Signal, Image and Video Processing manuscript No.
(will be inserted by the editor)

Content Based Video Copy Detection Based on Motion Vectors Estimated Using a Lower Frame Rate

Kasım Taşdemir · A. Enis Çetin

Received: date / Accepted: date

Abstract We propose a motion vector based Video Content Based Copy Detection (VCBCD) method. One of the signatures of a given video is motion vectors extracted from image sequences. However, when consecutive image frames are used they are not descriptive enough because most vectors are either too small or they appear to scatter in all directions. We calculate motion vectors in a lower frame rate than the actual frame rate of the video to overcome this problem. As a result we obtain large vectors and they represent a given video in a robust manner. We carry out experiments for various parameters and present the results.

Keywords Content Based Copy Detection · Similar Video Detection · Motion Vectors · Sequence Matching · Video Copy Detection

1 Introduction

Detecting videos violating the copyright of the owner comes into question by growing broadcasting of digital video on different media. Content based copy detection (CBCD) is an alternative way to watermarking approach to identify the ownership of video. In this approach, the video itself is considered as a watermark. Existing methods of video CBCD usually extract signatures, key-frames or fingerprints from images of video stream and compare them with the database which contains features of original videos. [3, 13, 14, 16, 15, 6] Several spatial or temporal features of videos are considered as signatures of videos such as intensity of pixels, color histograms and motion.[9, 4] The main advantage of CBCD over watermarking is that signature extraction can be done even if the video is distributed over the Internet or other media because the unique signature is part of the video itself.

K. Taşdemir
Department of Electrical and Electronics Engineering, Bilkent University
Ankara, TR-06800, Turkey
E-mail: tasdemir@ee.bilkent.edu.tr

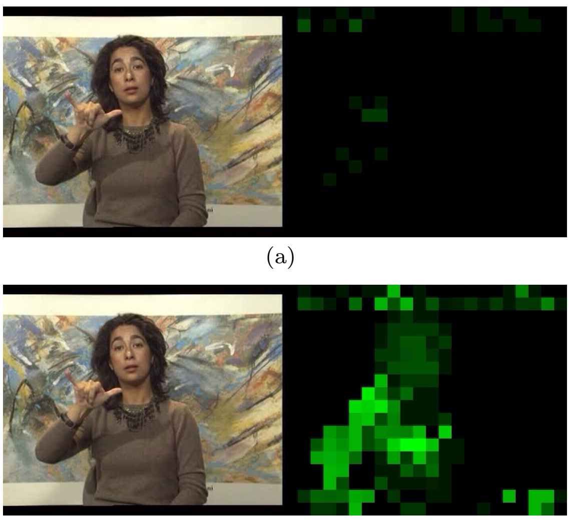
A. Enis Çetin
Department of Electrical and Electronics Engineering, Bilkent University
Ankara, TR-06800, Turkey
E-mail: cetin@bilkent.edu.tr

In CBCD algorithms, average color, color histogram, and motion are used as feature parameters or vectors. Each feature set has advantages over others. When a movie is recorded from a movie theater by a hand-held camera, then its color map, fps, size and position change and edges get soften. Color based algorithms will have difficulties detecting the camera recorded copy of an original movie because the information it depends on is significantly disturbed. However, motion in a copied video remains similar to the original video.

Motion information was considered as a weak parameter by other researchers [4]. In the article [12] it is shown that this is not true unless the motion vectors are extracted from consecutive frames in a video with a high capture rate. Most motion vectors are small or close to zero in a typical 25 Hz captured video and they may not contain any significant information. They also appear to scatter in all directions due to incorrect motion vector calculation because neighboring pixel values are close to each other in consecutive video frames. On the other hand, if we can detect the general motion trends in a video as representive of the video we get a reliable feature set of parameters. In this article, we calculate motion vectors in a lower frame rate than the actual frame rate of the video. As a result we obtain larger vectors compared to the motion vectors obtained at a higher rate and we experimentally show that they represent a given video in a robust manner.

1.1 Motion Vector (MV) Extraction and Feature Sets

In general, motion vectors are extracted using consecutive frames in many video analysis and coding methods. In order to capture the temporal behavior more efficiently we use every t^{th} and $(t+n)^{th}$, $n > 1$ frames instead of the traditional approach of using t^{th} and $(t+1)^{th}$ frames. In our approach, we use every t^{th} and $(t+n)^{th}$ frame for motion vector extraction. For example, human movements change slowly in a 25 fps video. If two consecutive frames are used in motion vector extraction step, resulting motion vectors will have small values because of the high capture rate of the video. Furthermore, some of the image-blocks (or macro blocks) inside the moving object may be incorrectly assumed as stationary or moving in an incorrect direction by the motion estimation algorithm because similar image blocks may exist inside the moving object. By computing the MVs using every n-th frame ($n > 1$) it is possible to get more descriptive motion vectors. In Fig. 1, instead of using two consecutive frames we use t^{th} and $(t+5)^{th}$ frames for MV computation and, as a result, MV displacements in the video will be high. As shown in Fig. 1, moving objects are clearly emphasized.



(a)

(b)

Fig. 1 Effect of lower fps in the motion vector estimation algorithm: (a) 151^{th} frame and its corresponding MV pattern of video "silent". MVs are extracted using the next frame. The MV magnitudes are small. (b)151^{th} frame of video "silent". MVs are extracted using every 5^{th} frame. The MV magnitudes are larger than (a).

We define the mean of the magnitudes of motion vectors (MMMV) of macro blocks of a given frame as follows:

$$MMMV(k) = \frac{1}{N}\sum_{i=0}^{N-1} r(k,i) \tag{1}$$

where $r(k,i)$ is the motion vector magnitude of the macro block in position i of k^{th} frame, and N is the number of macro blocks in an image frame of the video. We also define the mean of the phase angles of motion vectors (MPMV) of macro blocks of a given frame as follows:

$$MPMV(k) = \frac{1}{N}\sum_{i=0}^{N-1} \theta(k,i), \tag{2}$$

where $\theta(k,i)$ is the motion vector angle of the macro block in position i of the k^{th} frame of the video, and N is the number of macro blocks. The angle θ is in radians and $\theta \in (-\pi,\pi)$. So, the range of $MPMV$ is also in the same region: $MPMV(.) \in (-\pi,\pi)$.

We use the discrete MMMV(k) and MPMV(k) functions as the feature sets representing a given video. Example MMMV and MPMV plots are shown in Fig. 2 and Fig. 7, respectively. Storage requirement is low as both functions require a single real number for each frame k of the video. It is possible to divide the image frames into subimages and extract MMMV and MPMV values for each subimage but we experimentally observe that a single value for a given frame is sufficient to characterize a video.

In the following subsection we describe the method that we used for motion vector (MV) estimation in video.

1.2 Motion Vector Extraction

We extract motion vectors from image frames using the simple and efficient search (SES) algorithm [10] and use an exhaustive search (ES) [2] for block matching. However, other motion vector estimation methods can be also used.

Block matching is performed on the current frame (t) and a previous frame (t-u). The current frame is divided into square blocks of pixel size $N \times N$ called macro blocks (MB). Each block has a search area in the previous frame which has the size $(2W+N+1)\times(2W+N+1)$ where W is the amount of maximum vertical or horizontal displacement. Then, the best matching block is searched in the previous frame using the current block. The motion vector is defined as the (x,y) which makes the mean absolute difference (MAD) minimum. The MAD is expressed as

$$MAD(x,y) = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} |F_c(k+i,l+i) - F_p(k+x+i,l+y+j)| \quad (3)$$

where $F_c(.,.)$ and $F_p(.,.)$ are pixel intensities of the current and the previous frames respectively, (k,l) is the horizontal and vertical coordinates of the upper left corner of the image block and (x,y) is displacement in pixels.[10]

1.3 Exhaustive Search Algorithm

Another name of this algorithm is the Full Search algorithm. This calculates MAD for all possible locations in a given search window. As a result it gives the best possible match and the highest PSNR amongst any block matching algorithms.[2] This algorithm is straightforward to implement and gives the best results. The disadvantage of this algorithm is its high computational cost.

1.4 A Simple and Efficient Search Algorithm

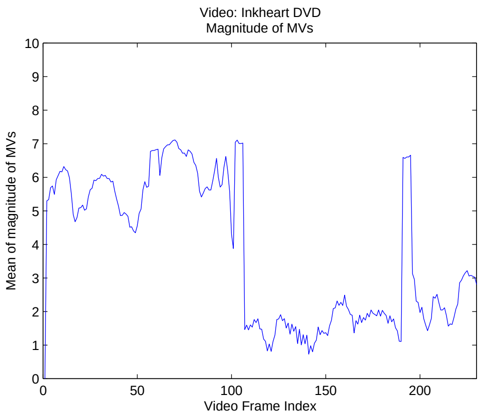
This algorithm is a modified version of the three step search (TSS) algorithm. [10,2] In the TSS algorithm a block is searched in some reference points of locations in the previous frame instead of searching all possible locations. First, points in the center and 8 points around the center are checked. If the minimum is at the lower right point, the search algorithm continues in the same manner with a smaller search window. After applying it three times, the location that gives the minimum MAD is found. The motion vector is decided as a vector from the center to that point.

2 Video Copy Detection Using MMMV and MPMV

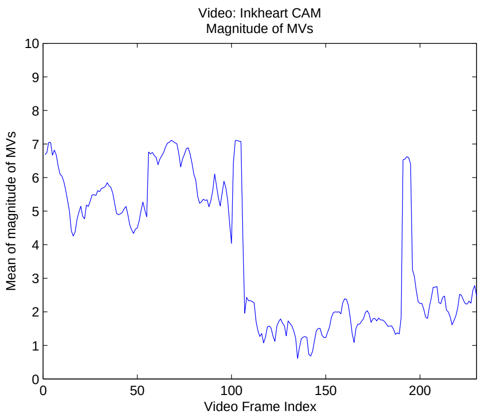
Searching and comparing the movies violating the copyright issues with official movies may not be a challenging problem if we know that the copied movie has exactly the same digital data as the original. However, in most cases unofficial movies are published with a small distortion or additions such as resizing, cropping, zooming in and out, adding a logo, changing the fps, changing color etc. Most encountered real life example is distribution of hand-held camera recorded movies of new movies from the movie theater. Since this unofficially made copy is a completely new record, it loses some of the features of the original movie. For instance, colors will change both due to the projector illuminating the curtain and during the camera recording. Depending on the quality of the recording device, its view point and its orientation, recorded movie may lose edges in frames or it may have different scale and perspective than the original movie. Color histogram based CBCD comparison methods have the disadvantage that they depend on the distorted color information. However, the motion vectors do not change as much as color information. This section investigates the similarity of MMMV-MPMV data of original movies and their hand-held camera versions. Table 1 shows the properties of the movies used in this section. Test videos have different size and fps. Videos with CAM extensions are copies obtained using a hand-held camera. In Sect. 3 we present extensive comparisons using a video database. Although the original and hand-held camera recorded videos have different fps and size, they have similar MMMV plots as shown in Fig. 2. Original movie in Fig. 2(a) and its hand-held camera recorded version from a movie theater (Fig. 2(b)) show significant similarities. The MVs are computed with a frame difference of n=5. In order to obtain a value that gives information about how much two movies resemble each other, the absolute difference is calculated as distance, D. Differencing the two features directly is not a good solution because of two reasons. The first reason is that they may have different fps values. So, each index of the original video should be compared with its corresponding index of the candidate video in terms of real time. However, most of the indices do not correspond to the same time instant. After calculating the indices corresponding to the nearest time instant, we use a search window in order to compare it with also its neighbors.

Table 1 Properties of original movies (with DVD extension) and the same movies recorded from a hand-held camera (with CAM extensions).

Movie Name	(Frames Per Second) FPS	Video Size
Desperaux DVD	24	640x272
Desperaux CAM	25	608x304
Inkheart DVD	25	624x352
Inkheart CAM	25	704x304
Mallcop DVD	30	608x320
Mallcop CAM	24	720x320
Spirit DVD	24	640x272
Spirit CAM	25	656x272



(a)



(b)

Fig. 2 Similarity of the MMMV plots of "Inkheart DVD" and "Inkheart CAM", (with n=5).

The second reason is that the sizes of frames of the videos can be different. If frame sizes are different, motion vectors of videos will be also different. The video with a larger frame size will have larger motion vectors. The MMMV data of videos will be scaled version of each other. In order to solve this problem we first normalize the MMMV and MPMV of the videos before making a comparison as follows:

$$\overline{MMMV}(t) = V(t) = \frac{MMMV(t) - \mu_{MMMV}}{\sigma_{MMMV}} \qquad (4)$$

where μ_{MMMV} is the mean and σ_{MMMV} is the standard deviation of the MMMV array, respectively.

The sum of absolute values of difference of normalized MMMV values of two videos o-original and c-copy are calculated as the distance $D(o, c)$ as follows:

$$D(o, c) = \frac{1}{N} \sum_t \min_{|d| \leq W} |V_o(t) - V_c(t + d)| \qquad (5)$$

where W is the search window width. We time align the videos manually and select W as 2 frames because the fps of most commercial videos are between 20 and 30. In Eq. 5, N is the number of frames in the movie $MMMV_o$. If the original and the candidate video have different fps, then their frame indices corresponding to the same time instance should be calculated first. So, instead of comparing corresponding frame indexes, the aim is to compare image frames corresponding to the same time instant.

We define another measure of the distance between two video clips based on estimating the $V_c(t)$ sequence of the video clip c using the $V_o(t)$ of the video clip o as follows:

$$D(o, c) = \frac{1}{N} \sum_t |V_o(t) - \sum_{k=-L}^{L} w_{k,t} V_c(t - k)| \qquad (6)$$

where $w_{k,t}$ are the weights of the $2L + 1$ order linear estimator. The weights are adaptively updated using the well-known LMS algorithm:

$$w_{k+1,t} = w_{k,t} + \lambda e(t) V_c(t) \qquad (7)$$

where λ is the adaptation constant and

$$e(t) = V_o(t) - \sum_{k=-L}^{L} w_{k,t} V_c(t - k) \qquad (8)$$

is the estimation error at frame t. The parameter λ can be selected as in the normalized LMS algorithm.

The distance D of a video of an original movie Inkheart and the same video recorded with a hand-held camera is shown in Fig. 3. The last plot shows the absolute of frame by frame $\overline{MMMV}$ difference. Since the $\overline{MMMV}$ plot of the two videos are similar, their average of absolute difference value is small, 0.35. However, the distance of two different videos are not small as shown in Fig. 4. Since the two different movies have different camera motions and object movements, their $\overline{MMMV}$ plots are not similar, $D(o, c) = 2.91$. However, distance of original video o and hand-held camera recorded video c is 0.35, $D(o, c) = 0.35$.

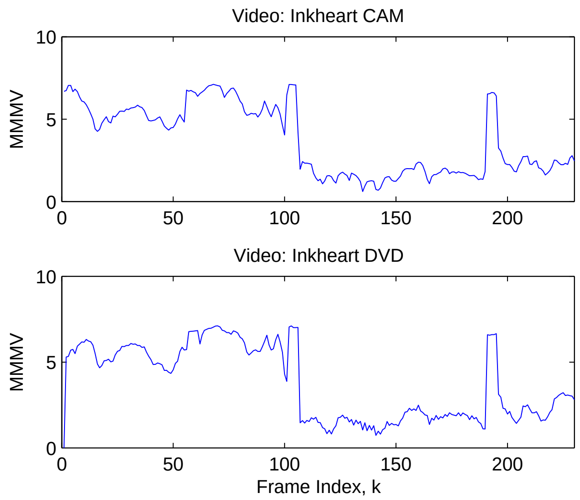


Fig. 3 MMMV plots of videos “Inkheart DVD” and “Inkheart CAM” videos. The distance between the MMMV plots, $D(o, c) = 0.35$.

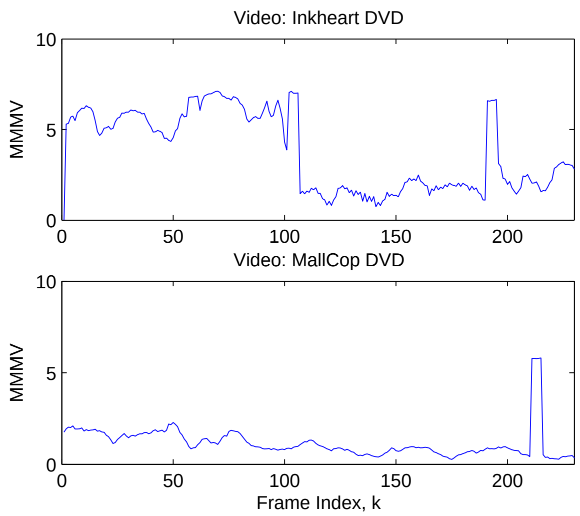


Fig. 4 MMMV plots of “Inkheart DVD” and “Mallcop.CAM” videos. The distance between the MMMV plots,$D(o, c) = 2.91$.

Comparison of distances of 8 test videos are listed in Table 2. Rows of Table 2 are original videos and columns are hand-held camera recorded versions. The diagonal elements of Table 2 is a measure of similarity of the original and copy of the video. Diagonal elements are expected to be smallest value in a given row because a video should be similar to its copy and different from the others. Diagonal elements are the smallest values which mean that the original videos are most similar to their camcorder copy in terms of $MMMV_N$. Although the camera recordings of video “Desperaux CAM” is at a very low quality and it has significant morphological distortions it successfully paired with its original version. Sample screen shots of same frames of videos of “Desperaux CAM” and “Desperaux DVD” are shown in Fig. 6. Side portions of the video is lost because of zoom in of the hand-held camera and camera focus is not adjusted so it is very blurred. $\overline{MMMV}$ plot and the distance plot of “Desperaux DVD” and “Desperaux CAM” are shown in Fig. 5.

Table 2 Average $MMMV_N$ distance of test videos. Diagonal results show the distance of original and its copy. V1 to V4 stands for the names of the test videos ”Desperaux”, ”Inkheart”, ”Mallcop”, ”Spirit”.

Distance	V1 CAM	V2 CAM	V3 CAM	V4 CAM
V1 DVD	**0.44**	1.23	0.9	0.86
V2 DVD	1.2	**0.08**	0.68	0.74
V3 DVD	0.85	0.54	**0.18**	0.75
V4 DVD	1.06	0.76	0.67	**0.29**

(a)

(b)

Fig. 5 The same frames of videos “Desparaux.DVD” and “Desparaux.CAM”, (a) the original movie frame and (b) the same frame for the video recorded by a hand-held camera. It is highly distorted.

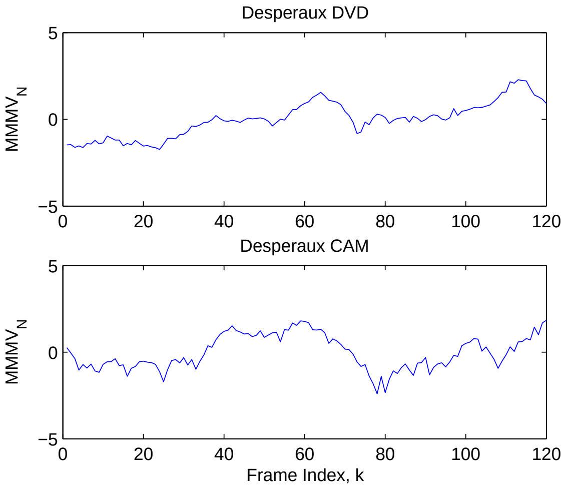


Fig. 6 $\overline{MMMV}$ plots of "Desparaux.DVD" and "Desparaux.CAM" video clips. The distance between the MMMV plots, $D(o,c) = 0.44$.

As discussed above angle information of motion vectors can be also used for video copy detection. The MPMV plots of "Inkheart DVD" and "Inkheart CAM" are shown in Fig. 7. The original video and the recorded video have very similar MPMV plots. Comparison results of test videos are listed in Table 3. Diagonal elements of the Table 3 are the smallest elements in a given row in Table 3. The distance between the original video and the corresponding copy pair is the smallest. So, $\overline{MPMV}$ data of similar videos are found to be the most similar data amongst test videos.

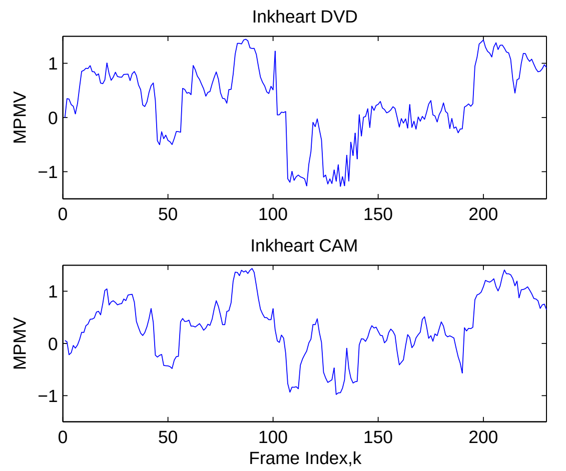


Fig. 7 The MPMV plots of "Inkheart DVD" and "Inkheart CAM" video clips. The distance between the MPMV plots, $D(o,c) = 0.22$.

Table 3 Average distance D of $\overline{MPMV}$ data of test vidoes. Diagonal results show the distance between the original and its copy. V1 to V4 stands for the names of the test videos "Desperaux", "Inkheart", "Mallcop", "Spirit".

Distance	V1 CAM	V2 CAM	V3 CAM	V4 CAM
V1 DVD	**0.29**	0.96	0.7	0.74
V2 DVD	1.03	**0.15**	0.85	0.86
V3 DVD	0.98	0.87	**0.4**	0.74
V4 DVD	0.62	0.75	0.59	**0.24**

As discussed above MMMV or MPMV information can be used as a feature of the video. Comparison results show that they can be used for detection of artificially or manually modified versions of original videos. Each has superior sides. As it is shown in Table. 3, phase information is more resistant to loss of some information and significant deformations in the video. Even magnitude data of the videos were not enough to detect the "Desperaux DVD" and "Desperaux CAM" as similar videos, phase data gave correct matching. On the other hand, MPMV is not rotation invariant but MMMV is rotation invariant. Therefore, both features should be used at the same time.

3 Experimental Results

A video database is available in Ref. [1]. Original videos in this database are compared with the transformed versions of the same videos. There are 47 original videos taken from Ref. [1]. Duration of the videos are 30 seconds. Each video has eight different transforms. The list of transformations is given in Table 4. As a result there are a total of $47 \times 9 = 423$ videos in the database. For each parameter set 1457 comparisons are performed.

Table 4 Video transformations

Transformation Name	Transformation Type
T1	A pattern inserted
T2	Crop 10% with black window
T3	Contrast increased by 25%
T4	Contrast decreased by 25%
T5	Zoom 1.2
T6	Zoom 0.8 with black window
T7	Letter-box
T8	Gaussian noise, $\mu = 0, \sigma = 0.001$

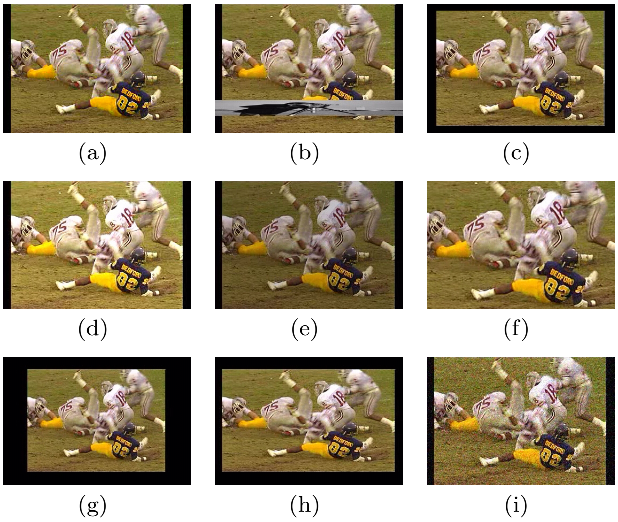


Fig. 8 Transformations: (a) original frame, (b) a pattern is inserted, (c) crop 10% with a black frame, (d) contrast increased by 25%, (e) contrast decreased by 25%, (f) zoom by 1.2, (g) zoom by 0.8 with in the black window, (h) letter-box, (i) additive Gaussian noise with $\mu = 0$ and $\sigma = 0.001$.

Original videos are compared with test videos in the database and its 8 transformations. For each test, the list of distance between the compared videos are calculated using Eq. 5 for different parameters or data types such as MMMV, MPMV etc..

The performance of each test is plotted using its receiver operating characteristics (ROC) curve. The ROC curve is a plot of false positive rate F_{pr} and false negative rate F_{nr}, or true positive rate T_{pr}. Let F_p, F_n and T_p the number of false positives (clips that matched with a different video), false negative (clips that should match, but did not) and true positive (hit; clips that matched correctly in the positive set). False positive , negative rates and true positive rate are defined as

$$F_{pr}(\tau) = \frac{F_p}{N_p}, F_{nr}(\tau) = \frac{F_n}{N_n}, T_{pr} = (\tau) = \frac{T_p}{N_p} \qquad (9)$$

where N_p and N_n are the number of maximum possible false positive and false negative detections. Threshold is τ and its value is varied from 0 to its maximum value with an increment of 1%.

Effect of increasing the frame skipping parameter n from 1 to 7 in motion vector extraction step is shown in Fig. 3. We can obtain more descriptive features of videos based on motion vectors if we use every 7^{th} frame instead of the current and the next frame in motion estimation step. As it is shown in Fig. 10 there is a dramatic increase in detection ratio with increasing n and we get the best result when $n = 7$ where the Area Under Curve (AUC) is 0.9996.

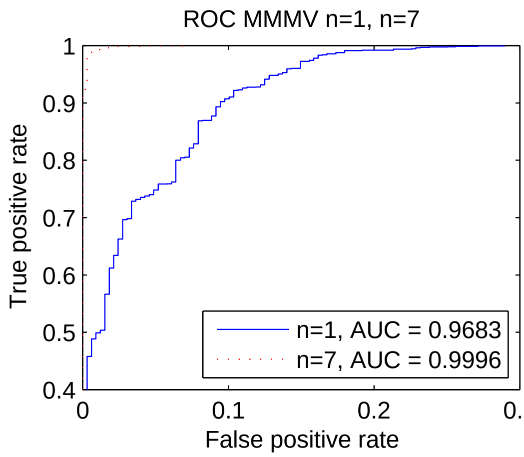


Fig. 9 Effect of increasing the frame distance $n = 1$ to $n = 7$ on $MMMV$ ROC curves.

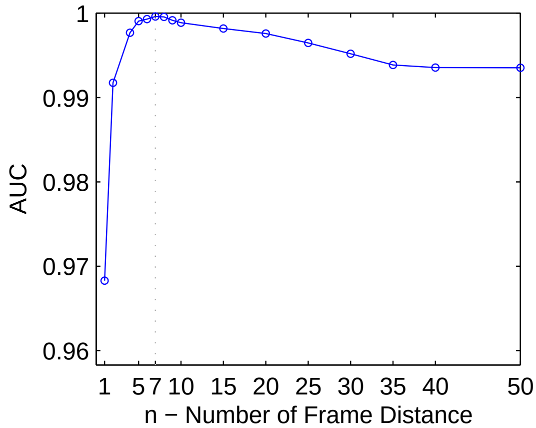
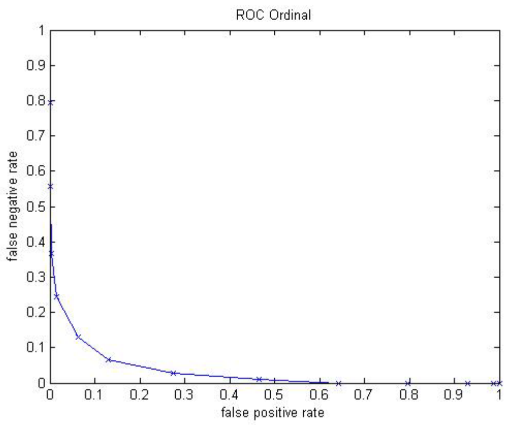


Fig. 10 Effect of varying n on AUC of $MMMV$ ROC curves. The best result is obtained when $n = 7$.

In Ref. [4] it is stated that *ordinal signature* introduced in Ref. [11] outperforms the *Motion Signature*. This is true when the motion vectors are extracted using the current and the next frame ($n = 1$). On the other hand, if motion vectors of videos are extracted using every 5^{th} frame, motion vector based MMMV plot is closer to the ideal case than the ROC curve of ordinal signature as shown in Fig 11. The area under the ROC curve (F_{nr} vs F_{pr}) of *ordinal signature* is 0.0311 which is higher than the area under the ROC curves of both MMMV and the MPMV signatures.



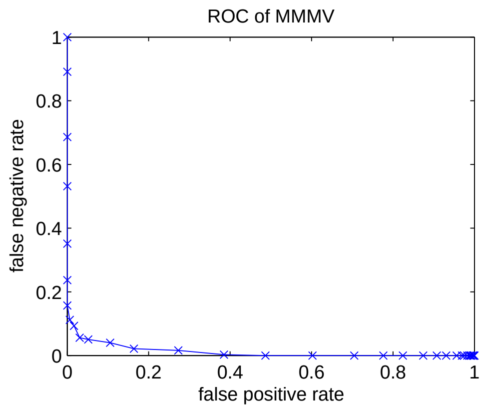
(a)

Fig. 11 The ROC curve (F_{nr} vs F_{pr}) of the *ordinal* signature. In this video database, the MMMV and the MPMV signatures have better performance than the *ordinal* signature when the frame difference parameter $n = 5$.

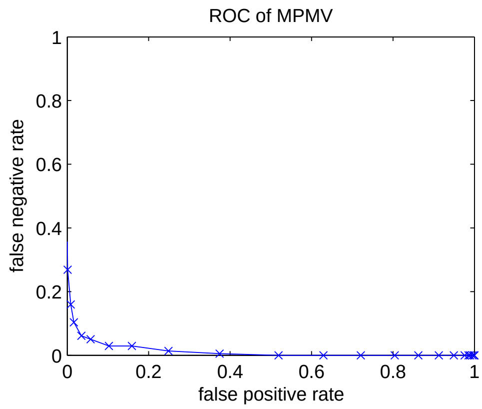
As pointed above the ROC curves of the proposed MMMV and MPMV methods are very close to each other as shown in Fig. 12. It is experimentally shown that the MMMV and the MPMV are good descriptive features for videos. In this database, the best results are obtained with the frame difference parameter $n = 7$ as it is shown in Fig. 10.

Number of Feature Parameters Per Frame

Extracted features are stored in a database. The size of the database is important for practical reasons. Therefore, the number of features extracted for each frame is another important criteria for CBCD algorithms. Table 5 summarizes the feature per frame (FPF) values of several algorithms. The FPF values of algorithms except MMMV and MPMV are taken from Ref. [9].



(a)



(b)

Fig. 12 The ROC curves of proposed methods when the frame difference parameter $n = 5$. (a) MMMV and (b) MPMV.

Table 5 Sizes of feature spaces

Technique	Features Per Frame
ViCopT[8]	7
AJ[5]	4.8
STIP[7]	73
Temporal[9]	0.09
Ordinal Meas.[4]	9
MMMV	1
MPMV	1

Table 5 shows that MMMV and MPMV algorithms consume less space for signatures than the other algorithms except the method called “Temporal”.[9]

4 Conclusions

In this article, we experimentally show that motion vectors are substantial signatures of videos as long as they are extracted in lower frame rates. Therefore, motion vectors can be used in CBCD algorithms.

Videos that have higher motion content give more reliable results and the videos having intensive motion activity are easier to distinguish when the neighboring image frames are used. However, videos containing slow moving objects have very little motion vectors and the vectors may appear to be random when the current and the next frame are used for motion vector computation.

In order to obtain reliable signature vectors for all videos motion vectors of the current and the next n^{th} frame ($n > 1$) are used in motion vector estimation algorithms. Resulting motion vectors provide a reliable representation for all types of videos. Magnitude and phase angles of motion vectors are used separately as feature parameters of a given video. It is experimentally shown that both the magnitude and the phase of vectors can be considered as unique signatures of the video. The proposed motion-based feature parameters are resistant to illumination and color changes in video.

Motion vectors do not change significantly up to a level of resizing, cropping and blurring of the video. Most video copy detection methods are not robust to cropping. The MPMV feature is a robust feature in action videos because the moving objects are cropped in video as they are the information bearing part of a typical video and the direction of the object is the same in both the original and the cropped copy. If the recorded video is in low quality, then phase information is less affected than the magnitude information of the frames. However, MPMV is not rotation invariant but MMMV is rotation invariant. Therefore, it is better to use both MMMV and MPMV at the same time.

Another important comparison criteria of the CBCD algorithms in terms of the practical results is the size of the feature set in a database. The MMMV and the MPMV information do not occupy much space in the database as other methods. They both occupy one byte (one feature) per frame in the database.

References

1. Alexis Joly: Internet archive movie database (2009). URL `http://www.archive.org`. [Online; accessed 10-August-2009; `http://www.worldscinet.com/ijprai/ijprai.shtml`]
2. Chan, E., Panchanathan, S.: Review of block matching based motion estimation algorithms for video compression. In: Electrical and Computer Engineering, 1993. Canadian Conference on, pp. 151–154 vol.1 (1993). DOI 10.1109/CCECE.1993.332213
3. Hampapur, A., Bolle, R.: Comparison of distance measures for video copy detection. In: Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on, pp. 737–740 (2001)
4. Hampapur, A., Hyun, K., Bolle, R.M.: Comparison of sequence matching techniques for video copy detection. In: M.M. Yeung, C.S. Li, R.W. Lienhart (eds.) Storage and Retrieval for Media Databases 2002, vol. 4676, pp. 194–201. SPIE (2001). DOI 10.1117/12.451091. URL `http://link.aip.org/link/?PSI/4676/194/1`
5. Joly, A., Buisson, O., Frelicot, C.: Content-based copy retrieval using distortion-based probabilistic similarity search. Multimedia, IEEE Transactions on **9**(2), 293–306 (2007). DOI 10.1109/TMM.2006.886278
6. Kobla, V., Doermann, D., Lin, K.I.D., Faloutsos, C.: Compressed domain video indexing techniques using dct and motion vector information in mpeg video. In: in Proc. of the SPIE Conference on Storage and Retrieval for Still Image and Video Databases V, pp. 200–211 (1997)
7. Laptev, I., Lindeberg, T.: Space-time interest points. In: IN ICCV, pp. 432–439 (2003)
8. Law-To, J., Buisson, O., Gouet-Brunet, V., Boujemaa, N.: Robust voting algorithm based on labels of behavior for video copy detection. In: MULTIMEDIA '06: Proceedings of the 14th annual ACM international conference on Multimedia, pp. 835–844. ACM, New York, NY, USA (2006). DOI http://doi.acm.org/10.1145/1180639.1180826
9. Law-To, J., Chen, L., Joly, A., Laptev, I., Buisson, O., Gouet-Brunet, V., Boujemaa, N., Stentiford, F.: Video copy detection: a comparative study. In: CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval, pp. 371–378. ACM, New York, NY, USA (2007). DOI http://doi.acm.org/10.1145/1282280.1282336
10. Lu, J., Liou, M.: A simple and efficient search algorithm for block-matching motion estimation. Circuits and Systems for Video Technology, IEEE Transactions on **7**(2), 429–433 (1997). DOI 10.1109/76.564122
11. Mohan, R.: Video sequence matching. In: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, vol. 6, pp. 3697–3700 vol.6 (1998). DOI 10.1109/ICASSP.1998.679686
12. Tasdemir, K., Cetin, A.: Motion vector based features for content based video copy detection. In: Pattern Recognition (ICPR), 2010 20th International Conference on, pp. 3134–3137 (2010). DOI 10.1109/ICPR.2010.767
13. Teodosio, L., Bender, W.: Salient video stills: content and context preserved. In: MULTIMEDIA '93: Proceedings of the first ACM international conference on Multimedia, pp. 39–46. ACM, New York, NY, USA (1993). DOI http://doi.acm.org/10.1145/166266.166270
14. Tonomura, Y., Abe, S.: Content oriented visual interface using video icons for visual database systems. In: Visual Languages, 1989., IEEE Workshop on, pp. 68–73 (1989). DOI 10.1109/WVL.1989.77044
15. Yeung, M.M.Y.: Analysis, modeling and representation of digital video. Ph.D. thesis, Princeton, NJ, USA (1996)
16. Zhang, H., Kankanhalli, A., Smoliar, S.W.: Automatic partitioning of full-motion video. Multimedia Syst. **1**(1), 10–28 (1993). DOI http://dx.doi.org/10.1007/BF01210504