Video Matching Using DC-image and Local Features

Saddam Bekhet, Amr Ahmed and Andrew Hunter

***Abstract*— This paper presents a suggested framework for video matching based on local features extracted from the DC-image of MPEG compressed videos, without decompression. The relevant arguments and supporting evidences are discussed for developing video similarity techniques that works directly on compressed videos, without decompression, and especially utilising small size images. Two experiments are carried to support the above. The first is comparing between the DC-image and I-frame, in terms of matching performance and the corresponding computation complexity. The second experiment compares between using local features and global features in video matching, especially in the compressed domain and with the small size images. The results confirmed that the use of DC-image, despite its highly reduced size, is promising as it produces at least similar (if not better) matching precision, compared to the full I-frame. Also, using SIFT, as a local feature, outperforms precision of most of the standard global features. On the other hand, its computation complexity is relatively higher, but it is still within the real-time margin. There are also various optimisations that can be done to improve this computation complexity.**

***Index Terms*—Video matching, DC-image, Video similarity, SIFT, Compressed domain.**

I. INTRODUCTION

THE volume of video data is rapidly increasing, more than 4 billion hours of video are being watched each month on YouTube and more than 72 hours of video are uploaded to YouTube every minute [1], and counters are still running fast. This is attributed to recent advances in technology, cheap digital cameras and the madness of web streaming either for personal or advertising purpose. The majority of available video data exists in compressed format MPEG-1, MPEG-2 and MPEG-4. Extracting low level features from compressed videos, without decompression, is still an open issue and has not been efficiently resolved. Extraction of low level features, directly from compressed domain, is the first step towards efficient video content retrieval. Such approach avoids expensive computations and memory requirement involved in decoding compressed videos. Working on compressed videos is beneficial because they are rich of additional, pre-computed, features such as DCT coefficients, motion vectors and Macro blocks types. DC coefficients specifically could be used to reconstruct a video frame with minimal cost.

Manuscript received March 23, 2013; revised April 03, 2013. This work is funded by Egyptian government as a PhD grant to the first author.
Saddam Bekhet, is an assistant lecturer at South Valley University ,Egypt, and a PhD student at School of Computer Science, University of Lincoln, Brayford Pool, Lincoln, LN6 7TS (e-mail: sbekhet@lincoln.ac.uk).
Amr Ahmed and Andrew Hunter are Faculty Members of School of Computer Science, University of Lincoln, Brayford Pool, Lincoln, LN6 7TS (e-mails: aahmed@lincoln.ac.uk , ahunter@lincoln.ac.uk).

However, most of the current techniques still inefficient in directly handling compressed videos, without decompressing them first. This becomes more crucial aspect for smart phones and tablets, where memory and speed are more strict constraints. So, new video similarity techniques need to be imposed to work in such environment; limited memory, processor and with compressed formats. All those advantages of detecting similarity from compressed videos are also expected to contribute to other higher-level layers of semantic analysis and annotation of videos, among other fields. An MPEG-X video consists of **"I", "P"** and **"B"** frames encoded using Discrete Cosine Transform (DCT) [2]**.** The DCT algorithm works by dividing an input image into 8*8 blocks (default block size). For each block, the DCT is computed and the result consists of one DC coefficient and 63 AC coefficients per block. A DC-image of an I-frame is the collection of all its DC coefficients**,** in their corresponding spatial arrangements. The DC image is 1/64 of its original I-frame size. Figure 1 shows an illustration of the DCT block structure. Figure 2 depicts samples of DC images reconstructed from I-frames.

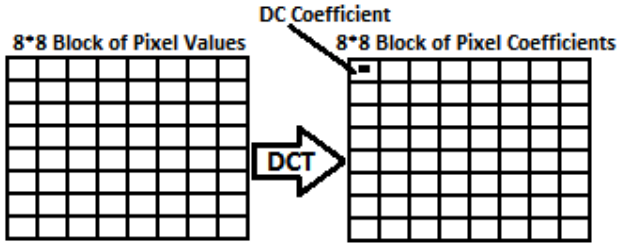

Fig. 1. DCT block structure.

Fig. 2. Sample DC-images of size 40*30 reconstructed from 320*240 I-frames, (A grayscale DC image results from an entirely grayscale video).

The DC-image is usually an image of size around 40 x 30 pixels. We, as humans, do not use it too much because of the availability of the full size image. However, a DC-image was found to retain most of the visual features of its original full I-frame. It has also been found that human performance on scene recognition drops by only 7% when using small images relative to full resolution images [3], as depicted in figure 3. This is very useful for computer vision algorithms, especially in relation to computation complexity of achieving the same task on the full size image. Taking advantage of the tiny size, timeless reconstruction and richness of visual content, the DC-image could be employed effectively alone or in conjunction with other compressed domain features (AC coefficients, macro-block types and motion vectors) to detect similarity between videos for various purposes; as automated annotation [4] or copy detection or any other higher layer built upon similarity between videos.

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

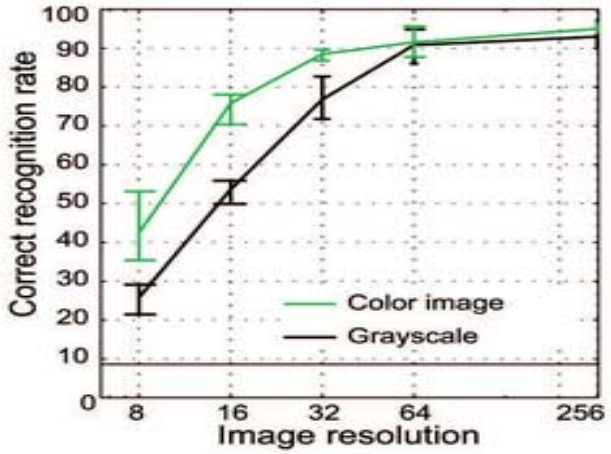

Fig. 3. Human performance on scene recognition as a function of image resolution [3].

II. RELATED WORK

In this section, we review key previous work related to the video analysis, and especially video similarity and matching, in the compressed domain. However, as the DC-image, once extracted, is a small or lower-resolution image, the relevant work on low-resolution small images will also be discussed. Initially the term "tiny image" was introduced in [3] during an attempt to construct a database of 80 million colored tiny images of size 32x32 collected over eight months from various search engines. Each image was loosely labeled with one of the 75,062 non abstract English nouns, as listed in the *Wordnet* lexical database [5]. The aim of this work was to perform object and scene recognition by fusing semantic information from *Wordnet* with visual features extracted from the image, using nearest neighbor methods. Image similarity was computed upon two measures, the first similarity measure is the sum of squared differences (SSD), over the first 19 principal components of each image. The second similarity measure accounts for the potential small scaling (horizontal mirror; translations and scaling up to 10 pixels) and small translations (within a 5x5 window), by performing exhaustive evaluation of all possible image shifts. The concept of tiny image was then adopted and extended in [6], [7], [8] who attempted to build a database of tiny videos. Approximately 50,000 tiny videos were used, in conjunction with the 80 million tiny images database, to enhance object retrieval and scene classification. Videos were collected from YouTube over a period of three weeks with all their metadata (e.g. title, description and tags). All video frames were resized to 40x30 pixels. Each frame was then converted to a one dimensional vector by concatenating all the three color channels. Similarity measures between videos were adopted from those used in tiny images [3]. In their approach, videos are treated as a bag of frames. Later the work was extended for the purpose of frame categorization. *Wordnet* was used to give initial voting for categorization purpose, based on available videos metadata. In their last paper [6] tiny videos were tested for video retrieval and classification. Each video was represented by a set of key frames which is used later for computing similarity between their respective videos. Key frames are chosen based on the intensity of motion frame sampling, which is computed over pixel level of consecutive frames. However, they utilized the available video tags, which is not always available. So, our work is more focused on videos before they have any tags or meta-data available which can be seen as a phase that can help in building such datasets for later use.

In the compressed domain, the DC-image has been used widely in shot-boundary detection and video segmentation due to its small size [9], [10], [11], [12], [13]. It was also utilized for keyframe extraction, instead of parsing the full frame to detect Keyframes [14], [15], [16], or even for video summarization purpose [17], [18]. For video retrieval, in [19] the DC-image was used to detect Keyframes, then attention analysis, is carried out on full I-frames, to detect salient objects. SIFT [20] is applied to detect interest points and track them in successive spatial salient regions and to build trajectories through other frames. Features of each salient region include color and texture are extracted and compared among videos for video retrieval purpose. But this method fails when either the visual features of the foreground object is not distinct or when video background contain rich details as it will produce meaningless salient regions which is not distinctive for a given video. An approach to match video shots and cluster them into scenes proposed in [21], the idea was taking into account variable number of frames to represent a shot (instead of only one keyframe). They used color and luminance histograms computed for every DC or DC+2AC to measure similarity between frames, as they are performing frame to frame similarity. The separation between color and luminance histograms was just for confirming the correctness of the results .The choice of DC-image or DC+2AC image depends on frame size, for frame of size 320*240 DC-image is selected and for frame size of 160*120 DC+2AC is selected, this makes representative frame images are always of size 40*30 and contains sufficient information for extraction.

In [22] matching between video clips is done using signatures built by extracting color values (Y-U-V) from DC-images sequence to form three different quantized histograms per each frame. The similarity between two videos is computed using sliding window technique, trying to find the best set of matching frames using histogram intersection. The approach of ordinal measures were applied on DC-images of each color component (Y-U-V) separately to generate fingerprint features for each frame which are accumulated to form video signature for later video matching within large databases [23]. Dimitrova *et al.* [24] demonstrated using DC coefficients of (Y-U-V) components separately and Motion vectors to build signature for video retrieval. Signature is extracted from every video frame and concatenated to form full video signature. Hamming distance is used to measure distances and rank similar videos. In addition the sliding window technique was used to determine the set of frames to compute signature from, by computing qualitative difference between DC values and motion vectors for those frames, so there can be multiple signatures of the same video depending on the window position. We can find that they did use the DC-image as a set of numeric values leaving all the visual information behind, in addition the sliding window technique is slow as it applies an exhaustive search process to align two signatures together.

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

Now we can summarize the usage of DC image from previous literature; (1)Video shot and scene detection, (2)Keyframe extraction, (3)Video summarization, (4)Video sequence matching, (5)Video signature generation and (6)Video copy detection.

Our main focus is on video sequence matching and similarity. From that point of view, techniques that utilize the DC-image can be classified based on feature extraction level and feature types, this is depicted in figure 4. For feature extraction level, two levels of extraction exist in literature. First, frame sequence matching in which every frame in a video is being processed to extract low level features to be used later for retrieval or signature building. The Second type is more compact and tries to reduce the amount of features being extracted and processed by using keyframes only, instead of using all video frames. Both approaches have disadvantages as they ignore the temporal aspect of a video and handles video as a bag of still images. Moreover, window alignment techniques will be needed in this case, which is based on exhaustive search among frames to find the best set of matching frames among videos. Regarding video signature built on those approaches it will be large and includes redundant information as it will be concatenation of individual frame signatures which violates the compactness aspect of signature we are aiming to achieve. For Keyframe based schemes, there is no fixed selection criteria for those Keyframes which could be applied to all videos; some techniques uses the first and last frames within a shot as key frames other uses the middle frame so, the resultant video signature may differ for same video with different Keyframe selection criteria.

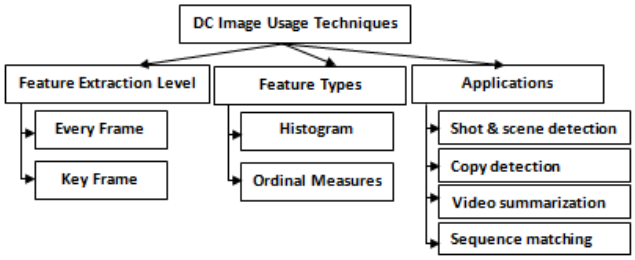

Fig. 4. DC images usage techniques.

For feature types which could be extracted from DC-image, there are: **Histogram (gray-scale, color)** [25] which is a global feature computed on frame level or video level (less common), in addition it could be built individually for each color component within frame depending on color space (YUV, HSV, RGB) where the similarity between videos depends on the similarity of underlying histograms and could be measured using histogram intersection or any other distance measurement function. Disadvantages of histograms are: (1)Relatively high computational cost (pixel level processing), (2)Not all video material has chrominance components; example IBM commercials videos are gray scale videos. This forbids constructing more discriminative histograms, (3)Highly dependent on underlying color space (HSV, RGB, YUV) as each of them exhibit significant variations in color representation, (4)The histogram, as a global feature does not capture spatial layout information, (5) More ever, two differently images in content are likely to have quite similar histograms, (6)Histograms cannot be used to describe specific object within video and (7)Histograms are highly dependent on frame size.

Ordinal Measures [26] is also a global feature. It was originally used for image matching and later adopted for video retrieval purpose. The idea works by partitioning an image into equal-sized blocks, which makes the image independent of its size, in other words two different size frames will have same number of ordinal measures, then sub images are ranked based on average color. Ordinal measures are invariant to luminance change and histogram equalization but within frame level, but it is not invariant to geometric transformations. Recently it has been extended to capture the temporal dimension of videos, as the blocking process could be done across video frames [27]. Ordinal measures are based only on color information which is not robust against color format change. In addition it does not capture any spatial information across video frames.

III. OUR PROPOSED APPROACH

In this section, our proposed DC-image based system for video similarity measurement is introduced. The proposed idea is to utilise local features, such as SIFT [20]**,** on the small DC-images *only* and track them across the video's I-frames (i.e. in the temporal dimension) to compare the similarity between videos. This introduces some challenges in extracting local features in such small images, as discussed later. Figure 5 below shows block diagram of our proposed system. The main stages of the system are:

1. Decoding the video and extracting the luminance DC-image sequence,
2. Extracting key points and their descriptors, in each DC-image.
3. Video matching, using the extracted features.

The following sub-sections describe those stages, including challenges and our contributions to facilitate the video matching on the small DC-images, without full decompression.

A. Extracting the DC image sequences

The process starts by decoding video and extracting luminance DC-images sequence from I-frames only. The reasons for focusing on the DC image only include is that:

- I-frame's DC-image is the quickest part that could be extracted from a compressed video without performing full decompression of video stream.
- I-frames in GOPs (Group Of Pictures) are inserted by the encoder when there is large residual change (residual is the amount of motion estimation error accumulated at the end of GOP), this could be analogous to key frames within a scene, in other words as a key frame is reprehensive to a scene, DC-image of an I-frame could be used as representative of a GOP. In addition, GOPs could be merged to specific length to limit number of I-frames DC-images and map them to be key frames like.
- I-frames will give about 10:1 initial compaction ratio assuming 10 frames per GOP [28] on average which means lower computations and faster results.
- Human eye is sensitive to small changes in luminance, but not in chrominance [3]. Thus we can discard the chrominance information and use luminance DC-image only.

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

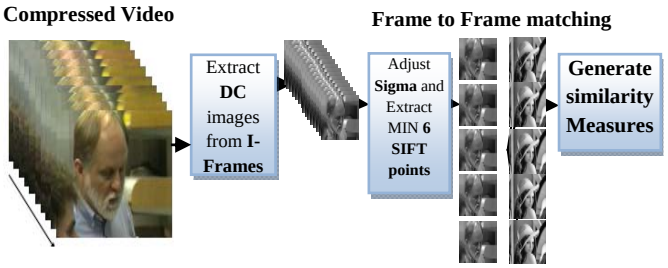

Fig. 5. Description of our proposed system to measure videos similarity based on DC image.

B. Extracting key points and descriptors

The second stage is extraction of key points and their descriptors. We used SIFT key points and descriptor because of its invariance nature to most of photometric and geometric transformations applied on image level. A typical full image of size 500*500 could generate more than 500 interest points. However, most of the DC-images would generate less than three SIFT key points, which is not enough for matching [20]. To work on the DC-image, which is of small size, we iteratively adjust the SIFT detector (through the sigma value; the amount of Gaussian blurring applied to an image) to be able to generate a minimum of six key points in each DC-image. Figure 6 and 7 below shows number of SIFT points per frame before and after our adjustment and enforcing the minimum of six SIFT key points.

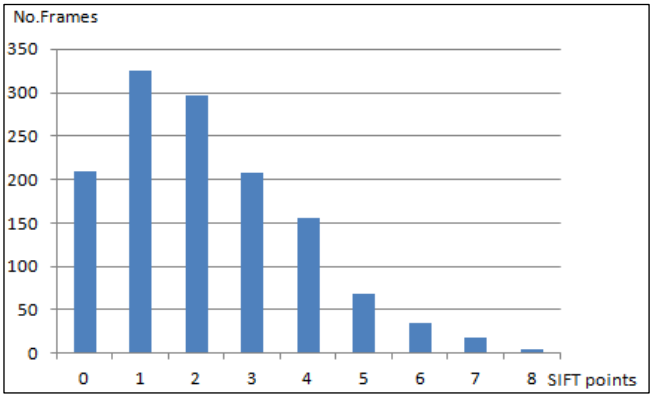

Fig. 6. Number of SIFT points per video frame, using the default SIFT.

With the enforcement of a minimum number of SIFT key points per DC-image, we facilitated for the DC- image to be used for video matching. The precision of matching, using DC-images compared with the full frame is discussed later in section IV.

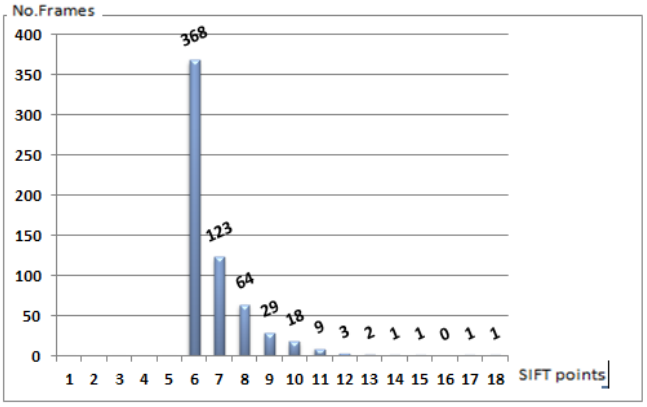

Fig. 7. Number of SIFT points per video frame; with enforcing a minimum of six SIFT points.

C. Video matching using features extracted from the DC images.

The third and final stage is the matching between videos. For simplicity, we adapted the frame-to-frame matching with dynamic programming approach. For each video pair, we compute a similarity measure that takes into account the temporal sequence of frames. This is done by searching for the longest matching sequence of frames between the two videos. To compute this matching value we used dynamic programming to find the optimal matching sequence of video frames. Optimality in this case means finding the best matching video frames (taking in to account the temporal order of frames) that maximizes the overall similarity score. Figure 8 shows a confusion matrix of a given two videos and the optimal matching values between their respective frames are highlighted in yellow. Following is the pseudo-code of the dynamic programming algorithm we use:

SET **M** to "Number of frames of the first video"+1;
SET **N** to "Number of frames of the second video"+1;
CREATE_MATRIX **OPT_MATCH[M][N]**;
INITALIZE **DISTANCE** matrix to all frame-to-frame similarity based on extracted SIFT features;
SET **OPT_MATCH** to 0;
FOR **I**=1 to **M** DO
  FOR **J**=1 to **N** DO
{
SET **MX** to MAX of (**OPT_MATCH[I-1][J-1]**+
  DISTANCE[I-1][J-1] AND **OPT_MATCH[I][J-1]**);
SET **OPT_MATCH[I][J]** to MAX of(**MX** AND **D[I-1][J]**);
}
RETURN **OPT_MATCH[M-1][N-1]** ;

Where **DISTANCE** is the confusion matrix between both video frames based on the number of matched SIFT points, and **OPT_MATCH** is the matrix which will contain the finial matching value , this value will be located in **OPT_MATCH[M-1][N-1]**,and MAX is function returns the maximum value of a given two numbers. The algorithm works by scanning the confusion matrix from left to right and up to bottom trying to find the highest match for each frame taking into account the previous frame match , next frame match and the temporal order of frames.

		Video 1																
		F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12	F13	F14	F15	F16	F17
Video 2	F1	66	83	50	33	83	66	50	50	0	33	83	66	66	66	83	83	66
	F2	50	66	50	33	50	50	33	66	0	50	50	66	66	50	90	66	50
	F3	50	66	50	33	50	50	33	50	16	33	50	50	50	50	66	50	33
	F4	42	42	42	28	42	28	28	57	28	42	28	42	42	42	42	42	28
	F5	66	66	50	33	50	50	50	50	50	50	50	50	66	33	66	50	33
	F6	66	66	33	16	50	50	50	50	33	50	50	66	50	33	66	50	33
	F7	28	57	71	57	28	57	42	57	57	57	71	71	57	4	57	57	57
	F8	42	42	57	71	28	42	28	57	57	57	42	42	28	28	42	42	42
	F9	33	50	66	50	33	50	50	50	50	50	50	33	33	50	50	50	50
	F10	66	66	83	83	66	66	66	66	50	66	66	50	50	66	66	50	50
	F11	66	50	66	83	50	50	66	66	33	50	50	33	33	50	50	33	50

Fig. 8. Finding matching similarity score between two videos.

We can see that our dynamic programming algorithm performs one to one mapping, means each frame will be matched to only one frame in other video. As the algorithm searches for the best matching relationships between videos

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

frames, some frames may not be matched, as they will reduce the overall matching value between videos (e.g. frames 1, 4 in video 1) and others won't be matched in case of matching two videos with different number of frames (e.g. frames 6, 7, 13 and 16 in video 1).

IV. Experiments and Results:

In this section we explain the experiments and present the results that support our work explained above. This section contains two experiments. The first is comparing between the DC-image and I-frame, in terms of matching performance and the corresponding computation complexity. The second experiment compares between using local features and global features in video matching, especially in the compressed domain and with the small size images. The First dataset used in our experiments consists of 100 videos adopted from [29] and divided into 34 boats videos, 36 cars videos and 30 tank videos. All the videos in the dataset were re-encoded to be of fixed height and width (352x240 pixels), so that all the DC-images are of equal size (44 x 30 pixels). The experiments ran on Intel Core i3- 3.30 GHZ computer with 4 Gigabytes of RAM.

A. DC-image vs. I-Frame

The purpose of this experiment is to evaluate the performance of the DC-image, in terms of matching and computational complexity, compared to the corresponding I-frame. The experiment used the framework explained in figure 5 and the dataset described earlier. Matching using the DC-image using SIFT [20] features took a total of 58.4 minutes for all videos in the dataset, while it took a total of 166.6 hours for the same dataset using the full I-frame. The average time (per frame) is 0.017 seconds for the DC-image, compared to 1.05 seconds for the I-frame (time includes reconstruction, SIFT key points extraction and matching). This shows that the computation complexity of using the DC-image is only 1.6% of the corresponding I-frame, which means a total reduction of 98.4% in processing time. Figure 9 shows the timing details for the DC-image and the I-frame image. To compare the matching performance, the retrieval precision was calculated over the top 1, 5 and 10 retrieved results. The DC-image, despite its highly reduced size**,** was found to have a slightly higher precision than the I-frame; with 13% for the top 1 retrieved result. Figure 10 depicts the precision details for the DC-image versus the I-frame **i**mage**.**

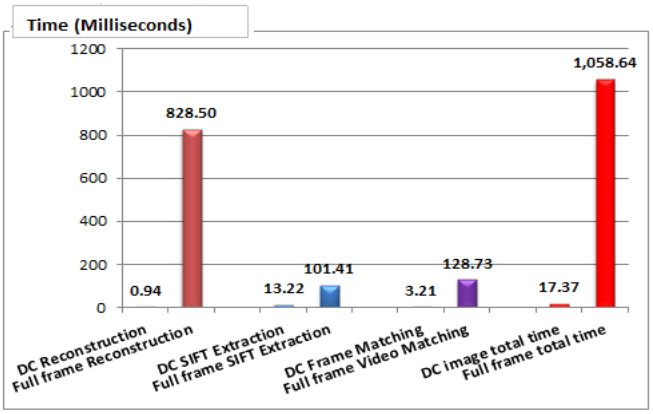

Fig. 9. DC versus I-frame timing performance.

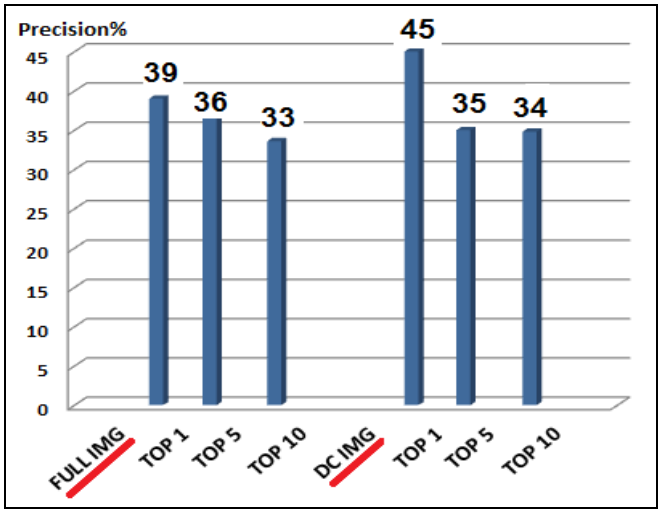

Fig. 10. DC image versus I-Frame retrieval precision.

B. Local vs. Global features

The purpose of this experiment is to evaluate the performance of using local and global features, on the DC-image, in terms of matching precision and computational complexity. The experiment used the framework explained in figure 5, and the dataset described earlier. For local features, we applied matching based on SIFT [20] as a local feature descriptor. For global features, we applied matching based on the luminance histogram, ordinal measures and finally the pixel difference [7].

The results, presented in figure 11, shows that SIFT [20] as a local feature descriptor outperforms global feature descriptors by 15.4% (compared to ordinal matching as the highest global feature method). However, SIFT's computation complexity was the highest, as depicted in figure 12. SIFT [20] took 16.43 milliseconds to match two frames, compared to only 2 milliseconds in pixel difference match (maximum time in case of global features). But SIFT is still within the real-time margin, while producing better matching performance. Even we are using all measures in its generic form so that none of them have advantage over the others; in addition none of them is taking the temporal dimension of video into account as it is computed based on the finial frame-to-frame confusion matrix, unless at the end for finding the finial matching value using dynamic programming techniques for all approaches.

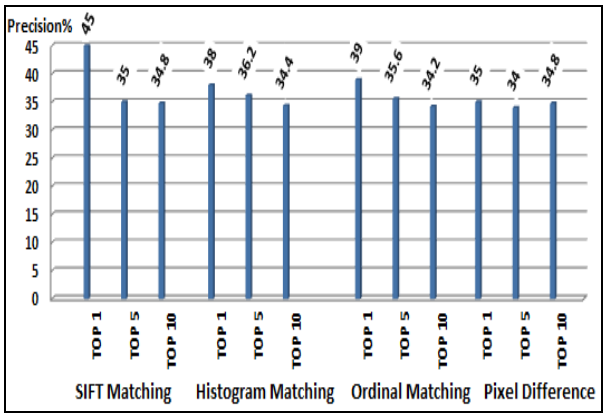

Fig. 11. DC image retrieval precision using SIFT, Luminance Histogram, Ordinal Measures and Pixel difference (Dataset from [29]).

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

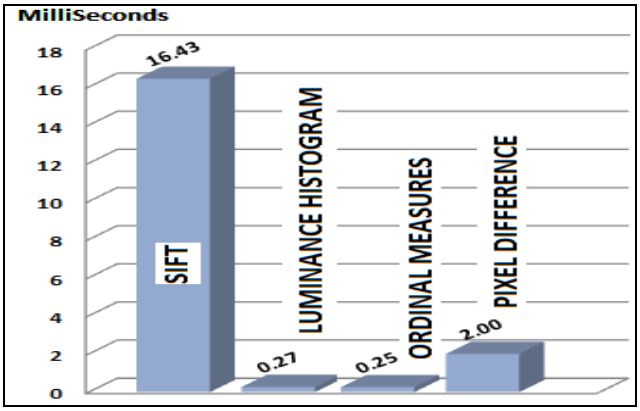

Fig. 12. DC timing analysis using different matching techniques SIFT, Luminance Histogram, Ordinal Measures and Pixel difference (Dataset from [29]).

V. CONCLUSION AND FUTURE WORK

In this paper, we presented a framework for video matching based on local features extracted only from the DC-image of MPEG compressed videos, without decompression. In addition, we discussed the relevant argument and evidences for video similarity techniques that works directly on compressed videos, without decompression, and especially utilising small size images. Two experiments were carried to support the above. First, we compared the DC-image with its full I-frame in terms of both the similarity precision and the computational complexity. We had to address the issue of using SIFT on such small-size images, before it can be used. The results show that using the DC-image, despite its small size, produces similar (if not better) similarity precision to the use of its corresponding I-frame. But using the DC-image has dramatically improved the computational performance, which make it a high candidate. Second, local features, such as SIFT, were compared to standard global features for the purpose of video similarity. The results shows that using SIFT, on DC-image only, slightly outperformed the accuracy of the global features. On the other hand, the computational complexity of using SIFT is relatively higher than those for the global features. But SIFT extraction and matching are still within the real-time margins, and we still have a number of optimisations to introduce to improve this computation complexity. We also plan to introduce more complex matching, instead of the Frame-to-Frame approach, and better incorporate the temporal information actively.

REFERENCES

[1] YouTube Statistics, Available in March 2013 http://www.youtube.com/yt/press/statistics.html

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

[3] A. Torralba, R. Fergus and W. T. Freeman, "80 Million tiny images: A large data set for nonparametric object and scene recognition," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 30, 2008, pp. 1958-1970.

[4] A. Altadmri and A. Ahmed, "A framework for automatic semantic video annotation," *Multimedia Applications and Tools*, vol. 64, no. 2, 2013, pp.1-25.

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

[6] A. Karpenko and P. Aarabi, "Tiny videos: a large data set for nonparametric video retrieval and frame classification," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, 2011, pp. 618-630.

[7] A. Karpenko and P. Aarabi, "Tiny videos: a large dataset for image and video frame categorization," in *Multimedia, 2009. ISM '09. 11th IEEE International Symposium on*, 2009, pp. 281-289.

[8] A. Karpenko and P. Aarabi, "Tiny videos: non-parametric content-based video retrieval and recognition," in *Multimedia, 2008. ISM 2008. Tenth IEEE International Symposium on*, 2008, pp. 619-624.

[9] Boon-Lock Yeo and B. Liu, "A unified approach to temporal segmentation of motion JPEG and MPEG compressed video," in *Multimedia Computing and Systems, 1995., Proceedings of the International Conference on*, 1995, pp. 81-88.

[10] J. Meng, Y. Juan and S. F. Chang, "Scene change detection in a MPEG compressed video sequence,"in *IS&T/SPIE Symposium Proceedings*, 1995.

[11] Jin-Long Zheng, Ming-Jun Li, Ming-Xin Zhang, Jian Zhou and Zai-De Liu, "An effective framework of shot segmentation based on I-frame in compressed-domain videos," in *Wavelet Analysis and Pattern Recognition, 2012 International Conference on*, 2012, pp. 84-90.

[12] P. Xu, L. Xie, S. F. Chang, A. Divakaran, A. Vetro and H. Sun, "Algorithms and system for segmentation and structure analysis in soccer video," in *Proc. ICME*, 2001, pp. 928-931.

[13] A. Divakaran, H. Ito, H. Sun and T. Poon, "Scene change detection and feature extraction for MPEG-4 sequences," in *Electronic Imaging'99*, 1998, pp. 545-551.

[14] Guozhu Liu and Junming Zhao, "Key frame extraction from MPEG video stream," in *Information Processing (ISIP), 2010 Third International Symposium on*, 2010, pp. 423-427.

[15] Eung Kwan Kang, Sung Joo Kim and Joon Soo Choi, "Video retrieval based on scene change detection in compressed streams," Consumer *Electronics, IEEE Transactions on*, vol. 45, pp. 932-936, 1999.

[16] O. N. Gerek and Y. Altunbasak, "Key frame selection from MPEG video data," in *Electronic Imaging'97*, 1997, pp. 920-925.

[17] J. C. S. Yu, M. S. Kankanhalli and P. Mulhen, "Semantic video summarization in compressed domain MPEG video," in *Multimedia and Expo, 2003. ICME '03. Proceedings. 2003 International Conference on*, vol.3, 2003, pp. III-329-32.

[18] J. Almeida, N. J. Leite and R. d. S. Torres, "Online video summarization on compressed domain," Journal *of Visual Communication and Image Representation*, 2011.

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

[20] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," International *Journal of Computer Vision*, vol. 60, 2004, pp. 91-110.

[21] M. M. Yeung and Bede Liu, "Efficient matching and clustering of video shots," in *Image Processing, 1995. Proceedings., International Conference on*, vol.1, 1995, pp. 338-341.

[22] 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. SPIE, Storage and Retrieval for Media Databases*, 2000, pp. 564-572.

[23] R. Mohan, "Video sequence matching," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 6, 1998, pp. 3697-3700.

[24] N. Dimitrova and M. S. Abdel-Mottaleb, "Video retrieval of MPEG compressed sequences using dc and motion signatures," Video *Retrieval of MPEG Compressed Sequences using DC and Motion Signatures*, 1999.

[25] R. C. Gonzalez and E. W. Richard, *Digital Image Processing.* , 3rd ed. Prentice Hall, 2008, pp. 142-143.

[26] D. N. Bhat and S. K. Nayar, "Ordinal measures for visual correspondence," in *Computer Vision and Pattern Recognition, 1996. Proceedings CVPR '96, 1996 IEEE Computer Society Conference on*, 1996, pp. 351-357.

[27] J. Almeida, N. J. Leite and R. da S Torres, "Comparison of video sequences with histograms of motion patterns," in *Image Processing (ICIP), 2011 18th IEEE International Conference on*, 2011, pp. 3673-3676.

[28] D. Fuentes, R. Bardeli, J. A. Ortega and L. Gonzalez-Abril, "A similarity measure between videos using alignment, graphical and speech features," *Expert Syst. Appl.*, vol. 39, 2012, pp. 10278-10282, 9/1.

[29] A. Basharat, Y. Zhai and M. Shah, "Content based video matching using spatiotemporal volumes," Computer. *Vision Image Understanding*, vol. 110, 2008, pp. 360-377.

ISBN: 978-988-19252-9-9
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)