Journal of Theoretical and Applied Information Technology
15th October 2019. Vol.97. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

GLOBAL DOMINANT SIFT FOR VIDEO INDEXING AND RETRIEVAL

KAMAL ELDAHSHAN[1], HESHAM FAROUK[2], AMR ABOZEID[3], M. HAMZA. EISSA[4]

[1,3,4] Dept. of Mathematics, Computer Science Division,

Faculty of Science, Al-Azhar University, Cairo, Egypt.

[2] Computers and Systems Dept., Electronics Research Institute, Cairo, Egypt.

E-mail: [1]dahshan@gmail.com, [2]hesham@eri.sci.eg,
[3]aabozeid@azhar.edu.eg, [4]mohammed_essa2001@yahoo.com

ABSTRACT

The massive volume of videos is highly demanding for produce an efficient and effective video indexing and retrieving frameworks. Extracting and representation of visual features plays a significant role in the video/image retrieval and computer vision. This paper proposes a new compact descriptor named Global Dominant Scale Invariant Feature Transform (GD-SIFT). The GD-SIFT requires fewer bits (16 bits) to represent each visual feature. Importantly, the proposed descriptor is vocabulary-free, training-free and suitable for online and real-time applications. Also, this paper proposes a new video indexing and retrieving framework based on the proposed GD-SIFT descriptor. The proposed framework is a content-based video indexing and retrieving, which helps to retrieve videos by text (e.g. Video name or metadata), image (video frame) or video clip. The experiments carried out on the standard Stanford I2V dataset. Our experiments demonstrated that, the GD-SIFT descriptor is more efficient (in terms of speed and storage) and achieved high accuracy (about 78%) with respect to the related works. Moreover, the results indicated that, the proposed descriptor is more robust to variations (e.g. Scale, rotation, etc.).

Keywords: *Video Indexing, Video Search, SIFT, Descriptor, Query-By-Image*

1. INTRODUCTION

The availability of communications, video recording devices and low cost storage technologies, allows a user to record and then create huge video databases. There is an increase demand for an efficient and effective indexing and retrieving framework to maintain the huge video databases. Indexing and retrieving videos from a huge video database is a challenging task [1, 2]. As result, many content-based video indexing and retrieval (CBVR) frameworks have been proposed in the literature.

CBVR can defined as "the automatic process of content-based classification of video data for fast access and retrieval" [3]. This definition mean that, extract information from the video content to perform specified queries [4]. Content-based refers to the actual video contents which might be local visual features (like colors, texture, motion or objects) and audio features [5].

Video is complex and contains a large amount of visual features. However, these features should be extracted, analyzed and stored in an efficient way. During the last decades, the number of different visual features which proposed in literature has increased significantly [6]. Many local features were proposed to be faster, more distinctive and robust under many different variations (e.g. scale, rotation, etc.) [7] .

Some popular and successful local features developed during the recent decade are Scale Invariant Feature Transform (SIFT) [8], Principal Component Analysis (PCA)-SIFT [9], Speeded Up Robust Features (SURF) [10] and Histogram of Oriented Gradients (HOG) [11]. Traditional local features have limitations in ubiquitous and real-time applications because of their large size (e.g. 128 bytes for a SIFT key-point) [12, 13].

Recently, binary features, such as Binary Robust Independent Elementary Features (BRIEF)[14], Binary Robust Invariant Scalable

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Key-points (BRISK) [7] and Fast Retina Key-point (FREAK) [15], are proposed to represent the local feature in a more distinctive way. However, these features are still large in size (e.g. ≥ 16 bytes per each key-point) while some low bit-rate image/video retrieval applications aim to be much smaller (e.g. ≤100 bits per feature) [16, 17].

The author's in [18] presented a compact SIFT descriptor (called dominant SIFT) which only uses 48 bits to describe the local features (each key-point) of the image. The main advantages of this descriptor are training-free, vocabulary-free and suitable for ubiquities and real-time applications.

This paper extends of the dominant SIFT and proposes a GD-SIFT for video features. The GD-SIFT more compact than the dominant SIFT and uses 16 bits (instead of 128 bits for SIFT [8] and 48 bits for dominant SIFT [19]) to describe each key-point. Also, we propose a framework for video indexing and retrieval based on the GD-SIFT and the time-constraint cluster algorithm. The proposed framework is a web application which helps user to upload and search for videos. The user can retrieve videos by text (e.g. video name or metadata), image (video frame) or video clip. The experiment results shows that, the GD-SIFT is more efficient (in terms of speed and storage) and achieved high accuracy (an average of 78%) with respect to the original SIFT [8] and the dominant SIFT [19] descriptors. Moreover, the proposed descriptor more robust to variations (e.g. Scale, rotation, etc.). Importantly, the proposed descriptor is vocabulary-free and training-free. Therefore, the proposed GD-SIFT is suitable for online and real-time applications.

The paper is organized as: section 2 introduces a related work. Section 3 explains the proposed GD-SIFT methodology. Section 4 discusses the experimental results. Finally, section 5 concludes the paper and suggests future work.

2. RELATED WORK

The SIFT feature includes two main parts: key-point detector and SIFT descriptor [8]. The key-point detector scans the input image to detect the interest points. Firstly, Gaussian filter of different scales is applied on the input image and then re-sized to produce a Gaussian scale-space. Neighboring images with the same resolution in this scale-space are subtracted to get the Difference of Gaussian (DoG) pyramid. The key-point is taken if and only if it is a local extremum in the DoG pyramid. The key-point localization is the last step applied to get the most stable key-points.

The standard key-point descriptor used by SIFT is created by sampling the magnitudes and orientations of the image gradient in the patch around the key-point, and building smoothed orientation histograms to capture the important aspects of the patch. A 4×4 array of histograms, each with 8 orientation bins, captures the rough spatial structure of the patch. This 128-element vector is then normalized to unit length and thresholded to remove elements with small values.

The SIFT key-points are particularly useful due to their distinctiveness, which enables the correct match for a key-point to be selected from a large database of other key-points. This distinctiveness is achieved by assembling a high-dimensional vector representing the image gradients within a local region of the image. The key-points have been shown to be invariant to image rotation and scale and robust across a substantial range of affine distortion, addition of noise, and change in illumination. Large numbers of key-points can be extracted from typical images, which leads to robustness in extracting small objects among clutter.

The fact that key-points are detected over a complete range of scales means that small local features are available for matching small and highly occluded objects, while large key-points perform well for images subject to noise and blur. Their computation is efficient, so that several thousand key-points can be extracted from a typical image with near real-time performance on standard PC hardware.

Ke et al. [9] introduced an alternate representation for local image descriptors for the SIFT algorithm. Compared to the standard representation, PCA-SIFT is both more distinctive and more compact leading to significant improvements in matching accuracy (and speed) for both controlled and real-world conditions. Each 41-pixel-by-41-pixel image patch centering at each key-point is extracted and rotated to line up with its dominant orientation. Gradient values in the x-direction and the y-direction for all pixels in the image patch are calculated to form a 2x39x39 = 3042-dimension vector.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Despite its name, PCA-SIFT does not reduce the SIFT feature vector, but the dimensionality of the detected interest points. Furthermore, each 3042 feature vector is projected onto a low-dimensional space. In order to execute this last task, a projection kernel is pre-computed using PCA over 21000 patches collected from diverse images that are not used later. This new less-dimensional feature vector speeds up applications using it, however, it may lead to less accurate results than those obtained by using SIFT descriptors. PCA-SIFT is demonstrated to achieve better results when it reduces its descriptor to a 36-dimensional feature vector.

The dimension of SIFT vector can be directly reduced by using PCA transform. Similarly to PCA-SIFT, a PCA transform matrix is pre-learned from an image database. At mobile devices, SIFT features extracted from query images are applied with PCA transform to achieve a more compact descriptor. This new compact descriptor is called as Reduced SIFT [20].

SIFT uses only grayscale information to detect key-points. Therefore, a lot of color information is discarded for color images. Alaa et al. [21] proposed a color SIFT (CSIFT), which combines color variance with the basis of SIFT and intends to beat the flaw of SIFT for color images.

Table 1: Comparisons Between SIFT and Its Variants.

	Key-point Detection		Key-point Description		
	Scale space	**Selection**	**Main direction Feature**	**Extraction**	**Size (bits)**
SIFT [8]	Multi-scale images convoluted by a Gaussian function	Detect extrema in Difference of Gaussian space (DoG)	• Compute a gradient amplitude of a square area (16×16). • Select the direction with the maximum gradient amplitude as the main direction	• Divide a 16×16 region into 4×4 sub-regions; • Compute a gradient histogram for each sub-region	128
PCA-SIFT [9]	Similar to SIFT	Similar to SIFT	Similar to SIFT	• Extract a 41×41 patch. • Construct a 3042-dimensional vector. • Use a project matrix to reduce the dimensionality	<=20
CSIFT [21]	• Combine grayscale information with color information. • Convolute by a Gaussian function	Similar to SIFT	Similar to SIFT	Similar to SIFT	384
Dominant SIFT [19]	Similar to SIFT	Similar to SIFT	Similar to SIFT	• Divide a 16×16 region into 4×4 sub-regions; • Compute a gradient histogram for each sub-region • Compute a dominate gradient histogram	48
GD-SIFT (Our proposed)	Similar to SIFT	Similar to SIFT	Similar to SIFT	• Divide a 16×16 region into 4×4 sub-regions; • Compute a gradient histogram for each sub-region • Encode the global dominate gradient histogram using time-cluster algorithm	**16**

To achieve a more compact descriptor, hashing, vector quantization (VQ) and transform coding (TC) are also considered [22]. Hashing is an effective way to represent the local feature by using a few bits [16], but it depends a lot on its hash functions. VQ technique represents each local

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

feature by a code-word of a pre-trained vocabulary [23], but the large size of vocabulary becomes a problem for devices having small memory [17]. TC framework maps the local feature from original feature space into the transform space using PCA technique which produces a small reconstruction error when reducing feature dimensions [17].

Tra et al. [18] presented a compact SIFT descriptor (called dominant SIFT) which only uses 48 bits to describe the local features (each key-point) of the image. The main advantages of this descriptor are training-free, vocabulary-free and suitable for ubiquities and real-time applications.

The SIFT and its variants methodology consists of two main steps: key-point detection and description. Based on this methodology, Table 1 summarizes the comparisons between SIFT and its variants including the proposed GD-SIFT.

3. METHODOLOGY

Given a video as a new input to the database, the video key-frames descriptors are extracted and they are stored in the system database. Once the system receives a query image, the similarity between the query image descriptor and the descriptors already stored in the database is measured by the Brute-Force matcher [24]. The resulted videos are ranked based on the measured similarity between the input descriptor and the already stored descriptors. The global overview of the proposed framework is shown in Fig. 1. The whole process is divided into two main stages: index stage (offline stage) and retrieval stage (online stage), all of them are explained as follows.

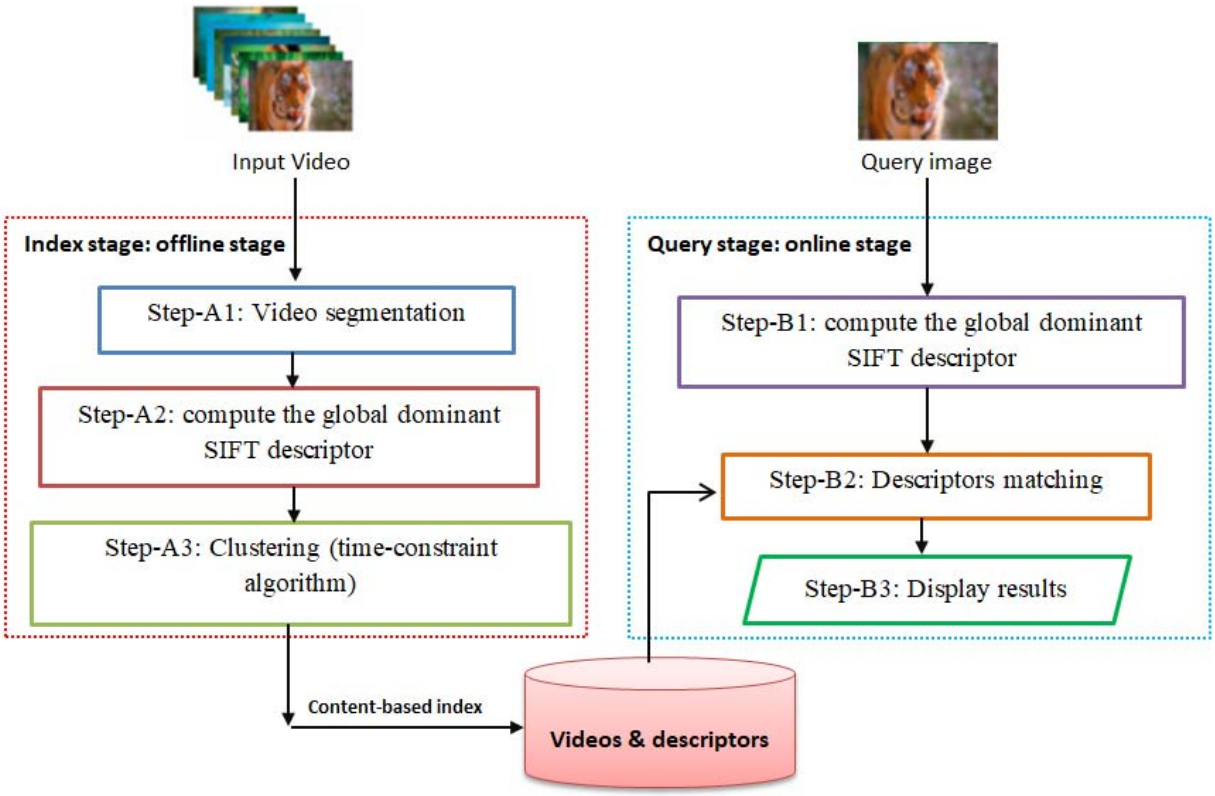


Figure 1: The Proposed Framework Architecture

3.1 Index Stage: An Offline Stage

The goal of this stage is to analysis the input video in order to get a better compact descriptor of the entire video frames. The index stage consists of three main steps; all of them are explained as follows.

Step-A1: Video segmentation

The input video is considered as a collection of representative key-frames which will

be processed to extract feature descriptors that represent its content. Since the computational cost is proportional to the amount of data (video frames) being processed, two steps are performed to reduce the quantity of data in both temporal and spatial domains: key-frame pre-sampling and resizing.

The objective of pre-sampling and resizing are reduce the computational complexity. The key-frame sampling method is depend on there are a visual redundancy between the consecutive frames in each second. Subsequently, instead of process the entire video frames a subset frame are processed based on a predefined sampling rate. The sampling rate can be defined by the second or by the frame number [25]. In this step, we select one frame per second to be the key-frame. Then, each selected key-frame is re-scaled to CIF (352 x 240) resolution.

Step-A2: GD-(GD) SIFT descriptor

The SIFT algorithm constructs a description for each key-point based on a 4×4 patch of pixels around the key-point. The final SIFT descriptors are constructed from 16 sub-histograms corresponding to 4×4 patch of pixels. In each patch sixteen gradients are quantized into 8 bins of the sub-histogram. Based on the statistical experiment in [19], there are a stronger correlation between bins in the same sub-histogram correlation than bins in different sub-histograms of a SIFT descriptor. Moreover, the sub-histogram values often concentrate on two or three adjacent bins after a circular shift.

For each SIFT vector: $(a^j)_{j\epsilon z\cap[0,15]}$

$where\ a^j = \{a_i^j \epsilon z \cap [0,256] | i \in z \cap [0,7]\}$

is a 8-bin sub-histogram. Suppose that $CS_n(a^j, i)$ be the consecutive *sum-n* at the index i which is defined as:

$$CS_n(a^j, i) = \sum_{k=i}^{i+n-1} a_k^j \qquad (1)$$

Where $a_m^j = a_{m(mod\ 8)}^j \forall\ m \in Z \cap (8, \infty)$ and $n \in \{1,2,3,4\}$, Let $MCS_n(a^j)$ be the maximum of $CS_n(a^j, i)$ where $i \in Z \cap [0,7]$.

Algorithm 1 describes the GD-SIFT descriptor generation. Only 8 positions are available in the consecutive sum-n. Therefore, during the experiment, we compute sum-3 and sum-1 to represent the whole SIFT descriptor for each key-point as 48 bits and 16 bits, respectively.

Algorithm 1: GD-SIFT descriptor generation

Input: $F_{t_k},\ k = 1,2,\dots,m$ *// the set of key-frames*

Output: $GD(F_{t_k}),\ k = 1,2,\dots,m$ *// a GD-SIFT for the key-frames*

Start

1. For each $F_{t_k},\ k = 1,2,\dots,m$
   1.1. Compute $D(F_{t_k})$ // The 128 SIFT descriptor for F_{t_k}
   1.2. Separate $D(F_{t_k})$ into 16 sub-vectors $a^j = \left[a_0^j, \dots\dots\dots, a_7^j\right]^T, j \in Z \cap [0,15].$
   1.3. Find the position of the maximum consecutive sum-n of $a^j : p^j = argmax_{i\in z\cap[0,7]}\ CS_n(a^j, i).$
   1.4. Encode the SIFT feature by $n \times 16$ bits
2. End loop

End

Step-A3: Clustering (time-constraint algorithm)

The objective of this step is to group the similar descriptors together and then select the most global descriptor per each group. The resulted representative descriptors should reflect the content and the structure of the video [26]. Therefore, we adopt the time-constraint cluster algorithm, as demonstrated in algorithm 2. The advantage of the time-constraint cluster is to group similar video frames together with its natural time ordering.

The time-constraint cluster algorithm has O(n) complexity where n is the number of video key-frames.

Where BF is the Brute Force matcher [24] and the threshold ε^c is used to control the similarity between descriptors. Through the experiment, we examined different threshold values and found that values between 0.04 and 0.2 are often good values.

For each cluster, the representative (global) descriptor is constructed by select the key-points that appear in all descriptors within the cluster.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Algorithm 2: the time-constraint cluster algorithm

Input: $GD(F_{t_k}),\ k = 1,2,\dots,m$ *// a GD-SIFT for the key-frames*

Output: $C_i,\ i = 1,2,\dots,s\ ;\ s < m$ *// a set of clusters*

Start

1. Initialize $i = 1$
2. Add F_{t_1} to the cluster C_i whose cluster centroid is $o_i = GD(F_{t_1})$
3. Loop for each $k: 2 \rightarrow m$
4. If BF$(GD(F_{t_{k1}}), o_i) \leq \varepsilon^c$) then
5. Add F_{t_k} to the cluster C_i
6. Update $o_i = \text{BF}(GD(F_{t_{k1}}), o_i)$
7. $k = k + 1$
8. Else
9. $i = i + 1$
10. Add F_{t_k} to the cluster C_i whose cluster centroid is $o_i = GD(F_{t_k})$
11. $k = k + 1$
12. End loop

End

3.2 Query Stage: An Online Stage

The goal of this stage is to analysis the input query image in order to get a better matching with the stored videos descriptors. The query stage consists of three main steps. The first step is compute the global dominate SIFT descriptor for the input query image (similar as step-A2).

The second step is descriptor matching to find the best matches with in the stored descriptors in the database. We adapt the Brute Force Matcher with L2-norm distance for finding a best matches [24, 27]. Finally, we display the retrieved videos ordered by its matching rate.

4. EXPERIMENTS AND RESULTS

This section presents the experimental settings including the dataset, evaluation criteria for video matching, and the experimental evaluations.

A prototype was implemented to test the proposed framework using OpenCV [28] and FFmpeg [29] libraries. All the experiments were performed on a computer device equipped with Intel (IR) core ™ i7 CPU and 8GB of RAM.

4.1 Dataset

The experiments carried out on 30 videos from the standard dataset developed by Stanford [30]. The descriptions of these videos are listed in Table 2. All videos are in H.264/mp4 format with different properties. To evaluate the proposed framework, we build a video database of about 2 hours, i.e. about 201270 video frames.

4.2 Quality Evaluation

The quality of retrieved videos is measured by compute the accuracy and compare it with the traditional SIFT [8] and the Dominate SIFT [19].

Given a query image, we retrieved top n (e.g. 20-50) relevant videos and then computed the accuracy as follows:

$$Accuracy = \frac{number\ of\ correctly\ retrieved\ videos}{Total\ number\ of\ query\ videos} \quad (2)$$

On the considered video database four query groups with total 40 queries are fired. The first group consists of 10 original query images, see Table 3. Then some modifications (e.g. crop, rotate left and right) carried out to the original images. The objective is to measure the accuracy of the proposed methodology and to assert its effectiveness in different cases.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

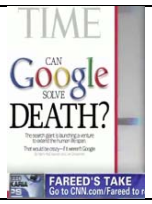
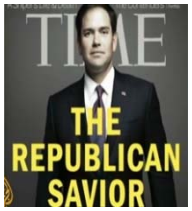
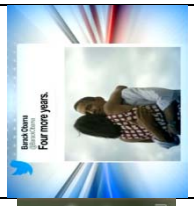
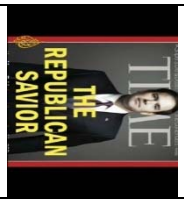
Table 2: Description Of Test Videos

Video no.	Video name	Duration	Size MB	No.Of Frames	Aspect ratio	Frame rate (FPS)	Resolution (W × H) (pixels)
1	Economist_V1	00:01:30	9.17	2160	16:9	24	854 × 480
2	Economist_V21	00:04:37	26.1	8310	16:9	30	854 × 480
3	Economist_V22	00:02:20	13.1	2400	4:3	30	720 × 480
4	Economist_V31	00:00:29	2.11	870	16:9	30	640 × 360
5	Economist_V32	00:02:47	15.9	5010	4:3	30	528 × 480
6	Economist_V33	00:00:29	11	870	16:9	30	640 × 360
7	Economist_V41	00:05:30	98	9900	16:9	30	280 × 720
8	Economist_V42	00:04:04	62.5	7320	16:9	30	280 × 720
10	Economist_V51	00:01:27	20.9	2610	16:9	30	280 × 720
11	Economist_V52	00:01:00	5.48	1800	16:9	30	280 × 720
12	Economist_V53	00:02:21	14.4	4230	16:9	30	854 × 480
13	Economist_V61	00:02:52	17.4	5160	4:3	30	640 × 360
14	Economist_V62	00:01:19	21.8	2370	16:9	30	720 × 480
15	Economist_V63	00:01:20	8.88	2400	4:3	30	280 × 720
16	Economist_V71	00:14:09	84.7	25470	4:3	30	720 × 480
17	Economist_V72	00:03:46	22.6	6780	4:3	30	720 × 480
18	Economist_V81	00:19:47	118	35610	4:3	30	720 × 480
19	Time_V1_1	00:01:15	7.6	2250	4:3	30	720 × 480
20	Time_V1_2	00:04:17	25.6	7710	4:3	30	720 × 470
21	Time_V1_3	00:02:52	17	5160	16:9	30	720 × 470
22	Time_V2	00:04:00	22.5	7200	16:9	30	280 × 720
23	Time_V3_1	00:01:30	9.33	2250	16:9	25	854 × 480
24	Time_V3_2	00:01:30	9.33	2250	16:9	25	854 × 480
25	Time_V4_1	00:04:37	27.8	8220	4:3	30	720 × 470
26	Time_V4_2	00:12:20	73.3	22200	4:3	30	720 × 470
27	Time_V4_3	00:02:55	21.2	5250	16:9	30	280 × 720
28	Time_V5	00:02:29	15.8	4470	4:3	30	720 × 480
29	Time_V6_1	00:03:02	18.2	5460	4:3	30	720 × 470
30	Time_V6_2	00:03:06	44.2	5580	16:9	30	280 × 720

Journal of Theoretical and Applied Information Technology
15th October 2019. Vol.97. No 19

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 3: *Examples Of The Query Images*

Groups	Query Images				
Group 1 (original Image)					
					
Group 2 (Cropped Image)					
					
Group 3 (Rotate Right Image)					
					
Group 4 (Rotate Left Image)					
					

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 4 shows the comparative results. The results demonstrate that, the proposed descriptor achieved an average accuracy of 0.775 with respect to the other compared descriptors. Moreover, the results indicated that, the proposed descriptor has high accuracy in case rotation left and right, as shown in figure 2. This is explained as the reduction of the false matches that is issued by SIFT.

Table 4: The Accuracy Of Different Descriptors

Descriptor / Query Groups	SIFT [8]	Dominate SIFT [19]	Global Dominate SIFT (proposed)
	Accuracy	Accuracy	Accuracy
Group1 (Original Image)	1	1	**1**
Group 2 (Cropping)	0.7	0.5	**0.7**
Group 3 (Rotate Left)	0.6	0.3	**0.8**
Group 4 (Rotate right)	0.5	0.4	**0.6**
Average	0.7	0.55	**0.775**

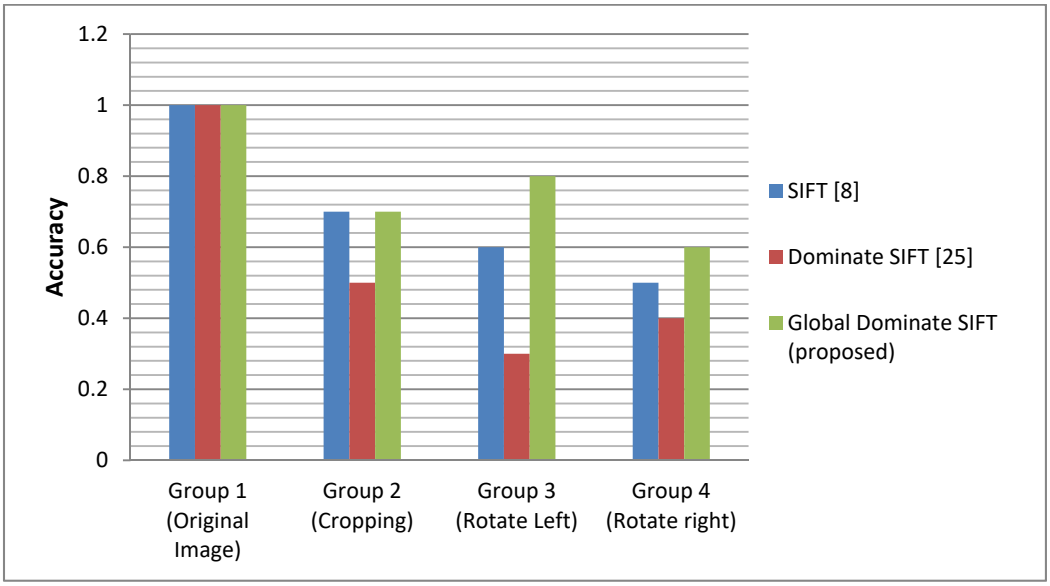


Figure 2: Descriptor Accuracy Evaluations for Original SIFT, Dominate SIFT and Our Proposed Global Dominate SIFT

4.3 Efficiency Evaluation

Reducing the storage space and increasing the retrieval speed are very important criteria for any video retrieval system. Therefore, the efficiency of the proposed descriptor is evaluated by computing the Average Retrieved Time (ART) and the required storage space. Table 5 shows the ART of different descriptors. The results demonstrate that, the proposed descriptor achieved a low ART value, 10.2 seconds, with respect to the other compared descriptors. Therefore, the proposed descriptor can considered as a promising solution for online applications. It is important to note that those results depend on the computational power of the target mobile device.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 5: The Average Retrieved Time (ART) Of Different Descriptors

Descriptor / Query Groups	SIFT [8]	Dominate SIFT (3) [19]	GD-SIFT (The proposed)
	ART (in second)	ART (in second)	ART (in second)
Group 1 (Original Image)	47.87	19.77	**11.05**
Group 2 (cropping)	52.16	18.23	**9.73**
Group 3 (Rotate Left)	48.29	16.34	**10.41**
Group 4 (Rotate right)	48.17	18.19	**9.59**
Average	49.12	18.13	**10.20**

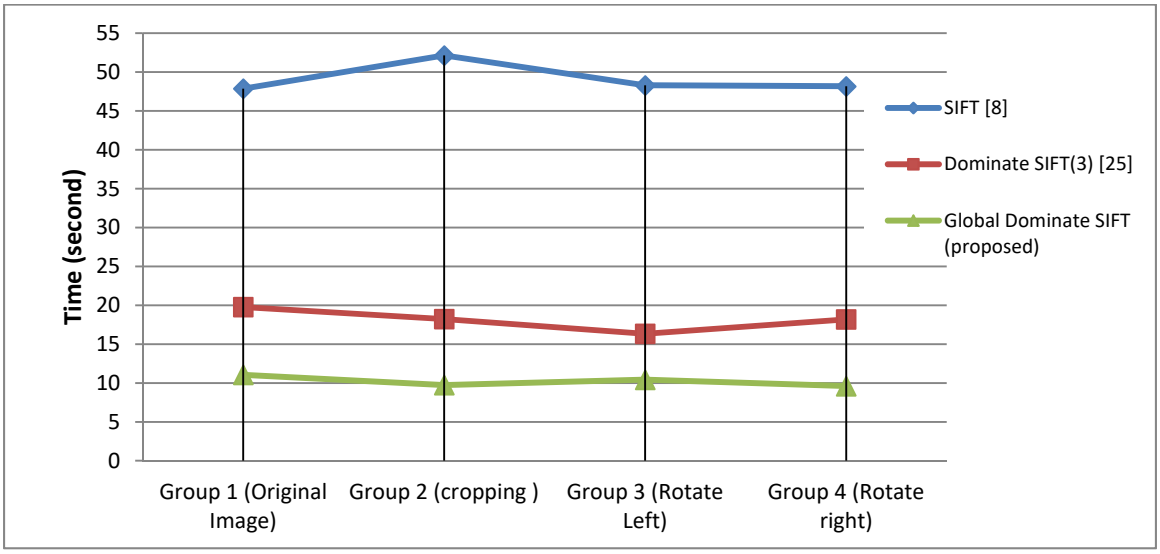


Figure 3: *Descriptor ART Evaluations For Original SIFT, Dominate SIFT and Our Proposed Global Dominate SIFT*

The global dominate SIFT uses 16 bits to represent each key-point which is 8 times and 3 times more compact than the original SIFT [8] and the Dominate SIFT(3) [19], respectively.

As shown in figure 4, the actual required space to store all the video attributes and descriptors are 212 MB for the global dominate SIFT, 610MB for dominate SIFT (3) and 1620MB for the original SIFT. Video attributes include code, name, resolution, size and location on the disc.

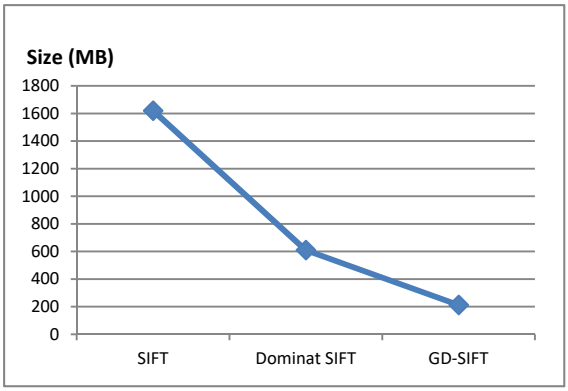


Figure 4: *Descriptor Memory Evaluations For Original SIFT, Dominate SIFT and Our Proposed GD-SIFT.*

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

4.3 Discussion

The proposed GD-SIFT requires only 16 bits to represent each key-point. Moreover, the time-constraint cluster algorithm was adopted to group the similar descriptors. For each cluster, the representative (global) descriptor was constructed by preserving the key-points that appear in all descriptors within the cluster. As shown in table 1, the GD-SIFT differs from other methods in the key-points extraction and description steps. It is worth mention that, the GD-SIFT is very suitable for video indexing and retrieving applications.

Although the proposed descriptor requires less storage, the required storage should decreased to be more suitable for a real application. For achieving better accuracy, motion features should be considered. Motion features are important for video indexing and retrieval. Extract moving objects, distinguish between camera motion, foreground motion and background motion. Combine motion features and static features are important for video indexing and retrieval.

5. CONCLUSIONS

In this paper, we proposed an efficient and effective video indexing and retrieving framework. This framework is based on a new compact descriptor which called GD-SIFT. The GD-SIFT descriptor used 16 bits to represent each key-point. Our experimental result shows that, the GD-SIFT descriptor achieved a high accuracy (an average of 78%) and more efficient (in terms of speed and storage) with respect to the related works. Moreover, the results indicated that, the proposed descriptor is more robust to variations (e.g. Scale, rotation, etc.). Importantly, the proposed descriptor is suitable for online and real-time applications and no need any vocabulary nor training.

REFERENCES:

[1] M. Ravinder and T. Venugopal, "Content Based Video Indexing and Retrieval Using Key Frames Discrete Wavelet Center Symmetric Local Binary Patterns (DWCSLBP)," *International Journal of Computer Science and Information Security,* vol. 14, no. 5, p. 699, 2016.

[2] K. Uma, B. Shekar, and M. Smitha, "Video clip retrieval: An integrated approach based on KDM and LBPV," in *Advances in Computing, Communications and Informatics (ICACCI), 2017 International Conference on*, 2017, pp. 1613-1618: IEEE.

[3] M.-H. Park and R.-H. Park, "EFFICIENT VIDEO INDEXING FOR FAST-MOTION VIDEO," *International Journal of Computer Graphics & Animation,* vol. 4, no. 2, p. 39, 2014.

[4] S. Kaavya and G. LakshmiPriya, "Multimedia Indexing and Retrieval: Recent research work and their challenges," in *Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on*, 2015, pp. 1-5: IEEE.

[5] M. P. Chivadshetti, M. K. Sadafale, and M. K. Thakare, "Content Based Video Retrieval Using Integrated Feature Extraction."

[6] I. Ihrke, K. N. Kutulakos, H. P. Lensch, M. Magnor, and W. Heidrich, "State of the art in transparent and specular object reconstruction," in *EUROGRAPHICS 2008 STAR–STATE OF THE ART REPORT*, 2008: Citeseer.

[7] G. M. Farinella, S. Battiato, and R. Cipolla, *Advanced topics in computer vision*. Springer, 2013.

[8] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision,* vol. 60, no. 2, pp. 91-110, 2004.

[9] Y. Ke and R. Sukthankar, "PCA-SIFT: A more distinctive representation for local image descriptors," in *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, 2004, vol. 2, pp. II-II: IEEE.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Computer vision and image understanding,* vol. 110, no. 3, pp. 346-359, 2008.

[11] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 2005, vol. 1, pp. 886-893: IEEE.

[12] J. He, S.-F. Chang, R. Radhakrishnan, and C. Bauer, "Compact hashing with joint optimization of search accuracy and time," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011, pp. 753-760: IEEE.

[13] B. Girod, V. Chandrasekhar, R. Grzeszczuk, and Y. A. Reznik, "Mobile visual search: Architectures, technologies, and the emerging

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

MPEG standard," *IEEE MultiMedia,* no. 3, pp. 86-94, 2011.

[14] M. Calonder, V. Lepetit, C. Strecha, and P. Fua, "Brief: Binary robust independent elementary features," in *European conference on computer vision*, 2010, pp. 778-792: Springer.

[15] A. Alahi, R. Ortiz, and P. Vandergheynst, "Freak: Fast retina keypoint," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 510-517: Ieee.

[16] V. Chandrasekhar, G. Takacs, D. Chen, S. Tsai, R. Grzeszczuk, and B. Girod, "CHoG: Compressed histogram of gradients a low bit-rate feature descriptor," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 2009, pp. 2504-2511: IEEE.

[17] J. Chen, L.-Y. Duan, R. Ji, and Z. Wang, "Multi-stage vector quantization towards low bit rate visual search," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2012, pp. 2445-2448: IEEE.

[18] A. T. Tra, W. Lin, and A. Kot, "Dominant SIFT: A novel compact descriptor," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 1344-1348: IEEE.

[19] A. T. Tra, W. Lin, and A. Kot, "Dominant SIFT: A novel compact descriptor," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 1344-1348: IEEE.

[20] R. E. G. Valenzuela, W. R. Schwartz, and H. Pedrini, "Dimensionality reduction through PCA over SIFT and SURF descriptors," in *Cybernetic Intelligent Systems (CIS), 2012 IEEE 11th International Conference on*, 2012, pp. 58-63: IEEE.

[21] A. E. Abdel-Hakim and A. A. Farag, "CSIFT: A SIFT descriptor with color invariant characteristics," in *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, 2006, vol. 2, pp. 1978-1983: Ieee.

[22] V. Chandrasekhar *et al.*, "Survey of SIFT compression schemes," in *Proc. Int. Workshop Mobile Multimedia Processing*, 2010, pp. 35-40.

[23] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Computer vision and pattern recognition, 2006 IEEE computer society conference on*, 2006, vol. 2, pp. 2161-2168: Ieee.

[24] Z. Pusztai and L. Hajder, "Quantitative comparison of feature matchers implemented in OpenCV3," 2016.

[25] H. Farouk, K. ElDahshan, and A. A. E. Abozeid, "Effective and Efficient Video Summarization Approach for Mobile Devices," *International Journal of Interactive Mobile Technologies (iJIM),* vol. 10, no. 1, pp. 19-26, 2016.

[26] H. Karray, M. Ellouze, and A. Alimi, "Indexing video summaries for quick video browsing," in *Pervasive Computing, Springer*, 2010, pp. 77-95.

[27] J. T. Arnfred and S. Winkler, "Fast-Match: Fast and robust feature matching on large images," in *2015 IEEE International Conference on Image Processing (ICIP)*, 2015, pp. 3000-3003: IEEE.

[28] OpenCV. (3/2019). *OpenCV library*. Available: https://opencv.org/

[29] FFmpeg. (3/2019). *FFmpeg Library*. Available: https://ffmpeg.org/

[30] A. Araujo, J. Chaves, D. Chen, R. Angst, and B. Girod, "Stanford I2V: a news video dataset for query-by-image experiments," in *Proceedings of the 6th ACM Multimedia Systems Conference*, 2015, pp. 237-242: ACM.