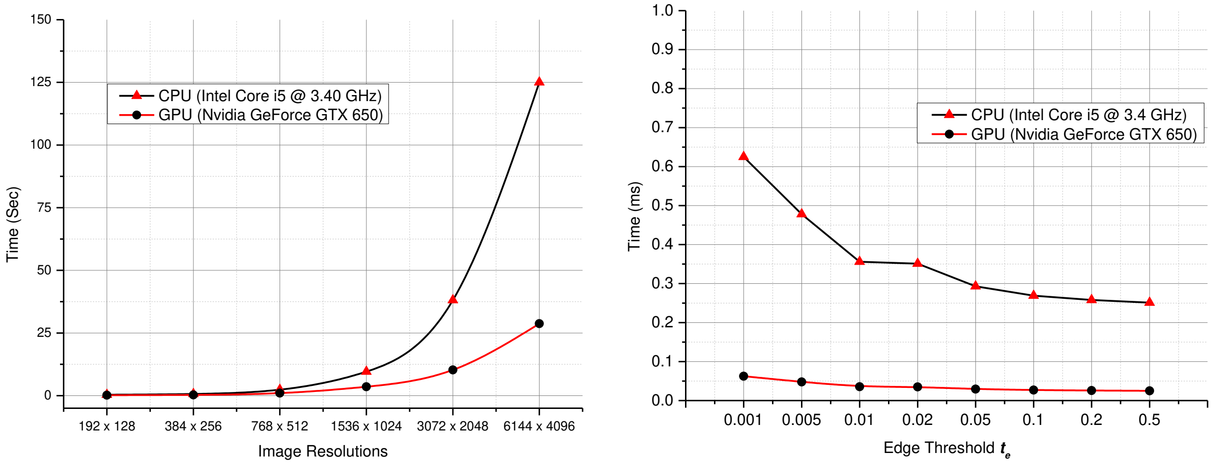
Video #	Total Frames	Frames Checked	Precision	Recall
1	8789	595	0.800	0.667
2	9504	638	0.900	0.818
3	15408	1027	0.700	0.875
4	11173	748	0.800	0.571
5	5880	394	0.700	0.636
6	11190	746	0.800	0.889
7	15360	1024	0.900	1.000
8	11400	760	0.800	0.615
9	4620	308	0.700	0.636
10	14134	936	0.800	0.615
Average of 30 videos	**9150**	**711.4**	**0.846**	**0.762**

To determine the most suitable color model, various experiments were performed. Results reported in Figure 8(b) presents HSI as the model of choice for our scheme. It performed significantly well as compared to other color models, achieving almost 15% better results. Again the choice of color models affects performance on the Corel dataset the most because of the same reason we discussed earlier. The performance with LAB color model was also satisfactory, however it carried with it a great computational burden so it was dropped.

The effect of quantization on retrieval performance from the conducted experiments is shown in the graph in Figure 8 (c). Various combinations of bits per component were attempted during the experiments. A balance between the number of representative colors and graph density was required for optimal storage and retrieval performance. These experiments were conducted keeping optimal values for other parameters. The performance hit on the datasets for low quantization indicates that a significant number of representative colors cannot be modeled with low quantization. Similarly, high quantization reduces performance by introducing less important colors into the quantized image and breaking significant and large color regions into many smaller color chunks. Additionally, it causes computation overhead. To achieve a balance between these two, it is suggested from the experiments that 2 bits per color component is the optimum.

4.7 Computation Time Analysis

The proposed scheme was implemented on two different hardware platforms i.e. CPU and GPU. Experiments were conducted on both platforms to observe their suitability for various image retrieval tasks. It was found that the graph construction module is the computationally expensive one. But since this operation is mainly carried out during the offline phase of image retrieval, it has little significance on the retrieval performance of the system. On the other hand, the graph matching module takes less time as compared to graph construction but this procedure is mostly carried out at the online retrieval stage. Hence, the graph matching module should be efficient for it to be adopted for real-time image retrieval applications. Figure 9 (a) shows the time comparison of graph computation module for different size images on a CPU and GPU. It can be seen that the GPU based implementation is almost 6x faster as compared to CPU based implementation. In Figure 9 (b), the time taken by the CPU and GPU for graph matching are shown for varying values of τ_e. The image size considered is 1536 × 1024. The reason for choosing a large size is to allow us witness the significant change that is usually noticed in such comparisons. It was observed that, for lower values of τ_e, relatively denser graphs are generated and hence more time is required for their matching. Reducing τ_e causes significant drop in the time requirement for graph matching. At $\tau_e = 0.01$, the GPU based implementation is 18x faster than CPU based implementation, requiring 35.6 sec for retrieving images from a database of 1 Million full HD images. For low resolution images, the proposed method takes on average 6 seconds to make 1M comparisons. These results verify that the GPU based implementation considerably improves computation times making it suitable for online image retrieval systems.



(a) (b)

Figure 9: Time comparison of (a) graph construction and (b) graph matching (image size = 1536 x 1024)

5. Conclusion

We presented a graph based image representation model with an efficient matching algorithm for image retrieval systems. Salient color and edge features of the image were represented as ARGs. Low-level color features were extracted from quantized images in the HSI color space whereas salient edge features were extracted from the salient edge map of the image. The reason behind the inclusion of saliency map was to achieve a representation closer to the human perception model. The extracted features were then modeled as nodes and edges in the SFW graph. These graphs were then stored as sparse arrays in a database. The

SFW graph of the query image was compared with graphs of other images in the dataset through a lightweight graph matching procedure for the retrieval of relevant images. With our segmentation and clustering-free framework, we were able to achieve discriminative and geometric transformation invariant content representation. The experimental results suggest that our proposed approach is both effective and efficient, achieving better results than other state-of-the-art methods. The GPU based implementation and the efficient graph matching procedure makes it suitable for online image retrieval systems as well. In the future, we will try to incorporate shape features, illumination invariant texture representations and some local graph structural features into our approach. It will further increase the efficiency and performance of our method.

Acknowledgements

This research is supported by the Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (2013R1A1A2012904).

References

1 Smeulders, A.W., Worring, M., Santini, S., Gupta, A., Jain, R.: Content-based image retrieval at the end of the early years. Pattern Analysis and Machine Intelligence, IEEE Transactions on **22**(12), 1349-1380 (2000).

2 Ji, R., Duan, L.-Y., Chen, J., Yao, H., Huang, T., Gao, W.: Learning compact visual descriptor for low bit rate mobile landmark search. In: IJCAI Proceedings-International Joint Conference on Artificial Intelligence2011, vol. 22, p. 2456

3 Yu, Y.-H., Lee, T.-T., Chen, P.-Y., Kwok, N.: On-chip real-time feature extraction using semantic annotations for object recognition. Journal of Real-Time Image Processing, 1-16 (2014).

4 Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91-110 (2004).

5 Bay, H., Tuytelaars, T., Van Gool, L.: Surf: Speeded up robust features. In: Computer vision–ECCV 2006. pp. 404-417. Springer, (2006)

6 Calonder, M., Lepetit, V., Strecha, C., Fua, P.: Brief: Binary robust independent elementary features. In: Computer Vision–ECCV 2010. pp. 778-792. Springer, (2010)

7 Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: Computer Vision, 2003. Proceedings. Ninth IEEE International Conference on2003, pp. 1470-1477. IEEE

8 Sural, S., Qian, G., Pramanik, S.: Segmentation and histogram generation using the HSV color space for image retrieval. In: Image Processing. 2002. Proceedings. 2002 International Conference on2002, vol. 2, pp. II-589-II-592 vol. 582. IEEE

9 Han, J., Ma, K.-K.: Fuzzy color histogram and its use in color image retrieval. Image Processing, IEEE Transactions on **11**(8), 944-952 (2002).

10 Hafner, J., Sawhney, H.S., Equitz, W., Flickner, M., Niblack, W.: Efficient color histogram indexing for quadratic form distance functions. Pattern Analysis and Machine Intelligence, IEEE Transactions on **17**(7), 729-736 (1995).

11 Huang, J., Kumar, S.R., Mitra, M., Zhu, W.-J., Zabih, R.: Image indexing using color correlograms. In: Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on1997, pp. 762-768. IEEE

12 Manjunath, B.S., Ohm, J.-R., Vasudevan, V.V., Yamada, A.: Color and texture descriptors. Circuits and Systems for Video Technology, IEEE Transactions on **11**(6), 703-715 (2001).

13 Wu, P., Manjunath, B., Newsam, S., Shin, H.: A texture descriptor for browsing and similarity retrieval. Signal Processing: Image Communication **16**(1), 33-43 (2000).

14 Won, C.S., Park, D.K., Park, S.-J.: Efficient use of MPEG-7 edge histogram descriptor. Etri Journal **24**(1), 23-30 (2002).

15 Ro, Y.M., Kim, M., Kang, H.K., Manjunath, B., Kim, J.: MPEG-7 homogeneous texture descriptor. Etri Journal **23**(2), 41-51 (2001).

16 Guo, Z., Zhang, D.: A completed modeling of local binary pattern operator for texture classification. Image Processing, IEEE Transactions on **19**(6), 1657-1663 (2010).

17 Fu, X., Wei, W.: Centralized binary patterns embedded with image Euclidean distance for facial expression recognition. In: Natural Computation, 2008. ICNC'08. Fourth International Conference on2008, vol. 4, pp. 115-119. IEEE

18 Zhang, B., Gao, Y., Zhao, S., Liu, J.: Local derivative pattern versus local binary pattern: face recognition with high-order local pattern descriptor. Image Processing, IEEE Transactions on **19**(2), 533-544 (2010).

19 Liao, W.-H.: Region Description Using Extended Local Ternary Patterns. In: ICPR2010, pp. 1003-1006

20 Vipparthi, S.K., Nagar, S.K.: Color directional local quinary patterns for content based indexing and retrieval. Human-Centric Computing and Information Sciences **4**(1), 1-13 (2014).

21 Belongie, S., Malik, J., Puzicha, J.: Shape matching and object recognition using shape contexts. Pattern Analysis and Machine Intelligence, IEEE Transactions on **24**(4), 509-522 (2002).

22 Ahmad, J., Jan, Z., Khan, S.M.: A Fusion of Labeled-Grid Shape Descriptors with Weighted Ranking Algorithm for Shapes Recognition. World Applied Sciences Journal **31**(6) (2014).

23 Qureshi, R.J., Ramel, J.-Y., Cardot, H.: Graph based shapes representation and recognition. In: Graph-Based Representations in Pattern Recognition. pp. 49-60. Springer, (2007)

24 Liu, G.-H., Yang, J.-Y.: Content-based image retrieval using color difference histogram. Pattern Recognition **46**(1), 188-198 (2013).

25 Chatzichristofis, S.A., Boutalis, Y.S.: CEDD: color and edge directivity descriptor: a compact descriptor for image indexing and retrieval. In: Computer Vision Systems. pp. 312-322. Springer, (2008)

26 Rahimi, M., Moghaddam, M.E.: A content-based image retrieval system based on Color Ton Distribution descriptors. Signal, Image and Video Processing, 1-14 (2013).

27 Liu, G.-H., Zhang, L., Hou, Y.-K., Li, Z.-Y., Yang, J.-Y.: Image retrieval based on multi-texton histogram. Pattern Recognition **43**(7), 2380-2389 (2010).

28 Walia, E., Pal, A.: Fusion framework for effective color image retrieval. Journal of Visual Communication and Image Representation **25**(6), 1335-1348 (2014).

29 Xia, C., Qi, F., Shi, G., Wang, P.: Nonlocal center–surround reconstruction-based bottom-up saliency estimation. Pattern Recognition **48**(4), 1337-1348 (2015).

30 Liu, G.-H., Li, Z.-Y., Zhang, L., Xu, Y.: Image retrieval based on micro-structure descriptor. Pattern Recognition **44**(9), 2123-2133 (2011).

31 Ortega, M., Rui, Y., Chakrabarti, K., Porkaew, K., Mehrotra, S., Huang, T.S.: Supporting ranked boolean similarity queries in MARS. Knowledge and Data Engineering, IEEE Transactions on **10**(6), 905-925 (1998).

32 Wang, X., Wang, Z.: A novel method for image retrieval based on structure elements' descriptor. Journal of Visual Communication and Image Representation **24**(1), 63-74 (2013).

33 Schwartz, S.H., Meese, T.: Visual perception: A clinical orientation. McGraw-Hill Medical Pub. Division, (2010)

34 Pashler, H.E., Sutherland, S.: The psychology of attention, vol. 15. MIT press Cambridge, MA, (1998)

35 Huang, S., Wang, W., Zhang, H.: Retrieving images using saliency detection and graph matching. In: Image Processing (ICIP), 2014 IEEE International Conference on2014, pp. 3087-3091. IEEE

36 Eimer, M.: The neural basis of attentional control in visual search. Trends in cognitive sciences **18**(10), 526-535 (2014).

37 Kastner, S., Ungerleider, L.G.: The neural basis of biased competition in human visual cortex. Neuropsychologia **39**(12), 1263-1276 (2001).

38 Livingstone, M.S., Hubel, D.H.: Anatomy and physiology of a color system in the primate visual cortex. J Neurosci **4**(1), 309-356 (1984).

39 Gonzalez, R.C., Woods, R.E.: Digital image processing. In. Prentice Hall Upper Saddle River, NJ, (2002)

40 Burger, W., Burge, M.J., Burge, M.J., Burge, M.J.: Principles of Digital Image Processing. Springer, (2009)

41 Jian, M., Lam, K.-M., Dong, J., Shen, L.: Visual-Patch-Attention-Aware Saliency Detection. (2014).

42 Rahtu, E., Kannala, J., Salo, M., Heikkilä, J.: Segmenting salient objects from images and videos. In: Computer Vision–ECCV 2010. pp. 366-379. Springer, (2010)

43 Hou, X., Zhang, L.: Saliency detection: A spectral residual approach. In: Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on2007, pp. 1-8. IEEE

44 Achanta, R., Süsstrunk, S.: Saliency detection using maximum symmetric surround. In: Image Processing (ICIP), 2010 17th IEEE International Conference on2010, pp. 2653-2656. IEEE

45 Tang, J., Lewis, P.H.: A study of quality issues for image auto-annotation with the corel dataset. Circuits and Systems for Video Technology, IEEE Transactions on **17**(3), 384-389 (2007).

46 Shao, H., Svoboda, T., Van Gool, L.: Zubud-zurich buildings database for image based recognition. Computer Vision Lab, Swiss Federal Institute of Technology, Switzerland, Tech. Rep **260** (2003).

47 Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on2006, vol. 2, pp. 2161-2168. IEEE

48 NVIDIA: GeForce GTX 650. http://www.geforce.com/hardware/desktop-gpus/geforce-gtx-650 (2015).

49 Havel, J., Dubská, M., Herout, A., Jošth, R.: Real-time detection of lines using parallel coordinates and CUDA. Journal of real-time image processing **9**(1), 205-216 (2014).

50 Features - Parallel Computing Toolbox. http://www.mathworks.com/products/parallel-computing/features.html?refresh=true (2015). Accessed 27-July-2015

51 Müller, H., Müller, W., Squire, D.M., Marchand-Maillet, S., Pun, T.: Performance evaluation in content-based image retrieval: overview and proposals. Pattern Recognition Letters **22**(5), 593-601 (2001).

52 Yang, Y.: An evaluation of statistical approaches to text categorization. Information retrieval **1**(1-2), 69-90 (1999).