Multimedia Tools and Applications
DOI 10.1007/s11042-015-2887-8

Compressed Video Matching: Frame-to-Frame Revisited.

Saddam Bekhet · Amr Ahmed · Amjad Altadmri · Andrew Hunter

Received: 19 January 2015 /Revised: 2 July 2015 / Accepted: 12 August 2015

Abstract This paper presents an improved frame-to-frame (F-2-F) compressed video matching technique based on local features extracted from reduced size images, in contrast with previous F-2-F techniques that utilized global features extracted from full size frames. The revised technique addresses both accuracy and computational cost issues of the traditional F-2-F approach. Accuracy is improved through using local features, while computational cost issue is addressed through extracting those local features from reduced size images. For compressed videos, the DC-image sequence, without full decompression, is used. Utilizing such small size images (DC-images) as a base for the proposed work is important, as it pushes the traditional F-2-F from off-line to real-time operational mode. The proposed technique involves addressing an important problem: namely the extraction of enough local features from such a small size images to achieve robust matching. The relevant arguments and supporting evidences for the proposed technique are presented. Experimental results and evaluation, on multiple challenging datasets, show considerable computational time improvements for the proposed technique accompanied by a comparable or higher accuracy than state-of-the-art related techniques.

This work is funded By SouthValley University- Egypt.

S. Bekhet
University of Lincoln
E-mail: sbekhet@lincoln.ac.uk

A. Ahmed
University of Lincoln
E-mail: aahmed@lincoln.ac.uk

A. Altadmri
University of Kent
E-mail: amjad.altadmri@gmail.com

A. Hunter
University of Lincoln
E-mail: ahunter@lincoln.ac.uk

Keywords F-2-F Matching · Compressed Domain · Local Features · Trajectories · SIFT · MPEG · DC-image

1 Introduction

The number of publicly available videos (especially compressed videos)[1] has enormously increased due to the proliferation in multimedia recording technologies and the exponential growth of storage mediums. Handling such enormous video numbers forced researchers to develop efficient matching techniques for various applications (e.g. searching and annotation). However, the majority of current video matching techniques were originally developed for uncompressed videos, and later adopted for matching compressed videos, through decompressing videos first. Such decompression is waste of processing time and does not match current real-time demands. One of the earliest and simplest uncompressed video matching techniques is the frame-to-frame (F-2-F) approach[21]. F-2-F operation was influenced by the image matching discipline, treating videos as a group of frames. Hence, the task is trying to find the best set of matching frame pairs based on features extracted from their respective full size frames. Only global features were used as they are computationally cheaper to extract than local features. On the other hand compressed videos (e.g. MPEG) offers a diverse set of pre-computed features (e.g. DC coefficients, macro block types) which are quite useful in revising traditional techniques such as F-2-F. Specifically, the DC-image of an I-frame, which defined as; the collection of all DC coefficients results from applying DCT transform on I-frame for MPEG compression purpose[23]. This tiny image $\sim (40 \times 30)$ pixels is an important compressed domain feature of size 1/64 of its respective full I-frame, which could significantly reduce the algorithms computational time.

In this paper, the traditional F-2-F video matching technique is revised and improved to be able to work on compressed videos directly avoiding the lengthy decompression process. Previous F-2-F techniques used only global features from full size decompressed frames, since it is cheaper than extracting local features from respective full size frames, especially with the added decompression time. Furthermore, it was difficult to extract local features from the tiny size DC-images, as they are weak and do not generate sufficient features for matching[17]. Thus, the revised technique takes advantage of local features, extracted directly from compressed stream DC-image sequences, by proposing an adaptive way to generate enough local features. This change allows the F-2-F to be used in real-time matching as a simple video matching technique, compared to a more complex techniques (e.g. trajectories). In addition, the revised technique is validated against state-of-art baselines based on multiple challenging datasets.

This paper is organized as follows: section 2 presents the literature review; the proposed work is presented in section 3; the experiments and analysis are presented in section 4; section 5 concludes the paper.

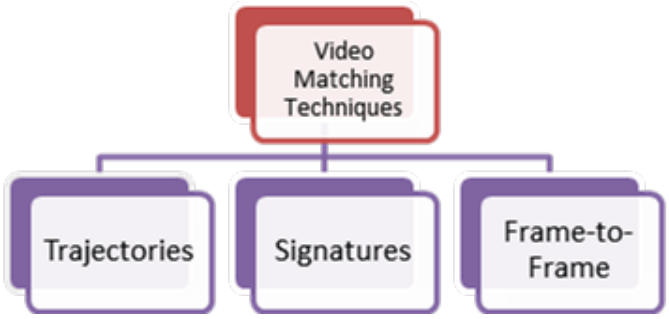

Fig. 1 various types of video matching techniques.

2 Literature review

In this section, we review key previous work related to video matching, with emphasis on the compressed domain. Generally, video matching techniques may be classified into three main categories: trajectories, signature based and F-2-F, as depicted in Fig.1. In trajectories, local features are utilized to track object/keypoint positions across video frames[7]. Trajectories usually describe the movement of salient objects[12], keypoints[7], or spatio-temporal volumes[12]. Although trajectories achieved reasonable matching results for uncompressed videos, its major drawback is the excessive computational cost needed to extract and track local features. For example, a typical image of size 500×500 could generate more than 2000 SIFT local keypoints [17], which is a huge number of keypoints to be tracked and filtered across all video frames. Furthermore, the added decompression time, when dealing with compressed videos, can reach 40% of CPU time[2]. This problem was addressed in[11] by utilizing motion vectors instead of keypoints, as they are pre-computed during the MPEG encoding process. However, obtaining the motion vectors still requires partial decoding; especially as they are not available for I-frames. Regarding signature matching, a generic utilization of motion vectors in conjunction with DC coefficients was introduced in[10], where the aggregation of both values in a one dimensional array used as a video signature. Moreover the actual matching is done using the sliding window technique, by computing direct difference between adjacent signature vectors for the currently matching frames pair. However, the sliding window involves exhaustive search and matching for every possible frame pair which dramatically effects the performance. In addition, the aggregation of DC coefficients in vector format does not benefit from any available visual information in such DC-image. A different signature was designed by Hua et al.[13] that utilize ordinal measures extracted from fully decompressed video frames. However, it does not suit real-time processing nor does it benefit from any MPEG features. This problem was tackled by Almeida et al.[5] by using I-frame's DC coefficients to compute the ordinal measures, since they are a pre-computed averages of their respective blocks. They implemented a motion histogram signature by computing tempo-

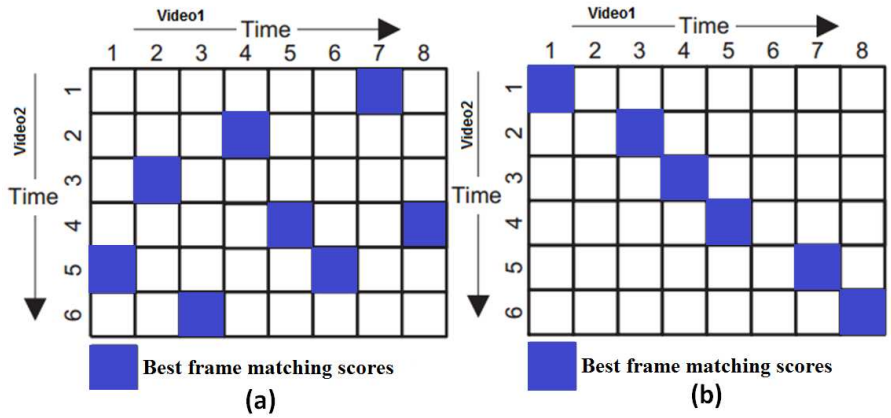

Fig. 2 Types of F-2-F matching, (a) unordered (b) ordered. In case of unequal video lengths, some frames will not be matched or might be matched to more than one frame depending on the matching algorithm[19]. The unordered matching is not common to be used as it discards the temporal order of video frames.

ral and spatial ordinal matrices for each I-frame. Both matrices are combined to form a normalized 6075 floating-point bin histogram, which is a quite large signature for matching in large databases. The important issue about video signatures is the type of features and the compactness of the signature. This is a critical issue as most of the extracted video features are high dimensional vectors, which makes it difficult to encode them into a compact signature. Thus, further research is still needed to improve the matching and/or feature descriptors.

The notion of a video "frame" in computer vision has been and still a major driving force for developing video matching techniques. F-2-F[19,21] is one of the earliest video matching techniques, borrowed from image matching discipline. The technique itself is very simple and intuitive, as it tries to match video frames as a pair of images. It mainly depends on the underlying features used for matching frame pairs. Throughout the literature, F-2-F has always been associated with global features e.g. color histogram[19] and pixel difference[14], since they could be computed faster than local features at full image resolution, Fig.2 depicts two different modes of using F-2-F technique (a) unordered matching that attempts to match frame pairs regardless of their temporal order,(less commonly used). Fig.2.b shows ordered matching that attempts to keep frames temporal order while matching them. Although F-2-F approach has a limited applicability on compressed videos due to its high computational cost, it was introduced in[9] as a part of a framework used to compare the DC-image versus the full frame performance. But it was tested on limited size datasets without reporting any comparisons with other baselines. Moreover, the technique itself has a lot of issues that need to be handled carefully as they affects its overall performance. First, the alignment problem

that arises in case of matching videos of different lengths[4], as the algorithm needs to decide the set of frames to be skipped during matching, without reducing the overall matching cost (see Fig.2.b, an 8 frames video is matched with a 6 frames video). Second, the selection of good matching criteria for the underlying frame pairs. Third, the dependency on global features for underlying frame matching while local features are more stable[15] and achieve better results[15,8,9]. Fourth, the time needed to apply this technique on full size frames, which represents an obstacle for real-time matching, especially with the additional decompression overhead. Discussions of those points along with the revised F-2-F technique are introduced in the next section.

3 The proposed F-2-F matching of compressed videos

This section introduces the revised F-2-F technique with the relevant arguments and supporting evidences. The inadequacy of F-2-F as a matching technique has been due to two main issues: computational cost (time) and accuracy. Time is related to the large number and size of video frames to be matched, while accuracy is related to using global features. Additionally, when dealing with compressed videos, extra computational overhead is introduced due to decompression. Hence, we propose to operate on compressed videos directly, without decompression, with reduced number of frames and reduced frame size. To reduce the number of frames, we propose to process I-frames only. For frame size, we propose to process the DC-image only, which is 1/64 of full I-frame size. Regarding the accuracy we propose to utilize local features instead of global features, since it is reported to be more robust[15,8]. Thus, the revised algorithm will be compared with the original F-2-F(global features on full frames) to verify the time improvement, and compared to recent baselines to verify accuracy improvement. Next sections investigate and present solutions for the proposed improvements in the revised F-2-F approach.

3.1 Local feature extraction in small images

We investigated the use of SURF and SIFT as local feature detectors/descriptors. However, due to the nature of the DC-image as a small image, extraction of those local features is problematic. It was reported that a minimum of three keypoints is needed for robust matching [17]. However, most of the DC-images produce less than three keypoints. As depicted in Fig.3.b and Fig.3.a, for SURF and SIFT respectively 62% of DC-images generates less then three keypoints, which is not sufficient for reliable matching. To overcome this issue, and facilitate for matching, we adapted SIFT to generate sufficient local features. SIFT was chosen because it was reported to be more robust than SURF[18]. Knowing that the effectiveness of SIFT is based on finding the most stable keypoints across different scale spaces using Gaussian function[17]. Equation1 shows the variable-scale Gaussian function, where sigma (σ) is the standard

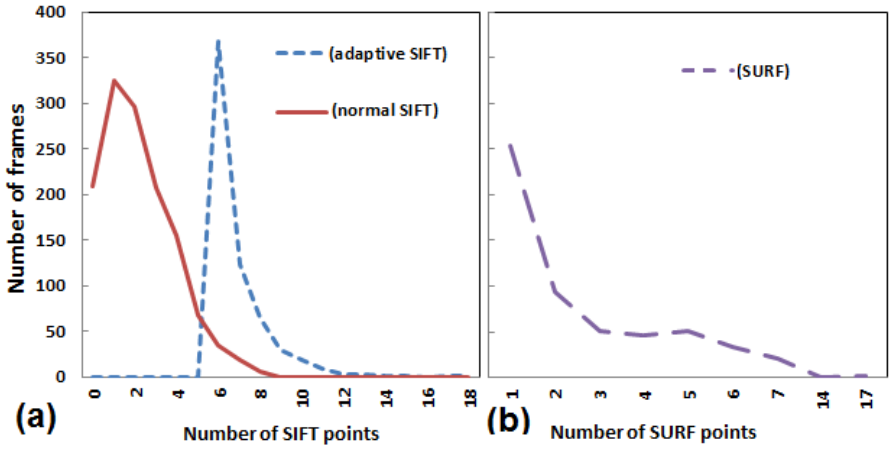

Fig. 3 SIFT/SURF local features extracted from DC-image (a) SIFT points extracted before adaptation (normal SIFT): 62% of frames have less than 3 keypoints and after our SIFT adaptation (adaptive SIFT): 100% of frames have enough keypoints for matching and (b) SURF points extracted from DC-image :62% of frames have less than 3 keypoints. Results based on BBC Rushes[22].

deviation, that controls the amount of blurring applied at different scale spaces to identify possible keypoints locations. Thus, the adaptive SIFT works by iteratively attempting different sigma values, starting from the default (1.6), and decrementing by a factor of 0.1 until the required minimum number of SIFT keypoints is obtained.

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} \mathbf{e}^{-(x^2+y^2)/2\sigma^2} \tag{1}$$

This adaptation facilitated generating a minimum of six key points from each DC-image, as depicted in Fig.3.a for the adaptive SIFT curve. This is double of the required minimum to ensure robust and efficient matching. Regarding time cost of this process, the overall speed of the revised F-2-F (including sigma adjustment) is presented in Table.1. But its individual timing could be investigated as following: Knowing that the maximum time to find suitable sigma is 0.01 seconds (per a single DC-image), which is the worst case scenario, this is not frequently happening (witnessed in the experiments), and with a common 10 I-frames per video shot, the sigma adjustment cost will be 0.01×10=0.1 second. This worst case scenario still represents only ∼17% of the total matching runtime cost(0.56 seconds from Table1).

3.2 SIFT matching

After extracting enough keypoints, we start matching each possible DC-image pair from the matching videos. The distance between two SIFT keypoints

(Equation2) is calculated using the cosine angle method [17]:

$$\theta_{i,j} = \arccos \frac{\boldsymbol{x}_i.\boldsymbol{y}_j}{|\boldsymbol{x}_i|.|\boldsymbol{y}_j|} \tag{2}$$

Where $\boldsymbol{x}_i$, $\boldsymbol{y}_j$ are SIFT vectors and θ is the angle between them representing the similarity score. A given interest point is matched to the most similar point (in feature space) provided that the distance is significantly less than of the next nearest one[18]. This condition is satisfied by the following Equation3:

$$\theta_{i,j1} < \alpha\ \theta_{i,j2} \tag{3}$$

Where (α) is a coefficient determining how much nearer, in feature space, $\boldsymbol{x}_i$ must be close to $\boldsymbol{y}_{j1}$ than to $\boldsymbol{y}_{j2}$ to be a good match, and $\boldsymbol{j}1$ and $\boldsymbol{j}2$ denote the closest and second-closest matches respectively. (α) is set to 0.6 during the experiments, as it achieved the highest matching scores.

3.3 Dynamic programming

Following the previous step of and the construction of the underlying F-2-F similarity matrix, by matching every frame pair based on their respective local features. The next step is to use dynamic programming to obtain the optimal set of matching frames, taking into account the temporal sequence of frames (ordered matching, Fig.2.b). Previous techniques of ordered F-2-F, that used dynamic programming, were limited only to find the longest matching frames sequence across or near the diagonal of the frame similarity matrix[19]. Hence, we adapted a new version which is able to locate the longest matching subsequence regardless of its location within the similarity matrix (i.e. not only around the diagonal).

Algorithm 1 finding the optimal matching sequence of frame pairs.

Input: M =Number of frames in first video+1;
N = Number of frames in second video+1;
DISTANCE= frame-to-frame similarity scores based on matched SIFT features;
Operation:
1. CREATE MATRIX OPT_MATCH [M][N];
2. SET OPT_MATCH to 0;
3. FOR I=1 to M DO
4. FOR J=1 to N DO
5. SET OPT_MATCH [I][J] to MAX of (OPT_MATCH [I-1][J-1]+ DISTANCE[I-1][J-1]
AND DISTANCE[I-1][J] AND OPT_MATCH[I][J-1]);
6. END FOR
7. END FOR
Output: OPT_MATCH [M-1] [N-1];

Table 1 Proposed dynamic programming algorithm.

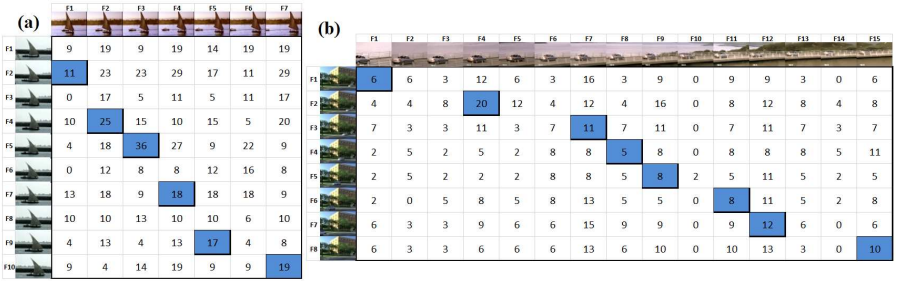

Fig. 4 Two real examples showing the underlying process of F-2-F technique in finding the optimal sequence of matching frames using dynamic programming approach, Horizontal axis is video1 and vertical axis is video2. Due to the difference in videos lengths, some frames might not be matched. For example, frame 6 in video1 in (a) and frames 2, 3, 5, 6, 10, 13 and 14 in video1 in (b). The proposed algorithm carefully skips frames with minimum effect on finial matching cost.

Table.1 shows the adapted dynamic programming algorithm, where OPT-_MATCH[M][N] is the underlying cost matrix used by the algorithm to keep track of the best matching frame pairs till the current matching position. Basically, the algorithm works by scanning the similarity matrix, row by row, trying to find the best match for each frame taking into account their respective temporal order and skipping frames with the lowest significance on the overall matching cost. After applying the algorithm the finial matching score between video shots is OPT_MATCH[M-1][N-1], which could be used later in retrieval. Furthermore, the exact set of matched frames could be extracted by backtracking through OPT_MATCH matrix starting from OPT_MATCH[M-1][N-1]. Fig.4 depicts a sample frame similarity matrices of a given two videos, with the optimal matching values between their respective frame pairs are highlighted.

In the next section we introduce the related experiments to quantitatively evaluate the performance, both time and precision wise. Also, the revised F-2-F technique is evaluated against previous F-2-F techniques, as well as other video matching techniques, and tested on various challenging bench mark data sets.

4 Experiments and Results

In order to examine and evaluate the performance of our revised approach, we tested against various datasets; (1)BBC RUSHES (335 videos)[22], (2)UCF11-(1600 videos)[16] and (3)Mixed dataset of BBC RUSHES and UCF11 datasets (300 videos). The first is a standard data set for video retrieval and contains a diverse set of challenging videos; mainly man-made moving objects (cars, tanks, planes and boats). The second is a standard dataset for action recognition and is widely used for retrieval purposes, as videos contain large variations

in object appearance, pose, scale as well as camera movement. The third is a mixture of BBC RUSHES and UCF11. The performance is evaluated using precision-over-N (P_N) standard measure[20](Equation4), following Leave-One-Out-Cross-Validation model (matching video to every other video in the dataset except itself). The computer used during the experiments is core i3 3.3 GHZ with 8Gbyte of RAM.

$$P_{Ni} = \sum_{j=1}^{q} \left(\frac{\sum_{r=1}^{N_i} rel(r)}{\sum_{r=1}^{N_i} (correct\ matches)} \right) \Big/ q \tag{4}$$

Where P_{Ni} is precision-over-N till rank i and N_i is the top-N matches for query video till rank i and q is the total number of queries being tested, and $Rel(r)$ is binary function defined as following in Equation5:

$$Rel(r) = \begin{cases} 1, & \text{item at rank(r) is correct match.} \\ 0, & \text{otherwise.} \end{cases} \tag{5}$$

P_N metric is selected, as the purpose of the revised F-2-F is retrieving a maximum set of top-N matches for a query video, this is similar to querying Google and having the best results in the first page, but the best one might be 2^{nd} or 5^{th} or so.). This specific evaluation is reflected in P_N metric which represents the relevance of the entire top-N results accumulatively. Thus the whole P_N curve need to be considered while comparing different P_N curves (not single P_N value at specific rank). Thus, during the following evaluation the percent(%) increase/decrease in total top-N matches would be reported as it would, emphasis the results.

The sequence of experiments in this section is as follows : section 4.1 investigates the effect of matching using DC-image compared to the full image based on global features. This evaluates and supports our claim for the real-time operational manner of the DC-image as a base for our work and proves the hypothesis of time improvement of the revised F-2-F compared to the main F-2-F(global features on full image). Section 4.2 examines matching using local features of DC-image versus its global features to emphasize the superiority of local features. Finally section 4.3 presents evaluation of the proposed work (F-2-F based on DC-image local features) against other baseline: namely trajectories[7] and the latest implementation of F-2-F technique[19], where comparing with baselines proves the hypothesis of accuracy improvement of the revised F-2-F.

4.1 DC-image vs. full image, using global features

The most important issue targeted by the proposed work is the computational cost issue. Earlier versions of F-2-F utilized global features(e.g. color

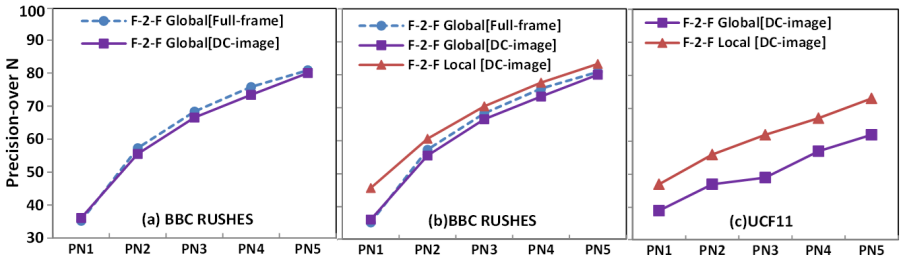

Fig. 5 (a) Matching precision for F-2-F based on global features considering the DC-image vs. full frame BBC Rushes, (b) F-2-F for DC-image and local features vs. global features for DC-image and full frame BBC Rushes, (c) F-2-F DC-image local features vs. global features UCF11.

histogram) of full size frames[19] which do not suit real-time, due to the processing of large sizes and numbers of frames. Using the DC-image is beneficial since it has highly reduced size (1/64 of I-frame) and could represent a given video in less frames (extracted from I-frames only). Following an experiment over BBC RUSHES datasets, it was found that the reduction in matching time using global features for the DC-image compared to the full size frame reached ∼95% as depicted in Table.2, while the performance (P_N) is lowered by 1.19% on average as shown in Fig.5.a. This is a relatively small number compared to the huge time reduction. This emphasizes the adequacy of the DC-image for faster processing with a slight effect on matching precision, which has been addressed and improved as discussed as following in section4.2.

4.2 DC-image local features vs. global features

This section examines the effect of local feature compared to global feature matching based on the DC-image. Fig.5.b and Fig.5.c depicts F-2-F (P_N) curves of local features versus global features (color histogram), based on the DC-image over two different datasets, BBC RUSHES and UCF11 respectively. It shows that local features significantly outperform global features, regarding DC-image, ($p<<0.05$), with 5.19% and 10.20% average higher P_N for BBC RUSHES and UCF11 respectively. More ever, the DC-image local features achieves almost comparable results to the full frame global features. This complies with previous findings about local features effectiveness as mentioned in the literature review. Regarding timing, F-2-F based on DC-image local features costs an extra 0.5 second to match a single pair of video shots as shown in Table.2. This is due to the fact that local features extraction and matching is more costly than global features, since it involves more complex computations[17]. Even with such timing difference, the proposed approach still works in real-time margin (away better than the main F-2-F with global features, on full frames) but with higher accuracy.

Technique	Average matching time per video pair(seconds)
F-2-F (Global features+ Full-image)	1.3
F-2-F (Global features+ DC-image)	0.059
F-2-F (Local features+ DC-image)	0.56

Table 2 Timing analysis for the proposed F-2-F based on local vs. Global features considering the DC-image and Full frame respectively.

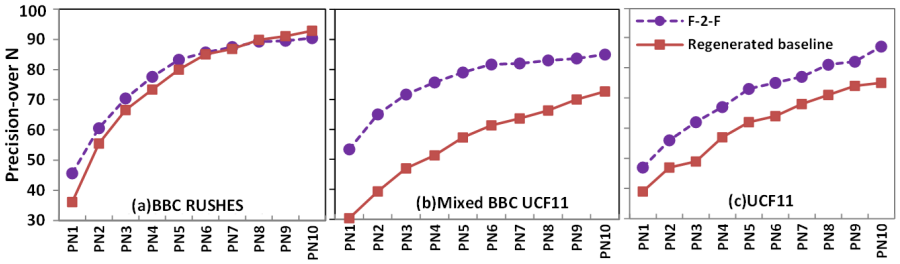

Fig. 6 The proposed F-2-F based on local features versus the regenerated F-2-F baseline, based on global features showing (a)2.3% on average higher P_N and 16.36% more correct matches (from Fig.7.a), over the full BBC Rushes dataset, (b)20% on average higher P_N over the full mixed BBC Rushes and UCF11 dataset, using the mixed dataset gives more consolidation for the output results,(6)10.1% on average higher P_N over UCF11 dataset.

4.3 Revised F-2-F matching vs. baselines

Following the results presented in previous sections, 4.1 and 4.2, we conclude that the proposed F-2-F technique based on DC-image and local features is able to work in real-time manner. In addition, it achieved higher levels of accuracy compared to the same technique based on global features. This section empirically validate the proposed technique versus other baselines. We chose two baselines. The first is the trajectories[7] approach which encodes SIFT local features in the form of trajectories that capture the spatial and temporal features of a video shot. The second is the latest implementation of the F-2-F technique[19] that utilized global feature (color histogram) for underlying frame matching. It worth mentioning that the regenerated baseline applied on the DC-image as a common base with the revised F-2-F, while the original F-2-F applied on full size frames.

Starting with the second baseline (F-2-F with global features), it reported an average precision of 60% based on five sample queries (only) out of 100 videos dataset. However, due to the difficulty of obtaining the dataset in order to obtain robust results based on a greater number of queries, we regenerated this baseline based on the BBC RUSHES, UCF11 and the mixed dataset of both, using the same color histogram global feature. For BBC RUSHES (Fig.6.a) although the regenerated baseline have higher P_N across the $8^{th}-10^{th}$ ranks, the revised F-2-F is better by 2.3% on average overall ranks. Also, it is able to retrieve 16.36% more correct matches, as depicted in Fig.7.a.

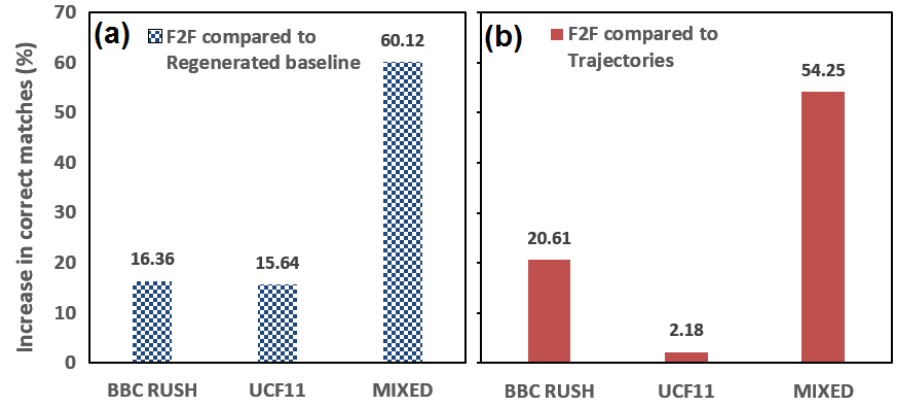

Fig. 7 Percent increase of correct matches(%) due to using F-2-F compared to: (a)the regenerated baseline and (b)trajectories baseline (values computed across top-10 ranks for each technique). Obviously F-2-F is able to retrieve more correct matches than mentioned baselines across all datasets.

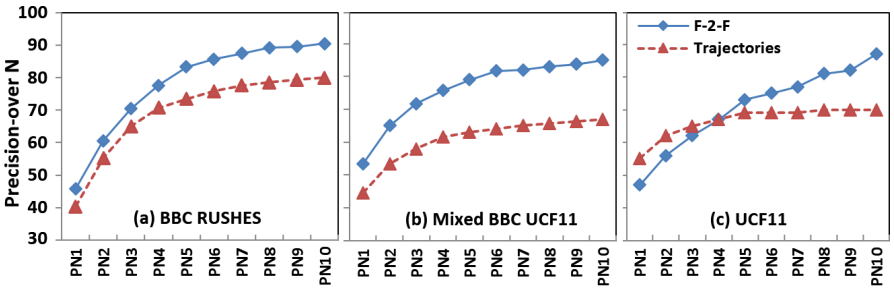

Fig. 8 the proposed F-2-F technique against trajectories baseline using (a)BBC RUSHES, showing 8.39% on average higher P_N, (b)Mixed BBC UCF11 dataset, showing 15.13% on average higher precision,(c)UCF11 dataset, showing 8.3% an average higher P_N across 4^{th} -10^{th} ranks, and an overall of 2.18% increase in total correct matches (from Fig.7.b.)

For the mixed dataset the revised F-2-F achieved 20% on average higher P_N compared to the regenerated baseline,as depicted in Fig.6.b. This corresponds to a 60.12% increase in correct matches as showed in Fig.7.a. Finally, regarding UCF11 dataset the revised F-2-F achieved 10.1% on average higher P_N, which corresponds to 15.64% increase in correct matches, as illustrated in Fig.6.c and Fig.7.a.

Regarding the trajectories baseline, Fig.8 depicts P_N curves for the revised F-2-F against trajectories, for BBC RUSHES, the mixed dataset and UCF11 respectively. A notable finding is that the revised F-2-F outperforms this sophisticated trajectories approach for the first two datasets Fig.8.a and Fig.8.b. For UCF11 (Fig.8.c), the revised F-2-F retrieves in total 2.18% more correct matches (depicted in Fig.7.b). Although the trajectories baseline retrieves less correct matches, it presents its (less correct matches) at earlier ranks (1^{st}-3^{rd}), while the revised F-2-F continues to retrieves more correct matches from 4^{th}

	P-Value
Revised F-2-F Local vs. Regenerated F-2-F Global (BBC Rushes)	0.01
Revised F-2-F Local vs. Regenerated F-2-F Global (mixed BBC Rushes and UCF11)	0.0000043
Revised F-2-F Local vs. trajectories (BBC Rushes)	0.0000012
Revised F-2-F Local vs. trajectories (mixed BBC Rushes and UCF11)	0.000000085

Table 3 Evaluation of F-2-F approach based on global features versus local features, for full image and DC-image respectively using unpaired t-test with confidence value of 95%.

to 10^{th} rank. This odd behaviour, because trajectories are designed to model the motion, and UCF11 is primary an action recognition dataset, where its ground truth rules are built on action similarity, while the revised F-2-F offers a generic matching based on local features. However, the overall performance of the the revised F-2-F is still higher, as it retrieves more correct matches (Fig.7.b). Furthermore, to test the significance of the proposed work against the baselines, a t-test was carried and the results are depicted in Table.3. The results confirm that the proposed F-2-F significantly outperforms the other two baselines (p-values $<<0.05$).

For timing analysis against baselines, Table.4 shows video matching times for the revised F-2-F (local features on DC-image) against the main baseline F-2-F (global features on full image), the regenerated baseline F-2-F(global features on DC-image) and trajectories. It is noticeable that, the main baseline F-2-F(global features on full image) is the most costly, as it involves decompressing of full frames which is a lengthy process and makes the entire approach not suitable for speedy matching. The revised F-2-F (local features on DC-image) comes as the second highest (best accuracy), achieving 56% time reduction over the main F-2-F baseline. Finally, trajectories and the regenerated baseline F-2-F(global features on DC-image) comes at the end, as the fastest techniques (with the lower accuracy), with only 0.2 and 0.5 seconds higher than the revised F-2-F. This extra time for the revised F-2-F (compared to the fastest), is due to the exhaustive comparison of SIFT keypoints among all possible frame pairs to fill the initial frame similarity matrix. However, the technique still works in real-time margin, and several optimizations could be done e.g. optimizing the code, reducing the number of matching frame pairs to fill the initial similarity matrix, replacing the dynamic programming algorithm by a faster one and improving the sigma adjustment process for a faster performance.

Finally, and in order to develop an interactive testing tool for the presented approach and to facilitate investigating the results, we developed a custom software to visualize the output matching results. Fig.9 and Fig.10 depict snapshots of real video matching examples from two different datasets

Technique	Average matching time(seconds) per video pair
Revised F-2-F (DC-image +Local features)	0.56
Regenerated baseline F-2-F (Global features+ DC-image)	0.059
Trajectories	0.36
Main baseline F-2-F (Global features+ Full-image)	**1.3**

Table 4 Timing analysis for the proposed F-2-F (global and local features) vs. trajectories.

Fig. 9 Snapshot of video finder, using F2F based on local features (BBC_RUSHES).

Fig. 10 Snapshot of video finder, using F2F based on local features (mixed BBC and UCF11).

visualized using the developed software. A web-based version (slower) is also available on `http://dcapi.blogs.lincoln.ac.uk/F2F-demo/`.

5 Conclusion

In this paper we proposed a revised F-2-F technique for matching compressed video shots, through local features extracted directly from DC-images. This is in contrast with previous versions of F-2-F that used global features extracted from full size frames, which was not practically applicable for real-time use, especially on compressed videos. Hence, we presented an improved accuracy

(compared to recent baselines) as well as faster technique (compared to the original F-2-F (global features from full frames)) for speedy videos matching. Moreover, problems of extracting SIFT features, from such small size DC-images, was successfully resolved by adaptively controlling the amount of Gaussian blurring applied at each octave layer of SIFT detection. This adaptation allowed extraction of sufficient keypoints for robust matching. Followingly, a similarity matrix is generated between matching videos frame pairs, and finally the matching is being done by a modified dynamic programming algorithm. This algorithm locates the best set of matching frame pairs, taking into account their respective temporal order. The revised technique was validated against some challenging datasets, that showed its robustness and ability to work in real-time environment with higher accuracy, compared to other matching techniques. Moreover, the technique could act efficiently to retrieve an initial set of maximum matching videos, to be followed by additional layers for further re-ranking of the videos and/or for further semantic analysis and annotation such as in[7,6]. Future work improvements may include, examining other sets of local features that incorporates color (e.g. CSIFT[3]), to further improve the results of the proposed work. In addition a re-ranking algorithm could be applied on the top-N for more precise results. Regarding time improvement the internal dynamic programming algorithm could be more refined for a faster operation and the sigma adjustment process could be tuned for a faster keypoints extraction.

References

1. Youtube statistics (2013). URL `http://www.youtube.com/yt/press/statistics.html`
2. Abbass, A., Youssif, A., Ghalwash, A.: Compressed domain video fingerprinting technique using the singular value decomposition. In: Proceedings of Applied Informatics and Computing Theory (2012)
3. Abdel-Hakim, A.E., Farag, A.A.: Csift: A sift descriptor with color invariant characteristics. In: Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on, vol. 2, pp. 1978–1983 (2006)
4. Adjeroh, D.A., Lee, M.C., King, I.: A distance measure for video sequence similarity matching. In: Multi-Media Database Management Systems, 1998. Proceedings. International Workshop on, pp. 72–79 (1998)
5. Almeida, J., Leite, N.J., da S Torres, R.: Comparison of video sequences with histograms of motion patterns. In: IEEE International Conference on Image Processing, pp. 3673–3676 (2011)
6. Altadmri, A., Ahmed, A.: Video databases annotation enhancing using commonsense knowledgebases for indexing and retrieval (2009)
7. Altadmri, A., Ahmed, A.: A framework for automatic semantic video annotation. Multimedia Tools and Applications **64**(3), 1–25 (2013)
8. Bannour, H., Hlaoua, L., el Ayeb, B.: Survey of the adequate descriptor for content-based image retrieval on the web: Global versus local features. In: Conference en recherche d'Information et Applications (CORIA'09), pp. 445–456 (2009)
9. Bekhet, S., Ahmed, A., Hunter, A., et al.: Video matching using dc-image and local features. Lecture Notes in Engineering and Computer Science **3**, 2209–2214 (2013)
10. Dimitrova, N., Abdel-Mottaleb, M.S.: Video retrieval of mpeg compressed sequences using dc and motion signatures. Video retrieval of MPEG compressed sequences using DC and motion signatures (1999)

11. Droueche, Z., Lamard, M., Cazuguel, G., Quellec, G., Roux, C., Cochener, B.: Content-based medical video retrieval based on region motion trajectories. In: Proceedings of International Federation for Medical and Biological Engineering, pp. 622–625. Springer (2012)
12. Gao, H.P., qiao Yang, Z.: Content based video retrieval using spatiotemporal salient objects. In: International Symposium on Intelligence Information Processing and Trusted Computing (IPTC), pp. 689–692 (2010)
13. Hua, X.S., Chen, X., Zhang, H.J.: Robust video signature based on ordinal measure. In: International Conference on Image Processing (ICIP '04), vol. 1, pp. 685–688 Vol. 1 (2004)
14. Karpenko, A., Aarabi, P.: Tiny videos: A large data set for nonparametric video retrieval and frame classification. Pattern Analysis and Machine Intelligence, IEEE Transactions on **33**(3), 618–630 (2011)
15. Kogler, M., del Fabro, M., Lux, M., Schoffmann, K., Boszormenyi, L.: Global vs. local feature in video summarization: Experimental results. In: 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies. SeMuDaTe (2009)
16. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos in the wild. In: IEEE Conference on Computer Vision and Pattern Recognition(CVPR '9), pp. 1996–2003. IEEE (2009)
17. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International journal of computer vision **60**(2), 91–110 (2004)
18. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. In: Computer Vision and Pattern Recognition, 2003. Proceedings. 2003 IEEE Computer Society Conference on, vol. 2, pp. II–257–II–263 vol.2 (2003)
19. Ng, C.W., King, I., Lyu, M.R.: Video comparison using tree matching algorithm. In: Proceedings of The International Conference on Imaging Science, Systems, and Technology, vol. 1, pp. 184–190 (2001)
20. Over, P., Awad, G.M., Fiscus, J., Antonishek, B., Michel, M., Smeaton, A.F., Kraaij, W., Qunot, G.: Trecvid 2010 an overview of the goals, tasks, data, evaluation mechanisms, and metrics (2011)
21. Shan, M.K., Lee, S.Y.: Content-based video retrieval based on similarity of frame sequence. In: Multi-Media Database Management Systems, 1998. Proceedings. International Workshop on, pp. 90–97. IEEE (1998)
22. TrecVid(2011): Trec video retrival task, bbc ruch 2005 (1-02-2011). URL `www.nplpir.nist.gov/projects/trecvid`
23. Yeo, B.L., Liu, B.: On the extraction of dc sequence from mpeg compressed video. In: Image Processing, 1995. Proceedings., International Conference on, vol. 2, pp. 260–263. IEEE (1995)

Saddam Bekhet is a PhD researcher at School of Computer Science in University of Lincoln, UK and an assistant lecturer in SouthValley University-Egypt. Received a MSc and BSc degrees in Computer Science at University of Assuit, Egypt in 2010 and 2005 respectively. Saddam's current research focus on content based video retrieval, especially from the visual content perspective, video and scene analysis understanding.

Amr Ahmed (BSc93, MSc98, PhD04, MBCS05, IEEE-CS08) is a Senior Lecturer, and the Founder and the Leader of the DCAPI (Digital Contents Analysis, Production, and Interaction: http://dcapi.lincoln.ac.uk) research group at the School of Computer Science, University of Lincoln, UK. His research focuses on the analysis, understanding, and interpretation of digital contents, especially visual contents. Amr's current research interests include Contents-Based Image/Video retrieval, video and scene understanding, semantic analysis, integration of knowledge and various modalities for scene understanding. Amr worked in the industry for several years, including Sharp Labs of Europe (SLE), Oxford (UK), as a Research Scientist, and other Engineering Consultants companies abroad. He also worked as a Research Fellow, at the University of Surrey, before joining the academic staff at the University of Lincoln in 2005. Dr. Ahmed is a Member of the British Computer Society (MBCS) and the IEEE Computer Society. He received his Bachelors degree in Electrical Engineering and M.Sc. degree (by research) in Computer and Systems Engineering, from Ain Shams University, Egypt, in 1993 and 1998 respectively, and his Ph.D. degree in Computer Graphics and Animation from the University of Surrey, U.K., in 2004.

Amjad Altadmri received the PhD degree from the School of Computer Science at the University of Lincoln, UK in 2013. A Bachelor of Engineering in Computer Science from University of Damascus, Syria in 2004. Amjads current research focuses on Semantics of Visual contents, both from visual contents area and the link to the semantic textual one. His other research interests include Video Understanding, Ontology and Commonsense.

Andrew Hunter studied for his BSc Mathematics and Computing and PhD Computer Graphics at Bath University. He worked for several years in industry, in computer graphics and CAD/CAM software, before returning to - academia. Prof. Hunter held posts at Sunderland and - Durham Universities, before joining the University of Lincoln and becoming Head of Department of Computing in 2004, Dean of Research in 2007, and Dean for Science, Technology and Engineering in 2010. Professor Hunter has published over 80 academic papers, including more than 20 in international journals. He has also developed several freeware and commercial artificial intelligence software packages. Professor Hunter's research interests are in computer vision and artificial intelligence, and particularly in medical applications of vision and automated surveillance. Major recent projects include: BRAINS, a TSB-sponsored project to develop embedded neurally-inspired smart sensor systems for surveillance based on FPGA technology; TOTALCARE, a TSB-sponsored project to develop an integrated automated

in-home monitoring system for assistive care, including novel behaviour analysis; and retinal analysis, including detection and measurement of features of the retinal vasculature, and the application of level set methods and novel shape features in lesion detection and classification. He leads a small group of three Research Assistants and PhD students.