Compact Signature-Based Compressed Video Matching Using Dominant Color Profiles (DCP)

Saddam Bekhet
School of computer science
University of Lincoln
Lincoln, UK
sbekhet@lincoln.ac.uk

Amr Ahmed
School of computer science
University of Lincoln
Lincoln, UK
aahmed@lincoln.ac.uk

***Abstract*— This paper presents a novel technique for efficient and generic matching of compressed video shots, through compact signatures extracted directly without decompression. The compact signature is based on the Dominant Color Profile (DCP); a sequence of dominant colors extracted and arranged as a sequence of spikes in analogy to the human retinal representation of a scene. The proposed signature represents a given video shot with ~490 integer values, facilitating for real-time processing to retrieve a maximum set of matching videos. The technique is able to work directly on MPEG compressed videos, without full decompression, as it utilizes the DC-image as a base for extracting color features. The DC-image has a highly reduced size, while retaining most of visual aspects, and provides high performance compared to the full I-frame. The experiments and results on various standard datasets show the promising performance, both the accuracy and the efficient computation complexity, of the proposed technique.**

Keywords—Video matching; DC-image; Video similarity; Dominant color profile; compressed video

I. INTRODUCTION

Since the proliferation of multimedia recording technologies and the exponential growth of storage mediums, videos became a major aspect of our life. More than 100 hours of video are uploaded to YouTube every minute and more than 6 billion hours of video are being watched each month [1]. As a result, the available volumes of videos are of incredible size especially those in compressed formats (e.g. MPEG). This emphasizes the need for efficient matching and retrieval systems, which are able to operate in a real-time manner to satisfy user needs. Utilization of low level features extracted directly from a compressed video frames is crucial as it avoids the lengthy process of decompressing a video to extract such features. This is particularly useful for real-time processing. The DC-image as a compressed domain feature proved to be a powerful feature. It was reported to be up to 62 times faster, in matching, than the full I-frame while achieving similar or better matching results based on local features [2]. Thus, the proposed DCP utilizes the DC-image by extracting dominant color information and arranges it in the form of spikes; which is analogues to the representation done by human retina in response to a scene's visual information as discussed later in section III. The paper is organized as follows; section II will present a brief biological background of human vision (spikes) to explain its analogy with the proposed technique, while section III will present the related work followed by the proposed DCP with its analysis and supporting experiments in sections IV, V and VI. Finally the paper is concluded in section VII.

This work is funded by SouthValley University –Egypt

II. HUMAN VISION SYSTEM

The notion of video 'frame' in computer vision is compulsory, since the only available imaging input devices are frame-based, that samples scene frames regularly at a constant rate even if they do not introduce new information, which acts as a burden on vision algorithms as it wastes processing time [3]. However, biological research indicated that humans tend to see in a frame-free scenario [4], as it was discovered that biological neurons in human's retina (photoreceptors) only emit electrical impulses (*spikes*) in response to incident photons from current scene triggered by particular event such as; local luminance increases or decreases [5]. Then, information is sent to the brain as a wave of spikes with specific timing for further complex processing [6, 7]. This confirms that the human's eye is a change detecting device for frame-free vision [4]. *Fig.1* depicts the structure of the human retina, showing the location of the photoreceptors that are responsible for firing the spiral spikes pattern being triggered by a rotating disk with black dot.

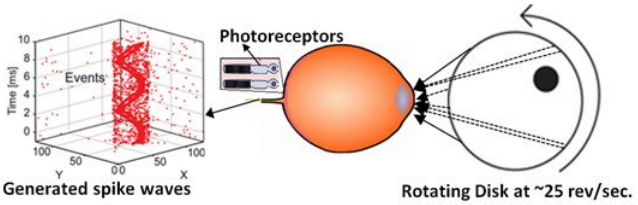

Figure 1. Illustration of human retina showing the photoreceptors firing spikes in response to a rotating disk (adapted from [3]).

As a core conclusion, biologists identified that the power of the human's visual systems is attributed to two distinct factors. The first is the representation of the visual information sent to the brain, while the second is processing of the information inside the brain. Although much research done in both, the inner brain processing still has unrevealed secrets. But there is some sort of agreement between scientists about the representation of visual information as being absorbed by human retina in the form of spikes in response to incident

1051-4651/14 $31.00 © 2014 IEEE
DOI 10.1109/ICPR.2014.674

light[6, 7]. Knowing that there is no complex scene analysis being done at this early stage, even when the information reaches the brain, it is being processed in its native format (spikes), and this inspires our design of the DCP. The result is that each video is mapped to a unique signature consisting of a sequence of spatio-temporal spikes that encodes color changes for each block across video frames which represented by the luminance information extracted from video frames in a way similar to the spikes, and is able to represent scenes in analogues way to human retina which results in efficient and compact signature to match video shots. The DCP is presented in detail in section III, following the literature review.

III. LITERATURE REVIEW

In this section, we review previous key work related to the compressed domain video matching. An MPEG compressed stream is rich with pre-computed features; Table1 depicts a generic comparison between various features available in MPEG stream, referring to the most recent work on each feature. As seen from Table1, the DC-image is a powerful feature of MPEG compressed stream due to its small size, which makes it faster to perform various video analysis tasks as fully discussed in [2]. Some of the early and on-going work in compressed video matching emerged from non-compressed video matching work. For example, motion vectors based trajectories [8] were used for matching instead of keypoints based trajectories, as they are pre-computed during the MPEG encoding process. However, obtaining the motion vectors still requires partial decoding; especially they are not available for **I**-frames.

TABLE I. CLASSIFICATION COMPRESSED DOMAIN FEATURES

Feature	Type	Pros.	Cons.
DC coefficients	**Spatial**	▪ No decompression to extract from **I**-frames [2]. ▪ Used as a replacement of **I**-frames [2]. ▪ Efficient for copy detection [13]. ▪ Fast for complex operations.	▪ Needs special attention to extract interest points, due to its small size [2]. ▪ Full decompression is needed case extracted from **P** & **B** frames.
AC coefficients	**Spatial**	▪ Partial decompression is needed for extraction.	▪ Do not reveal any visual information unless reconstructed [14].
Motion Vectors	**Temporal**	▪ Partial decompression is needed for extraction. ▪ Pre-computed motion feature. ▪ Efficient in shot detection [15].	▪ Describes block movement and do not carry motion information across GOP's [16]. ▪ Only for **P**&**B** frames. ▪ Do not encode any visual information.
Macroblock Types.	**Spatial**	▪ Partial decompression needed for extraction. ▪ Suitable for copy detection and fingerprinting [13, 17].	▪ Encodes only metadata about block compression information (ex. intra coded, skipped) [13]. ▪ Do not encode any visual information.

A generic utilization for motion vectors in conjunction with DC coefficients was adapted in [9], where the aggregation of both values used as a video signature, while the actual matching is done using the sliding window technique, by computing direct difference between adjacent DC values and motion vectors for currently matching frames pair. An apparent drawback of such approach, is that the DC values was used as a set of numeric values, rather than an image, which ignores the visual information that can potentially be extracted from the DC-image. In addition to the exhaustive search for the sliding window that uses frame-to-frame matching. A different technique, implemented in [10], by utilizing the DC-image to identify keyframes. Then, salient regions were extracted from the full size keyframes and tracked using their respective SIFT keypoints across consecutive **I**-frames for later matching. Although this approach uses DC-image to extract keyframes to reduce computational cost, still a full decompression is done to reconstruct the full **I**-frames which is not effective for real-time processing.

Hua et al. [11] attempted to use ordinal measures as a video signature extracted from fully decompressed video frames, which does not suit real-time processing neither takes benefit of any MPEG features. This problem was tackled by Almeida et al. [12] by using **I**-frame DC values which act as pre-computed ordinal measures and implemented a motion histogram signature by computing temporal and spatial ordinal matrices for each **I**-frame. Both matrices are combined to form a normalized 6075 floating-point bin histogram, which is a quit large signature for matching in large databases.

In an attempt to standardize image and video retrieval descriptors, MPEG-7 released a group of descriptors [18]. However, 70% (11 out of 17) were dedicated for images and cannot handle videos effectively, due to the temporal nature of videos. Thus, more dedicated research is needed for effective video matching. The term "tiny image" [19] was introduced during an attempt to construct a database of 80 million small images of size 32×32. The dataset was used to perform object and scene recognition by fusing metadata extracted from WordNet [20] with visual features extracted from images, based on nearest neighbour methods. Later, the concept of tiny images was adopted for videos. The aim was an attempt to build a database of tiny videos [21], of approximately 50,000 videos, reconstructed by sampling full videos into 40×30 pixel frames, with their available annotations. The dataset were tested for video retrieval using sum of squared pixel difference (SSD) measure [19] between videos' keyframes, leaving any temporal information without utilization and relying on annotations which is neither accurate (due to dependency on human element) nor always available. Color proved to be a powerful feature regarding image retrieval [22, 23] and video retrieval [24, 25], as it's strongly related to semantic similarity [26, 27]. Color by nature is invariant to partial occlusion, cropping, translation or affine transformations such as scaling, rotation, shear or reflection [24]. Color feature is very powerful and could act as a building block for efficient video matching techniques, especially in absence of any semantic

cues [26]. Furthermore, humans tend to see scenes as a set of dominant colors [26, 28] as it was found that a small number of colors are sufficient to describe any region instead of a full color space [29]. Those important remarks were taken by researchers to utilize and develop more efficient video retrieval techniques starting by using the popular color histogram extracted from DC-images to act as signature for video retrieval [30, 31], or even the more sophisticated histograms such as Dominant Color Histograms (DCH) [32]. DCH procedure starts by converting the RGB frame in to quantized HSV frame, followed by extraction of dominant colors from each frame and maps them to a quantized histogram that keeps only longer duration dominant colors across each shot. DCH was used for video retrieval purpose [33] and for object tracking in CCTV videos [34]. Recently it was used for video summarization [35]. DCH is still a global feature that doesn't encode neither spatial nor temporal information [26]; also it is a color space dependent which makes it not effective for robust video retrieval. Those problems will be covered along with the DCP in next section.

IV. PROPOSED DOMINANT COLOR PROFILE

In this section we present the basis of our proposed Dominant Color Profile (DCP) with the relevant arguments and supporting evidences. Basically, the core idea of the DCP is that every block of each DC-image is being represented by its dominant color (spike), where the sequence of dominant colors for each block is kept as descriptive color profile. The whole set of all blocks DCP's across video acts as a compact signature for the entire video, where the spatial localization of the spikes is preserved with each block position, and the temporal localization preserved in DCP sequence order across video frames. Thus, each video is mapped to a sequence of spatio-temporal spikes that encodes color changes for each block across video frames.

The DCP is built based on three important remakes emerges from previous literature review which are; (1) *DC-image,* (2) *spikes and* (3) *dominant colors*. The process of DCP construction starts by dividing the DC-image into blocks (for a DC-image of size 40×30, 49 blocks are used with block size of ~25 pixels). For each block, the dominant color is extracted to act as a spike for this block, and the process repeated for every I-frame's DC-image. Thus each block will have its own DCP, which acts as a wave of spikes that describes the block's behavior across the video. In this way, the group of DCP's for all blocks acts as a signature that encodes a video as a series of dominant color spikes in analogies with the human's retina scene representation.

Consider two videos $\mathbf{V_1}= \{\mathbf{f_1},\mathbf{f_2},\mathbf{f_3},\ldots\mathbf{f_n}\}$ and $\mathbf{V_2}= \{\mathbf{f_1},\mathbf{f_2},\mathbf{f_3}\ldots\mathbf{f_m}\}$ where **m** and **n** are the number of I-frames of videos **V1** and **V2** respectively. Each frame $\mathbf{f_i}$ of a given video, represented by its DC-image, will be divided into **Z** blocks $\mathbf{f_i}= \{\mathbf{b_1},\mathbf{b_2},\mathbf{b_3},\ldots\mathbf{b_z}\}$. For each block $\mathbf{b_j}$, its respective dominant color $\mathbf{d_j}$ is being extracted to act as a spike fired from this specific block. Thus, a given frame $\mathbf{f_i}$ will be represented by its respective sequence of blocks dominant colors $\{\mathbf{d_1},\mathbf{d_2},\mathbf{d_3},\ldots\mathbf{d_z}\}$.

This process of dividing into blocks and extracting dominant colors will be repeated for every **I**-frame's DC-image, where the dominant color $\mathbf{d_j}$ of each block $\mathbf{b_j}$ at every frame $\mathbf{f_i}$ will be concatenated together to form a Dominant Color Profile which will be called block's DCP. The entire video's DCP can be defined as following,

$$\mathbf{V_1DCP} = \left\{\{\mathbf{d_1^1} \cup \mathbf{d_1^2} \ldots \cup \mathbf{d_1^z}\} \cup \{\mathbf{d_2^1} \cup \mathbf{d_2^2} \ldots \cup \mathbf{d_2^z}\} \ldots \cup \{\mathbf{d_n^1} \cup \mathbf{d_n^2} \ldots \cup \mathbf{d_n^z}\}\right\}$$

$$\mathbf{V_2DCP} = \left\{\{\mathbf{d_1^1} \cup \mathbf{d_1^2} \ldots \cup \mathbf{d_1^z}\} \cup \{\mathbf{d_2^1} \cup \mathbf{d_2^2} \ldots \cup \mathbf{d_2^z}\} \ldots \cup \{\mathbf{d_m^1} \cup \mathbf{d_m^2} \ldots \cup \mathbf{d_m^z}\}\right\}$$

, where $\mathbf{V1^{DCP}}$ will be video1 signature and $\mathbf{V2^{DCP}}$ will be video2 signature. The similarity of two videos is then measured by the distance between their respective DCP signatures using the euclidean distance, due to its robustness and efficiency [36], as in equation 1:

$$\mathbf{D(V_1, V_2) = Euclidean(V_1^{DCP}, V_2^{DCP})} \qquad (1)$$

To facilitate for the matching process case unequal length videos (also, unequal length DCPs); we adopted a simple procedure by expanding the shorter signature with appended DCP values copied from its beginning.

A. Extraction of dominant colors

Through literature exists two main methods to extract dominant colors; clustering [29, 37] and quantization [38, 39]. Regards clustering the general idea is to group similar pixel colors into set of clusters, where each cluster is represented by its centroid [40], which acts as the dominant color. However, there are some problems associated with clustering algorithms in general; regarding the excessive computational time to find clusters [41] and the manual initialization of initial cluster seeds [38]. Also, in some cases the final set of selected dominant colors may be far away from those identified by human as dominant colors [29]. For quantization, the process operates by mapping each range of colors to one representative color [39], by separating continuous colors into quantized groups. Generally there are two major problems associated with quantization techniques:

- Results loss; as an entire color space (e.g. >14 million colors in RGB) will mapped to a small set of colors.
- Quality of quantized colors depends on a predefined quantization parameter used to group similar colors, such parameter is color space dependent.

As a conclusion, neither clustering nor quantization is perfect for dominant colors extraction. But quantization is advantageous over clustering as it operates in real-time; in addition the major problem with quantization comes from mapping an entire color space to a small set of representative colors as expressed in (2):

$$\mathbf{Efficiency} \propto \frac{\mathbf{1}}{\mathbf{color\ space\ size}} \qquad (2)$$

Since MPEG is natively subsampled using YC_bC_r color space, the full resolution grayscale Y-channels could be used as a base for DCP, to improve the quality of quantized colors since it's mostly consisting of 256 intensity levels, and not necessarily losing information, since human eye is more sensitive to intensity changes rather than chrominance changes [42]. As a practical example, considering the RGB color space

with more than 14 million colors quantized to 71 values though HSV color space [39], this means that on average 197183 colors will be mapped to one color. For the grayscale and assuming a quantization of value of 16, this means that each 16 colors will be mapped to one color, which is more distinctive than quantizing a full color space.

There are a number of factors and issues that had to be analyzed for the DCP to be efficient, such as *number of blocks* per DC-image, *number of dominant colors* per block and *quantization factor*. Each factor is investigated in next sub-sections, where all experiments tested on two standard datasets; BBC Rushes [43] and UCF11 [44] datasets. The first is a standard data set for video retrieval and contains diverse set of challenging videos; mainly man-made moving objects (cars, tanks, planes and boats), while the second is a standard dataset for action recognition used widely for retrieval purposes as videos contains large variations in object appearance, pose, scale as well as camera movement. The performance of DCP is evaluated using **p**recision-over-***N*** (***P*$_N$**) standard measure [45], calculated over three ranks; *first*, *fifth* and *tenth*, where a weighted average is calculated for those ranks as in equation (3), given that **{α, β, γ}** are the weighting parameters; and **α > β > γ**, which gives more weight to higher ranks; as we want to maximize the set of correct retrieved matches. In following experiments we set **α, β** and **γ** to **1**, **0.8** and **0.2** respectively.

$$\mathbf{wAvg} = \frac{(\alpha \boldsymbol{P_{10}} + \beta \boldsymbol{P_5} + \gamma \boldsymbol{P_1})}{\mathbf{3}} \qquad (3)$$

B. *Effective number of blocks per DC-image.*

The first DCP factor is the number blocks per DC-image to extract dominant colors from. We found that blocking slightly increases the retrieval precision (up to certain level) as it forces more spatial matching constraints, allowing each part in DC-image to contribute in the overall video DCP. *Fig.2.a* shows the effect of increased blocks number against the weighted average precision curve, for BBC RUSHES dataset. It was detected that 49 blocks yields notable high precision, corresponding to 0.68% increase over no-blocking at all, which is equivalent to 24% percent increase in top-1 precision (from 41% to 51%). *Fig.2.b* depicts same graphs but for UCF11 dataset, it shows that 49 blocks is still distinguishable as it yields in 22% increase in precision than no-blocking. Keeping in mind that increased blocking per DC-image adds more computational cost which affects computation complexity without any major precision gain. Thus 49 blocks (7×7) is chosen as the effective blocking size per DC-image.

B. *Effective number of dominant colors.*

The second factor is the number of dominant colors- per each DC-image block that will be aggregated in the block DCP. *Fig.3.a* and *Fig.3.b* confirms that one dominant color per block achieves higher results, as increased number of dominant colors did not dramatically improve the results; despite it increases the matching time and size of the signatures.

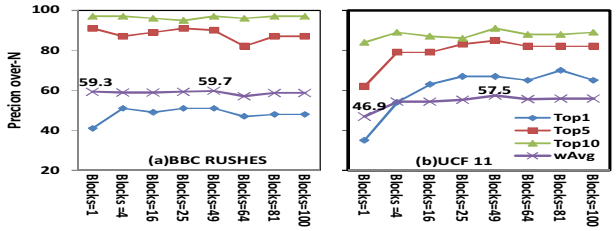

Figure 2. Effect of number of blocks on the precision-over-N curves for (a) BBC RUSHES and (b) UCF11 datasets.

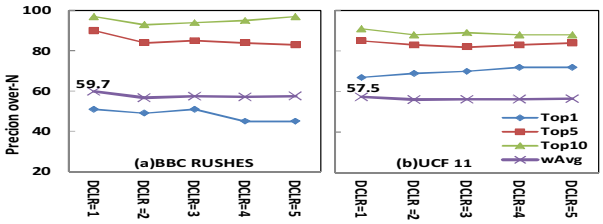

Figure 3. Effect of number of dominant colors on precision-over-N curves for (a) BBC RUSHES and (b) UCF11 datasets.

C. *Effective quantization parameter.*

Quantization parameter is an important factor in DCP design, where the idea is to map different degrees of the same color to their basic color e.g. pale white and white should be grouped as white, this is illustrated in *Fig.4*, as it shows 256 grayscale levels quantized to two different sets; 32 levels and 16. We can find that as the quantization value increase we get less discrimination between colors. *Fig.5.a* and *Fig.5.b* depicts the effect of increased quantization value against (***P*$_N$**) curves, we notice that a quantization value of 16 could be distinguished from no-quantization effect, with only ~0.84% and ~1.2% increase for BBC RUSH and UCF11 respectively. This small difference because the grayscale is naturally quantized.

Figure 4. Grayscale levels quantized into two different levels

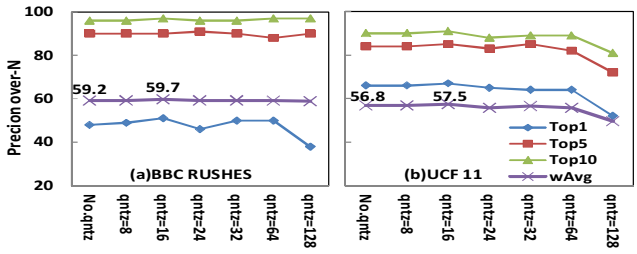

Figure 5. Effect of quantization parameter on precision-over-N curves for (a) BBC RUSHES and (b) UCF11 datasets.

C. *Effective Selection of color space.*

Most of the work on dominant colors either uses RGB [41] or HSV [39]. Furthermore others claimed that even a specific color space is not an important factor in dominant color extraction [41]. In our case we are using the grayscale MPEG Y-channel for several reasons:

- It could be extracted without full decompression.

- It's a neutralized way to represent DCP without being biased to a specific color space.
- MPEG stream is encoded natively using YC_bC_r, thus no further decompression needed, which saves time.
- Human's eye is more sensitive to luminance changes rather than chrominance changes [42].
- Quantization in grayscale is much easier and natively related to the idea of light distribution as we have only 256 grayscale levels, while in RGB or HSV complex approaches need to be considered.

To verify our assumptions we tested grayscale DCP over versus DCP over HSV. For HSV conversion, we adopted a widely used conversion algorithm from [39] that maps RGB color space to 71 HSV values. *Fig.6.c* depicts comparison between grayscale DCP and HSV DCP. We notice that grayscale DCP outperforms HSV DCP through ranks 1 to 7 and both merges together at ranks 8 to 10, which means higher results and less computing time. Thus, DCP over grayscale is selected, as no further processing needed for the MPEG grayscale Y-channel.

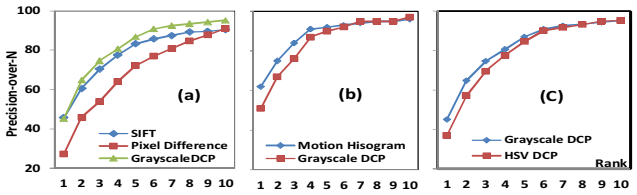

Figure 6. (a) DCP vs. SIFT vs. Pixel Difference (full BBC RUSH), (b) DCP vs. Motion Histogram (100 videos BBC RUSH) and (c) HSV DCP vs. grayscale DCP (full BBC RUSH).

V. DCP VERSUS STATE OF ART BASELINES.

Regarding comparison against state of art baselines, we chosen SIFT enhanced by dynamic programming to incorporate video temporal dimension [2], motion histogram [12] and pixel difference implemented in tiny videos [21]. The first two baselines, work through compressed domain and consider only DC-image sequence, while the third operates on videos of the same size as DC-image. Regarding motion histogram, the results was obtained from the author of [12] directly, who cooperated by running his algorithm on a selected dataset from BBC RUSH. *Fig.6.a* and *Fig.6.b* depicts DCP ($\boldsymbol{P_N}$) curves against the baselines. It is clear that grayscale DCP shows a considerable improvement over SIFT and pixel difference in both high and low ranks. For motion histogram, on average it is 6% higher than DCP in lower ranks (top 1 to 6), while DCP have the same precision (or slightly higher) in higher ranks (7 to 10).

TABLE II. COMPUTATIONAL COST OF DCP VS. MOTION HISTOGRAM.

	Motion Histogram	**DCP**
Computation Complexity	O(3N) [b]	O(N) [b]
Signature Size per Video (~ 10 **I**-frames per shot [2]).	6075 floating point number	~ 490[a] integer numbers
Speed	———	~12 times faster.
Similarity Measure	Euclidean distance	

a. 49 blocks (7x7) * 10 I-Frames/video-clip

b. (N) is the number of processed video frames

Furthermore, *Table2* depicts a deeper abstract comparison between DCP and motion histogram, and shows that the DCP outperforms the motion histogram in terms of signature size and matching time (91% reduction in both), which makes it more suitable for real-time processing.

VI. TIMING ANALYSIS OF DCP.

Since real time constrains are crucial for DCP, this section provides timing analysis regarding for grayscale DCP against base lines. For motion histogram we do not have any information about timing except some abstract estimation presented in *Table2*. As depicted in *Table3* it is notable that all the techniques work in real time but, DCP is 49 times faster than SIFT and 6 times faster than pixel difference. This leaves more time for the DCP to extend and enhance its work, even for building further layers which operate on its output for further precise results.

TABLE III. TIMING ANALYSIS FOR DCP VERSUS. BASELINES.

	Average Frame Match Time(Milliseconds)
DCP	0.33
Pixel difference	2
SIFT	16.43

VII. CONCLUSION.

In this paper we proposed an efficient technique for matching compressed video shots, through compact signatures extracted directly without decompression, by using Dominant Color Profile (DCP). Taking advantage of the DC-image small size, DCP arranges color information in similar way to scene representation by the human's retina, in the form of spikes. Both spatial and temporal information are encoded within the DCP, in an efficient and compact way that suites real-time matching. In addition to evidences and experiments, a detailed analysis about various parameters that controls the DCP construction and behavior was presented, namely; quantization factor, number of blocks and number of dominant colors. The results obtained also proved DCP's robustness against various and challenging datasets and its ability to work in real-time environment. Furthermore, the DCP could act efficiently to retrieve an initial maximum set of matching videos through its efficient computations. It also facilitates for further layers to work on top for further re-ranking of the videos and/or for further semantic analysis and annotation such as in [46]. On the other hand, there are a number of improvements for the DCP, which we are working on. For example, we are working on a better encoding for DCP contents to be more compact and ideally of a fixed length signature, regardless of the video shot length. Moreover, plugging a second layer, with more sophisticated local features (e.g. SIFT) to improve the ranking of the selected maximum matching set.

REFERENCES

[1] (2013). *YouTube Statistics* [Online]. Available: http://www.youtube.com/yt/press/statistics.html.

[2] S. Bekhet, A. Ahmed and A. Hunter, "Video matching using DC-image and local features," in *Proceedings of the World Congress on Engineering,* UK, 2013, pp. 2209-2214.

[3] T. Delbruck and P. Lichsteiner, "Freeing vision from frames," *Neuromorphic Eng,* vol. 3, pp. 3-4, 2006.

[4] S. Thorpe, D. Fize and C. Marlot, "Speed of processing in the human visual system," *Nature,* vol. 381, pp. 520-522, 1996.

[5] P. Lichtsteiner, C. Posch and T. Delbruck, "A 128× 128 120 dB 15 μs Latency Asynchronous Temporal Contrast Vision Sensor," *IEEE Journal of Solid-State Circuits,* vol. 43, pp. 566-576, 2008.

[6] B. Linares-Barranco, "Spike-based vision processing. seeing without frames," in *IEEE International Symposium on Circuits and Systems(ISCAS '07),* 2006, .

[7] S. J. Thorpe, "Spike-based image processing: Can we reproduce biological vision in hardware?" in *Computer Vision Workshops(ECCV '12),* 2012, pp. 516-521.

[8] Z. Droueche, M. Lamard, G. Cazuguel, G. Quellec, C. Roux and B. Cochener, "Content-based medical video retrieval based on region motion trajectories," in *Proceedings of International Federation for Medical and Biological Engineering,* 2012, pp. 622-625.

[9] N. Dimitrova and M. S. Abdel-Mottaleb, "Video retrieval of MPEG compressed sequences using DC and motion signatures," *Google Patents,* 1999.

[10] Han-ping Gao and Zu-qiao Yang, "Content based video retrieval using spatiotemporal salient objects," in *International Symposium on Intelligence Information Processing and Trusted Computing (IPTC),* 2010, pp. 689-692.

[11] Xian-Sheng Hua, Xian Chen and Hong-Jiang Zhang, "Robust video signature based on ordinal measure," in *International Conference on Image Processing (ICIP '04),* 2004, pp. 685-688 Vol. 1.

[12] J. Almeida, N. J. Leite and R. da S Torres, "Comparison of video sequences with histograms of motion patterns," *IEEE International Conference on Image Processing,* 2011, pp. 3673-3676.

[13] A. S. Abbass, A. A. A. Youssif and A. Z. Ghalwash, "Compressed domain video fingerprinting technique using the singular value decomposition," in *Proceedings of Applied Informatics and Computing Theory,* Spain, 2012, .

[14] A. B. Watson, "Image compression using the discrete cosine transform," *Mathematica Journal,* vol. 4, pp. 81, 1994.

[15] P. Panchal and S. Merchant, "Performance evaluation of fade and dissolve transition shot boundary detection in presence of motion in video," in *Proceeding of Emerging Technology Trends in Electronics, Communication and Networking,* 2012, pp. 1-6.

[16] N. Dimitrova and F. Golshani, "Motion recovery for video content classification," *ACM Transactions on Information Systems (TOIS),* vol. 13, pp. 408-439, 1995.

[17] A. S. Abbass, A. A. A. Youssif and A. Z. Ghalwash, "Hybrid-Baesd Compressed Domain Video Fingerprinting Technique," *Computer and Information Science,* vol. 5, pp. p25, 2012.

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

[19] A. Torralba, R. Fergus and W. T. Freeman, "80 Million Tiny Images: A Large Data Set for Nonparametric Object and Scene Recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 30, pp. 1958-1970, 2008.

[20] G. Miller and C. Fellbaum, "Wordnet: An electronic lexical database," 1998.

[21] A. Karpenko and P. Aarabi, "Tiny Videos: A Large Data Set for Nonparametric Video Retrieval and Frame Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 33, pp. 618-630, 2011.

[22] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Qian Huang, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker, "Query by image and video content: the QBIC system," *Computer,* vol. 28, pp. 23-32, 1995.

[23] M. Sabitha and R. Hariharan, "Hybrid Approach for Image Search Reranking," *International Journal of Science and Research (IJSR),* vol. 2, pp. 123-128, 2013.

[24] T. J. Liu, H. J. Han, X. Xin, Z. Li and A. K. Katsaggelos, " A Robust And Lightweight Feature System For Video Fingerprinting," 2012.

[25] D. DeMenthon and D. Doermann, "Video retrieval using spatio-temporal descriptors," in *Proceedings of the 11th ACM international conference - on Multimedia,* 2003, pp. 508-517.

[26] A. Mojsilovic, Jianying Hu and E. Soljanin, "Extraction of perceptually important colors and similarity measurement for image matching, retrieval and analysis," *IEEE Transactions on Image Processing,* vol. 11, pp. 1238-1248, 2002.

[27] B. E. Rogowitz, T. Frese, J. R. Smith, C. A. Bouman and E. Kalin, "Perceptual image similarity experiments," *Human Vision and Electronic Imaging III,* vol. 3299, pp. 576-590, 1998.

[28] S. Kiranyaz, S. Uhlmann and M. Gabbouj, "Dominant color extraction based on dynamic clustering by multi-dimensional particle swarm optimization," in *7th International Workshop on Content-Based Multimedia Indexing,* 2009, pp. 181-188.

[29] Yining Deng, B. S. Manjunath, C. Kenney, M. S. Moore and H. Shin, "An efficient color representation for image retrieval," *IEEE Transactions on Image Processing,* vol. 10, pp. 140-147, 2001.

[30] M. R. Naphade, M. M. Yeung and B. L. Yeo, "A novel scheme for fast and efficient video sequence matching using compact signatures," in *Proc. Storage and Retrieval for Media Databases,* 2000, pp. 564-572.

[31] S. K. Avula and S. C. Deshmukh, "Frame based Video Retrieval using Video Signatures," *International Journal of Computer Applications,* vol. 59, pp. 35-40, 2012.

[32] T. Lin and H. Zhang, "Automatic video scene extraction by shot grouping," in *15th International Conference on Pattern Recognition,* 2000, pp. 39-42.

[33] Tong Lin, Chong-Wah Ngo, Hong-Jiang Zhang and Qing-Yun Shi, "Integrating color and spatial features for content-based video retrieval," in *Proceedings International Conference on Image Processing,* 2001, pp. 592-595 vol.3.

[34] Liyuan Li, Weimin Huang, I. Y. -. Gu, Ruijiang Luo and Qi Tian, "An Efficient Sequential Approach to Tracking Multiple Objects Through Crowds for Real-Time Intelligent CCTV Systems," *IEEE Transactions on Systems, Man, and Cybernetics,* vol. 38, pp. 1254-1269, 2008.

[35] S. S. Kanade and P. Patil, "Dominant Color Based Extraction of Key Frames For Sports Video Summarization," *Journal of Advances in Engineering & Technology,* vol. 6, pp. 504-512, 2013.

[36] Dengsheng Zhang and Guojun Lu, "Evaluation of similarity measurement for image retrieval," in *Neural Networks and Signal Processing, 2003. Proceedings of the 2003 International Conference On,* 2003, pp. 928-931 Vol.2.

[37] R. Jin, C. Kou, R. Liu and Y. Li, "A color image segmentation method based on improved K-means clustering algorithm," in *Proceedings of the International Conference on Information Engineering and Applications,* 2013, pp. 499-505.

[38] R. Roopalakshmi and G. R. M. Reddy, "Compact and efficient CBCD scheme based on integrated color features," in *International Conference on Recent Trends in Information Technology,* 2011, pp. 880-883.

[39] Hong Shao, Yueshu Wu, Wencheng Cui and Jinxia Zhang, "Image retrieval based on MPEG-7 dominant color descriptor," in *Proceedings of Young Computer Scientists,* 2008, pp. 753-757.

[40] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis.* Wiley New York, 1973.

[41] N. Yang, W. Chang, C. Kuo and T. Li, "A fast MPEG-7 dominant color extraction with new similarity measure for image retrieval," *Journal of Visual Communication and Image Representation,* vol. 19, pp. 92-105, 2, 2008.

[42] J. C. S. Yu, M. S. Kankanhalli and P. Mulhen, "Semantic video summarization in compressed domain MPEG video," in *Proceedings of International Conference on Multimedia and Expo ICME,* 2003, pp. III-329-32 vol.3.

[43] (2013). *Trec video retrival task, bbc ruch 2005 (1-02-2011)* [Online]. Available: www.nplpir.nist.gov/projects/trecvid.

[44] J. Liu, J. Luo and M. Shah, "Recognizing realistic actions from videos “in the wild”," in *IEEE Conference on Computer Vision and Pattern Recognition(CVPR '9),* 2009, pp. 1996-2003.

[45] P. Over, G. M. Awad, J. Fiscus, B. Antonishek, M. Michel, A. F. Smeaton, W. Kraaij and G. Quénot, "TRECVID 2010–An overview of the goals, tasks, data, evaluation mechanisms, and metrics," 2011.

[46] A. Altadmri and A. Ahmed, "A framework for automatic semantic video annotation," *Multimedia Applications and Tools,* vol. 64, pp. 1-25, 2013.