CONTENT BASED VIDEO COPY DETECTION USING MOTION VECTORS

A THESIS

SUBMITTED TO THE DEPARTMENT OF ELECTRICAL AND

ELECTRONICS ENGINEERING

AND THE INSTITUTE OF ENGINEERING AND SCIENCE

OF BİLKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS

FOR THE DEGREE OF

MASTER OF SCIENCE

By

Kasım Taşdemir

August 2009

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. A. Enis Çetin(Supervisor)

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Prof. Dr. Volkan Atalay

I certify that I have read this thesis and that in my opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Dr. Onay Urfalıoğlu

Approved for the Institute of Engineering and Science:

Prof. Dr. Mehmet Baray
Director of the Institute

ABSTRACT

CONTENT BASED VIDEO COPY DETECTION USING MOTION VECTORS

Kasım Taşdemir
M.S. in Electrical and Electronics Engineering
Supervisor: Prof. Dr. A. Enis Çetin
August 2009

In this thesis, we propose a motion vector based Video Content Based Copy Detection (VCBCD) method. Detecting the videos violating the copyright of the owner comes into question by growing broadcasting of digital video on different media. Unlike watermarking methods in VCBCD methods, the video itself is considered as a signature of the video and representative feature parameters are extracted from a given video and compared with the feature parameters of a test video. Motion vectors of image frames are one of the signatures of a given video. We first investigate how well the motion vectors describe the video.

We use Mean value of Magnitudes of Motion Vectors (MMMV) and Mean value of Phases of Motion Vectors (MPMV) of macro blocks, which are the main building blocks of MPEG-type video coding methods. We show that MMMV and MPMV plots may not represent videos uniquely with little motion content because the average of motion vectors in a given frame approaches zero.

To overcome this problem we calculate the MMMV and MPMV graphs in a lower frame rate than the actual frame rate of the video. In this way, the motion vectors may become larger and as a result robust signature plots are

obtained. Another approach is to use the Histogram of Motion Vectors (HOMV) that includes both MMMV and MPMV information.

We test and compare MMMV, MPMV and HOMV methods using test videos including copies and the original movies.

Keywords: Content Based Copy Detection, Similar Video Detection, Motion Vectors, Sequence Matching, Video Copy Detection

ÖZET

HAREKET VEKTÖRLERİ İLE İÇERİK TABANLI KOPYA VİDEO SEZİMİ

Kasım Taşdemir
Elektrik ve Elektronik Mühendisliği Bölümü Yüksek Lisans
Tez Yöneticisi: Prof. Dr. A. Enis Çetin
Ağustos 2009

Bu tez çalışmasında, hareket vektörleri tabanlı bir İçerik Tabanlı Kopya Video Sezim (İTKVS) metodu önerilmektedir. Sayısal videoların farklı ortamlardaki yayınının giderek artması, telif haklarını ihlal eden videoların tespit edilmesi işini gündeme getirmiştir. İTKVS yönteminde, gizli damgalama yöntemlerinden farklı olarak videonun kendisi bir imza kabul edilmektedir ve temsili öznitelik parametreleri çıkartılarak test videosunun öznitelik parametreleriyle karşılaştırılmaktadır. Resim çerçevelerinin hareket vektörleri, videoya ait imzalardan biridir. Öncelikle, hareket vektörlerinin bir videoyu ne kadar iyi temsil edebileceğini incelemekteyiz.

MPEG türündeki video kodlamalarının yapı taşı olan hareket vektörlerini kullanarak Hareket Vektörlerinin Büyüklüklerinin Ortalama Değerini (HVBO) ve Hareket Vektörlerinin Açılarının Ortalama Değerini oluşturmaktayız. HVBO ve HVFO grafiklerinin, az hareket içeren videoları temsil edemeyebileceğini, çünkü hareket vektörlerinin ortalamasının sıfıra yaklaştığını göstermekteyiz.

Bu sorunu aşmak için HVBO ve HVFO grafiklerini asıl çerçeve hızından daha düşük çerçeve hızında hesaplanmıştır. Bu şekilde hareket vektörleri daha

büyük hale gelebilir ve sağlam video imza grafiği elde edilir. Diğer bir yaklaşım ise HVBO ve HVFO bilgilerini beraber kullanan Hareket Vektörleri Histogramı (HVH) yöntemidir.

HVBO, HVFO ve HVH yöntemleri, asıl ve kopya videoları içeren test videolarıyle test edilmiş ve karşılaştırılmıştır.

Anahtar Kelimeler: İçerik Tabanlı Kopya Sezimi, Benzer Video Sezimi, Hareket Vektörleri, Dizi eşleme, Kopya Video Sezimi

ACKNOWLEDGMENTS

I gratefully thank my supervisor Prof. Dr. Enis Çetin for his supervision, guidance and suggestions throughout the development of this thesis. He was much more than a supervisor.

I would also like to thank Prof. Dr. Volkan Atalay, and Dr. Onay Urfalıoğlu for reading, commenting, and making useful suggestion on my thesis.

It is a pleasure to express my special thanks to my family for their love, support and encouragement throughout my life.

Many thanks to all of my close friends for their help and friendship throughout all these years. Special thanks to Uğur Toreyin, Osman Günay, Fatih Erden, Ahmet Güngör, Serdar Çakır, Hakan Habiboğlu, İhsan İnaç, Akif Taşdemir and M. Yasin Siviş.

I would also like to thank TÜBİTAK for providing financial support throughout my graduate study.

Contents

List of Figures

List of Tables

To my supervisor Prof. Dr. Enis Çetin ...

Chapter 1

INTRODUCTION

Detecting the videos violating the copyright of the owner comes into question by growing broadcasting of digital video on different media. Digital videos are distributed on TV channels, web-tv, video blogs and public video servers. There is a huge amount of videos in various databases already shared and sharing speed is also increasing day by day. This makes the tracing of video content a very hard problem. Also, it is hard to control the copyright of a huge number of videos uploaded everyday for the owner of popular video web server companies. Content based copy detection (CBCD) is an alternative way to watermarking approach to identify the ownership of video. CBCD and watermarking are two approaches that are used for protection of the copyright. In watermarking methods, non-visual information is inserted into the video sequence that can be retrieved later and analyzed [1] -[4]. However, there is no sufficiently robust watermarking algorithm yet [5]. In contrast, CBCD considers video itself as a watermark. Existing methods of CBCD usually extract signatures or fingerprints from images of video stream and compare them with the database which contains features of original videos [6]. Several spatial or temporal features of videos are considered as signatures of videos such as intensity of pixels, color histograms and motion [5, 7]. The main advantage of CBCD over watermarking is that signature extraction

can be done even if the video is distributed because the unique signature is the video itself.

In CBCD algorithms, video color, intensity or motion are used as features or in feature vectors. Each feature has advantages over others. If 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. This thesis investigates how well motion vectors describe a video and proposes a new spatio-temporal video feature. Proposed motion based feature parameters are used as a CBCD feature and experimental results are presented.

Motion information was considered as a weak parameter by other researchers [7]. This is true when the motion vectors are extracted from consecutive frames. In a typical 25 Hz captured video most motion vectors are very small and they may not really contain any significant information. On the other hand, when we select large motion vectors as representive of the video we get a reliable feature set representing a given video. In Chapter 3, we present the new approach based on significantly large motion vectors and we present another method based on motion vectors computed by resampling the video with a lower fps. In this way, motion vectors (MVs) become significantly large and they clearly represent a video.

Chapter 2

RELATED WORKS

The CBCD methods are different in terms of the features they use. Most of the earlier video matching schemes reduce the video sequence into a small set of key-frames [8],[9],[10] then they use an image sequence matching method to match the key frames [11]. These algorithms have important drawbacks. One of the problems is that the process may fail when a shot is missed. Secondly, choosing the key frame which will be used as the representation of the shot is not a clearly solved step [12]. The most important drawback of these algorithms is that they ignore the temporal behavior of the video. This drawback was noticed by Kobla *et al.* in [13] and they include some motion information with spatial information.

Spatio-temporal features seem to be more robust and immune to digital and encoding distortions. Mohan [14] uses temporal activities of the videos in order to find the video pairs. It extracts "actions" from videos and uses them as fingerprints. Then it applies a sequence matching technique to find the pair of the video from the fingerprint database. Mohan defines an "action" as a pattern of activity occurring over a period of time. In order to define an action, they reduce the intensity image of each i^{th} frame to 3×3 blocks. They compute

the ordinal measure of frames by taking the average of intensity of each block into an array $y(i)$. Finally, they construct a fingerprint vector consisting of $y(i), y(i+1), ..., y(i+n)$. In order to compare two videos X and Y, they compare fingerprints of videos $[x(i), x(i+1), ..., x(i+n)]$ and $[y(i), y(i+1), ..., y(i+n)]$ using Euclidean distance. Kim and Vasudev [15] improve this method by using different block sizes.

The color histogram of a frame is another feature that is used by some of the researchers [16,17]. Satoh [16] uses color histogram for matching shots and and also for detecting shot-boundaries. Yeh and Cheng [17] propose a fast method that is 18× faster than other algorithms for sequence matching. They use an extended HSV color histogram.

Some video similarity detection methods use uncompressed MPEG video to directly extract the features. Content of the frames, DC values of macro blocks or motion vectors are used as features. Ardizzone *et al.* [18] use motion vectors for feature extraction. They use global motion feature or motion based segmented feature as a signature of the video. In global motion extraction step, statistical distribution of directions (i.e., an angle histogram) is calculated. The angle histogram is computed by dividing the [-180°,180°] interval into subintervals. Sum of magnitudes of motion vectors in intervals constructs the angle histogram. In motion based segmentation, motion vectors are clustered and labeled. Labels are given according to the similarity of motion vectors or the histogram of motion vector magnitudes. Dominant regions are taken into account in comparison step.

Joly, Frelicot and Buisson extract local fingerprints around interest points in [19]. These interest points are detected with the Harris detector and compared using the Nearest Neighbor method. They propose statistical similarity search in [20],[21]. Joly *et al.* use this method and propose distortion-based probabilistic approximate similarity search technique in order to speed up scanning in content based video retrieval framework [22].

Zhao *et al.* extract PCA-SHIFT descriptors and use it for video matching in [23]. They use the nearest neighbor search for matching and SVMs for learning matching patterns with their duplicates. Law *et al.* propose a video indexing method using temporal contextual information which is extracted from local descriptors of interest points in [24][25]. They use this contextual information in a voting function.

Poullot *et al.* present a method for monitoring a real time TV channel in [26]. They use the method for comparing the incoming data with indexed videos in database. Innovations of the method are z-grid for building indexes, uniformity-based sorting and adapted partitioning of the components.

Lienhart *et al.* [27] use color coherence vector to characterize the key frames of the video. Sanchez *et al.* [28] discuss using color histograms of key frames for copy detection. They test the developed system on TV commercials and the system is sensitive to color variations. Hampapur [29] uses edge features but he ignores the color variations. Indyk *et al.* [30] use distance between two scenes as its signature. However, it is a weak and limited signature. Naphade *et al.* [31] use histogram intersection of the YUV histograms of the DC sequence of the MPEG video. It is an efficient method in terms of compression. Küçüktunç proposes a multimodal framework for matching video sequences [32]. First, he matches the faces in the frames then he matches the non-facial shots using low-level visual features.

2.1 Perceived Motion Energy Spectrum Based Shot Retrieval

Motion information is an important feature of video for human perception. Ma *et al.* [33] describe a way to imitate the human perception. In the paper, perceived

motion based shot content representation, namely, *perceived motion energy spectrum* ($PMES$) is proposed for content-based video retrieval. With this method human perceived movements can be distinguished. $PMES$ is constructed by using a temporal filter to eliminate disregarded object motions and a global motion filter to discriminate object motions from camera motions.

In a video there are human regarded and disregarded object motions. In most cases camera motion such as pan, zoom etc are disregarded motions by a human. In light of human perception behavior information, we can say that it would be better if the object motion and the camera motion are used separately instead of single dominant motion. The proposed method in this paper matches with human's perception well, and avoids object segmentation and global motion estimation which are all difficult tasks.

2.1.1 Perceived Motion Energy Spectrum

There are two or one motion vectors in each macro block of MPEG stream, often referred as motion vector field (MVF). Magnitude of the vector corresponds the moving speed of the object in the scene, so it can used to compute the energy of motion region or object at macro block scale if atypical samples are removed. Humans can perceive an object better if its motion intensity and its appearance duration are high. So, motion energy of a macro block at position (i, j) can be considered as the average of motion magnitudes of motion block at position (i, j) over its appearance duration.

Angle information of motion vectors are not reliable as magnitudes. Nevertheless, we can say that if camera movement such as panning is the case, motion vector angles of macro block at position (i, j) should point to one direction. So, if there is a consistency in the direction of the motion vectors in temporal domain, this means that camera movement is dominant to object movement. PMES

depends on the mentioned two assumptions. In $PMES$, a **temporal energy filter** which accumulates the energy along the temporal axis and a **global motion filter** which extracts actual object motion energy is used. Thus, $PMES_{i,j}$ forms $PMES$ image.

2.1.2 Temporal Energy Filter

The atypical motion vectors usually result in inaccurate energy accumulations. Before computing the $PMES$ images, atypical motion vectors are eliminated by using a modified median filter in spatial domain. $Mag_{i,j}$ corresponds to magnitude of motion vector of macro block at position (i, j) $MB_{i,j}$. The elements in the filter's window at macro block $MB_{i,j}$ are denoted by $\Omega_{i,j}$ in MVF, W_s is the width of window. The filter magnitude of motion vector is computed by

$$Mag_{i,j} = \begin{cases} Mag_{i,j} & (\text{if } Mag_{i,j} \leqslant Max4th(Mag_k)) \\ Max4th(Mag_{i,j}) & (\text{if } Mag_{i,j} > Max4th(Mag_k)) \end{cases} \tag{2.1}$$

where $(k \in \Omega_{i,j})$, and the function $Max4th(Mag_k)$ returns the fourth value in the descending sorted list of magnitude elements $\Omega_{i,j}$ in the filter window. Then a temporal energy filter is applied to each spatial filtered magnitudes at macro block position (i, j) along a time duration of L_t. Thus, 3-D spatio-temporal tracking volume with spatial size of W_t^2 and the temporal duration of L_t is constructed. Each magnitude for each macro block in the tracking volume are sorted in a list along the duration side of volume. The temporal filter trims the magnitude list from both sides with an amount determined by α. Rest of the elements of list are averaged and considered as the mixture energy. "Mixture" means that it contains both camera motion energy and object motion energy. Mixture energy is denoted by 2.2.

$$MixEn_{i,j} = \frac{1}{(M - 2\lfloor \alpha M \rfloor W_t^2)} \sum_{m=\lfloor \alpha M \rfloor + 1}^{M - \lfloor \alpha M \rfloor} Mag_{i,j}(m) \tag{2.2}$$

where M is the total number of magnitudes in tracking volume, and $\lfloor \alpha M \rfloor$ equals to the largest integer not greater than αM; and $Mag_{i,j}(m)$ is the magnitude value in the sorted list of tracking volume. The trimming parameter $\alpha(0 \leqslant \alpha \leqslant 0.5)$ controls the number of data samples excluded from the accumulating computation. Then, mixture energy is normalized into range [0,1] as defined by 2.3 in order to form motion energy spectrum

$$\overline{MixEn_{i,j}} = \begin{cases} MixEn_{i,j}/\tau & (\text{if } En_{i,j}/\tau \leqslant 1) \\ 1 & (\text{if } En_{i,j}/\tau > 1) \end{cases} \tag{2.3}$$

A reasonable truncation threshold τ is selected easily according to encoded parameter in a MPEG stream.

2.1.3 Global Motion Filter

Perceived motion or actual object motion is extracted from mixture energy $\overline{MixEn_{i,j}}$ by filtering with global motion filter. Camera motions have distinctive behavior. When camera moves or changes its direction the macro block $MB_{i,j}$ has similar motion vector angles over a time duration. So, probability distribution function of angle of motion vectors of macro blocks over tracking volume can be considered as a clue for camera motion. The consistency of angle of motion vector in tracking volume can be measured by entropy. The normalized entropy reflects the ratio of camera motion to object motion. Higher entropy corresponds to poor consistency of angle. PDF of angle variation can be obtained from normalized angle histogram. Angle of a motion vector is in range $[0, 2\pi]$. This range is divided in to n angle range. Angles in each range are accumulated for each macro block over tracking volume. Thus, an angle histogram with n bins is formed for each $MB_{i,j}$, denoted by $AH_{i,j}(t), t \in [1, n]$. The probability distribution function $p(t)$ is defined as 2.4.

$$p(t) = AH_{i,j} / \sum_{k=1}^{n} AH_{i,j}(k) \tag{2.4}$$

Using 2.4, the angle entropy $AngEn_{i,j}$ can be computed as following

$$AngEn_{i,j} = -\sum_{t=1}^{n} p(t) \log p(t) \tag{2.5}$$

where the value range of $AngEn_{i,j}$ is $(0, \log n]$. $AngEn_{i,j}$ reaches its maximum value when $p(t) = 1/n, t \in [1, n]$. In the paper, normalized angle entropy is considered as a ratio of global motion, denoted by $GMR_{i,j}$,

$$GMR_{i,j} = AngEn_{i,j} / \log n \tag{2.6}$$

where $GMR_{i,j} \in (0, 1]$. Camera motion becomes dominant in the mixture energy $\overline{MixEn_{i,j}}$ when $GMR_{i,j}$ approaches to 0. In order to emphasize the object motions $GMR_{i,j}$ is used as a scaling number.

2.1.4 Generating PMES Images

Since we know the motion energy of a macro block and camera/object motion ratio, we can create an image of moving objects with their motion energies. In order to reduce the effect of camera motion vectors, since these are ignored by human as mentioned before, $\overline{MixEn_{i,j}}$ is scaled by $GMR_{i,j}$. The definition is as follows

$$PMES_{i,j} = GMR_{i,j} \times \overline{MixEn_{i,j}} \tag{2.7}$$

After quantizing $PMES_{i,j}$ values at each macro block into 256 levels, a gray level $PMES$ image is generated. Dark regions in the image correspond to no motion or camera motion dominant regions and light regions correspond to object motions. Intensity of the image denotes the magnitude of the object motion.

2.1.5 PMES Images Based Shot Comparison

The paper proposed a comparison method for $PMES$ images. Images are segmented into $m \times n$ panes. Then, normalized energy histograms with $m \times n$ bins

are constructed by averaging $PMES_{i,j}$ in each pane respectively, denoted by EH(p),

$$Sim(q,s) = \frac{\sum_{k=1}^{m\times n} \min EH_q(k), EH_s(k)}{\sum_{k=1}^{m\times n} \max EH_q(k), EH_s(k)} \tag{2.8}$$

where $Sim \in [0,1]$ and $Sim = 1$ indicates that two shots are most similar to each other.

In [34] PME is used for extracting key frames of a video. They assume that most salient visual content is the best candidate for being the key frame. So, a kind of motion activity map is introduced as triangle model and the frames at the top of triangles are selected as key frames. PME is average magnitude of motion vectors $Mag(t)$ in a frame scaled by probability of most significant angle of motion vectors in that frame $\alpha(t)$. It is defined as follows,

$$PME(t) = Mag(t) \times \alpha(t). \tag{2.9}$$

where

$$\alpha(t) = \frac{\max\left(AH(t,k), k \in [1,n]\right)}{\sum_{k=1}^{n} AH(t,k)}. \tag{2.10}$$

$$Mag(t) = \frac{\left(\frac{\sum Mix_{FEn_{i,j}(t)}}{N} + \frac{\sum Mix_{BEn_{i,j}(t)}}{N}\right)}{2} \tag{2.11}$$

$Mix_{FEn_{i,j}(t)}$ and $Mix_{BEn_{i,j}(t)}$ are forward and backward motion vector energies calculated similar to $MixE_{n_{i,j}(t)}$ in 2.2

Chapter 3

Video Copy Detection Using Motion Vector Features

Motion vector information can be used as a signature of the video because each video has its own characteristic motion vector patterns. Section 3.2 investigates the uniqueness of the motion vector patterns of movie frames with some example movie scenes and their corresponding motion vector related data. Section 3.4 analyzes the similarity between the mean of the magnitude and the phase of motion vectors data of the original movie and the artificially distorted or re-recorded movie with a camera recorder. Experimental results are also presented. Section 3.5 proposes a method, histogram of motion vectors, that uses both the magnitude and the phase information as a feature of a given video and uses it in the content based copy detection problem.

3.1 Motion Vector Extraction

Motion vectors are extracted using motion estimation algorithms. Motion estimation plays an important role in almost all video compression and transmission

methods including the MPEG-family of coding methods [35]. In this thesis, we used a simple and efficient search (SES) algorithm [36] and an exhaustive search (ES) [37] for block matching.

Block matching is performed on the current frame (t) and the previous frame (t-1). The current frame is divided into square blocks of pixel size $N \times N$. 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

$$MAD(x, y) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |F_c(k + i, l + i) - F_p(k + x + i, l + y + j)| \qquad (3.1)$$

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 [36].

3.1.1 Exhaustive Search Algorithm

Another name of this algorithm is the Full Search algorithm. This is the most computationally expensive block matching 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 [37]. This algorithm is straightforward to implement and gives the best results. The disadvantage of this algorithm is its high computational cost.

3.1.2 A Simple and Efficient Search Algorithm

This algorithm is a modified version of the three step search (TSS) algorithm [36],[37]. In the TSS algorithm a block is searched in some reference points of

locations in the previous frame instead of searching all possible locations. An example TSS procedure is shown in Fig. 3.1 for $W = 7$. 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. In our case, the motion vector of this macro block is $(3, 7)$.

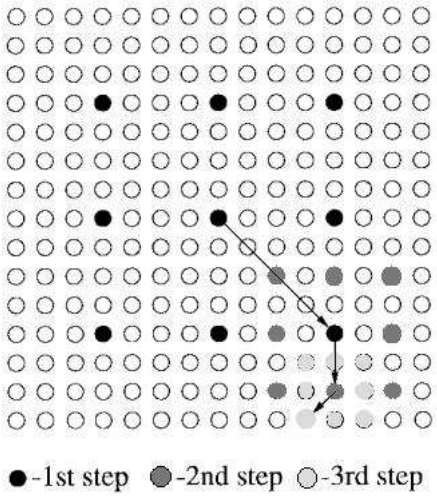


Figure 3.1: The TSS procedure for ($W = 7$).

The TSS assumes that frames have unimodal error surface which means that the block matching error decreases monotonically as the search is along the global minimum error direction. Simple and efficient (SES) block matching algorithm claims that checking all points in the TSS algorithm is unnecessary when the surface has unimodal error. We use this algorithm in Chapter 3

3.1.3 A Modified Motion Vector Extraction Algorithm

In general, motion vectors are extracted using consecutive frames. If the video is recorded in high fps and the movements in the video are relatively slow, which is

a typical case, motion vectors have low magnitudes. As a result, motion vector dependent feature of a video which has low motion vector magnitudes is not a strong representation of the video. As it is described in Section 3.2 it affects the accuracy of the CBCD comparison results. However, temporal behavior of a video is an important feature of the video. We propose a motion vector extraction algorithm to increase the motion vector magnitudes. In the traditional approach, motion vectors are extracted using i^{th} and $(i+1)^{th}$ frame. In our approach, we use every i^{th} and $(i+n)^{th}$ frame for motion vector extraction. An example of the algorithm is shown in Fig. 3.2(b) where n is 5.

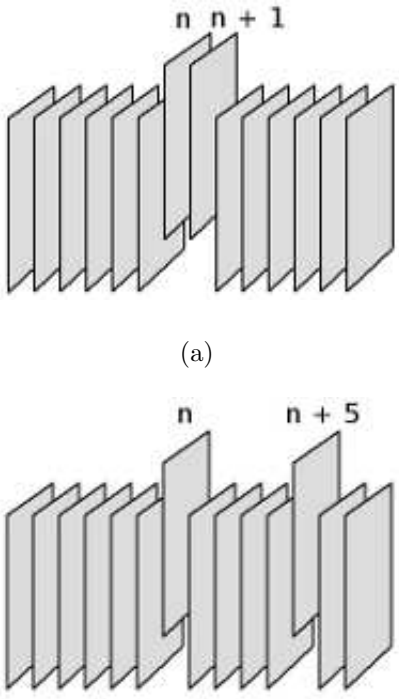


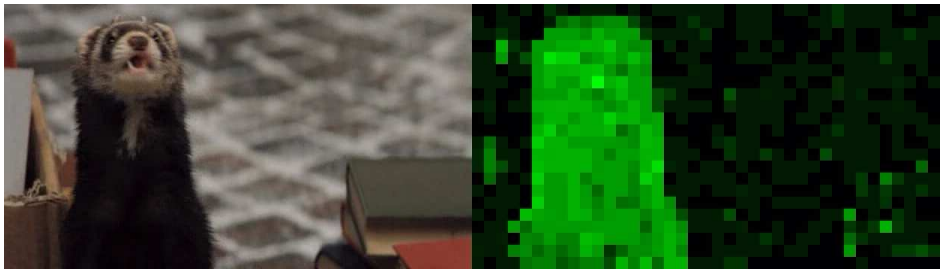
Figure 3.2: (a) Motion vector extraction algorithms use the current and the next frame; (b) The current and the $(n+5)^{th}$ frame is used in this thesis.

This method increases the size of the motion vectors because we sample the video in a lower fps than the original fps.

3.2 Motion Vectors as a Signature of Video

Sports videos, documentaries, surveillance camera recordings etc. have different nature. Each video has its own specific motion patterns. Therefore, motion vectors of macro blocks contain a descriptive information about the video. Spatial characteristics of motion vectors of some videos are shown in Fig. 3.2. For instance, there are small and slightly changing movements in a video of an anchorman talking in front of a stationary background as in Fig. 3.3(a). Moving blocks of the video are marked on the right hand side of Fig. 3.2. When a large object is moving as in Fig. 3.3(b) significant motion vectors appear in the corresponding area of motion vector magnitude graph. However, videos recorded from moving cameras have a dense motion vector field because all macro blocks slide into different places. As it is seen from the motion vector map of Fig. 3.3(c), the field corresponding to a dog and a man has less motion because the camera is tracing the running dog and the man. Thus, motion vectors are descriptive features representing the video as each video has its own specific motion vector field behavior in both spatial and temporal domains.

(a)



(b)

(c)

Figure 3.3: Left half of the images are one image frame of the video. Right half of the images are magnitude image of corresponding motion vectors of 16x16 macroblocks. (a) a salesman presenting a device with slow hand gestures, (b) a weasel moving its body in front of a stationary camera, and (c) a dog and a trainer runing while the camera tracks them.

Temporal behavior of motion vectors also contains unique signatures. In Figures 3.4, 3.5 and 3.6, each element of the plotted data is the mean of the magnitudes of motion vectors (MMMV) of macro blocks of a corresponding frame. The MMMV is defined as follows:

$$MMMV(k) = \frac{1}{N} \sum_{i=0}^{N-1} r(k,i) \tag{3.2}$$

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.

The video of "Salesman", has low motion content and the MMMV plot has slight variations as shown in Fig. 3.4. The first half of the movie "Inkheart" contains high motion activity scene. After the 110th frame the camera view changes to a still scene as shown in Fig 3.5. So, each movie has a unique motion behavior temporally and this property can be used for content based copy detection or video indexing and searching algorithms.

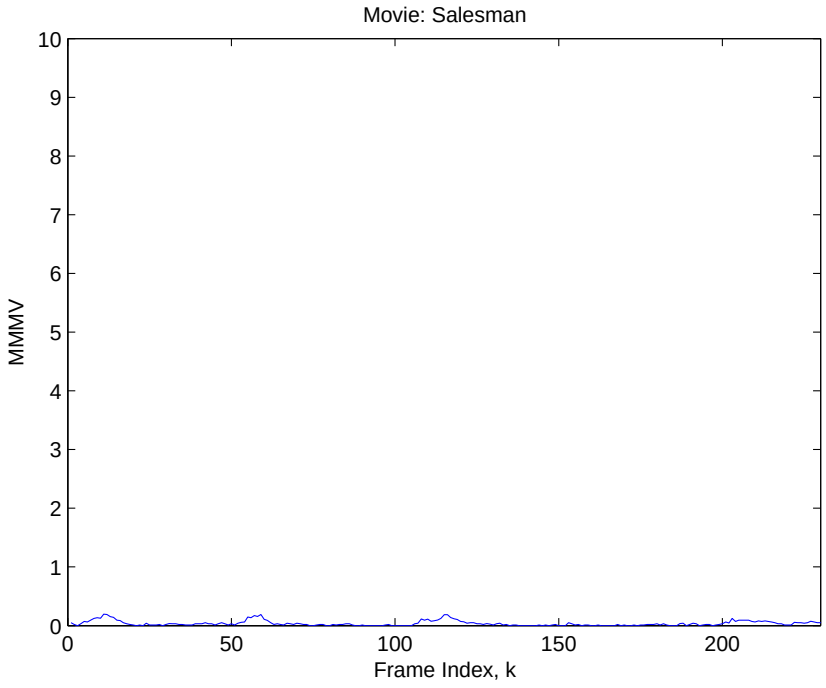


Figure 3.4: The MMMV plot of "Salesman". Magnitudes of motion vectors are small as there is a single person who only moves his lips and hands in the movie.

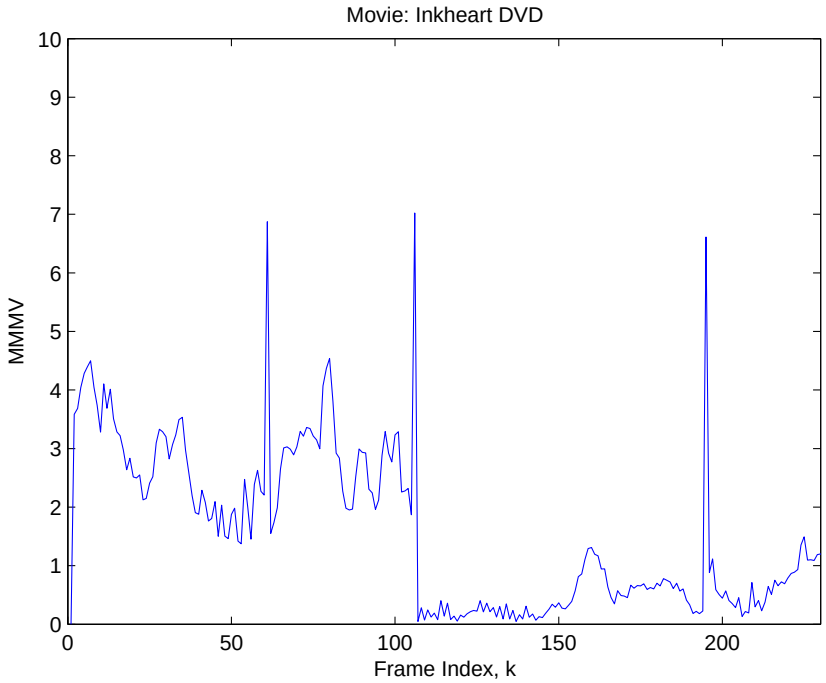


Figure 3.5: The MMMV plot of "Inkheart". First 110 frames of the movie has a high motion activity, rest shows that there are only small motions in the scene.

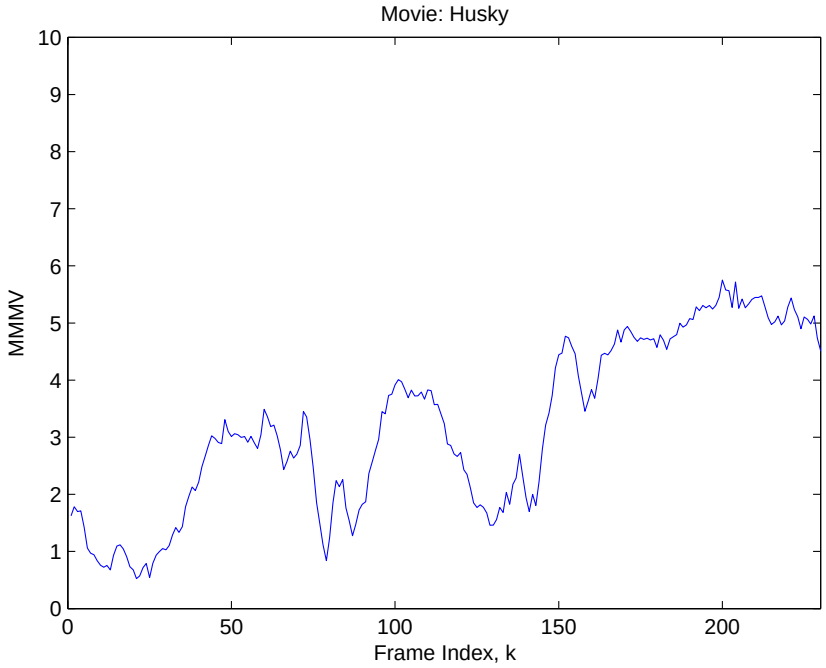


Figure 3.6: The MMMV plot of "Husky". A moving camera is tracking the running dog and man.

Previous plots show examples of temporal motion vector behaviors of different videos. Motion vectors of macro blocks of a movie also contain direction information which is ignored in magnitude plots. As shown in Fig. 3.7, 3.8 and 3.9 phase plots also contain unique information about a given video. The mean of the phase of motion vectors (MPMV) of macro blocks of a given frame (MPMV) are plotted in Figures 3.7, 3.8 and 3.9. The MPMV is defined as follows:

$$MPMV(k) = \frac{1}{N} \sum_{i=0}^{N-1} \theta(k, i), \tag{3.3}$$

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: $MPMV(.) \in (-\pi, \pi)$.

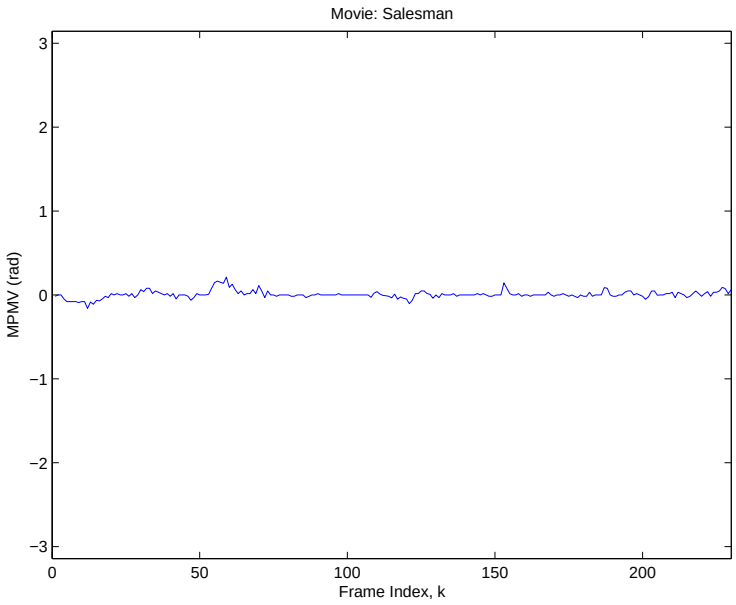


Figure 3.7: The phase angle of motion vectors MPMV are small since there are only slowly moving objects in the movie. If magnitudes of MVs of both x and y directions are 0, phase is assumed to be 0.

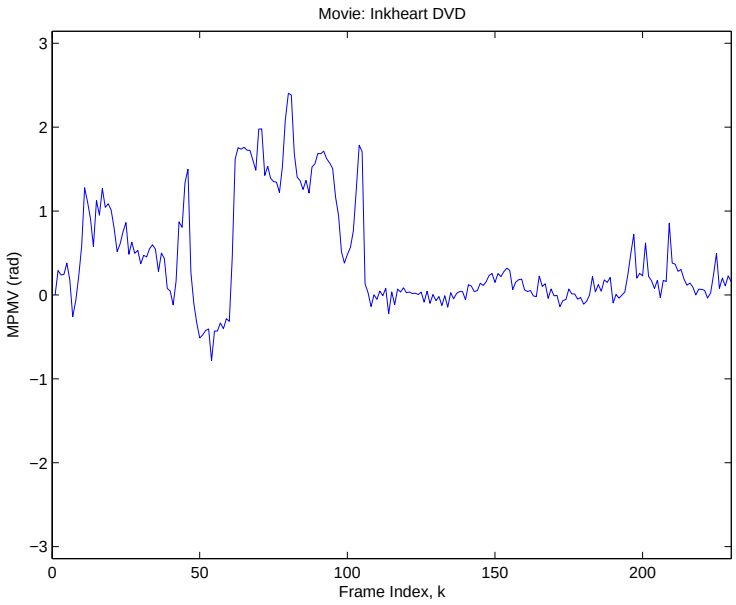


Figure 3.8: In the middle frames of the movie, most of the macro blocks tend to move one direction which is due to a camera motion. There is no significant phase information in the rest of the movie.

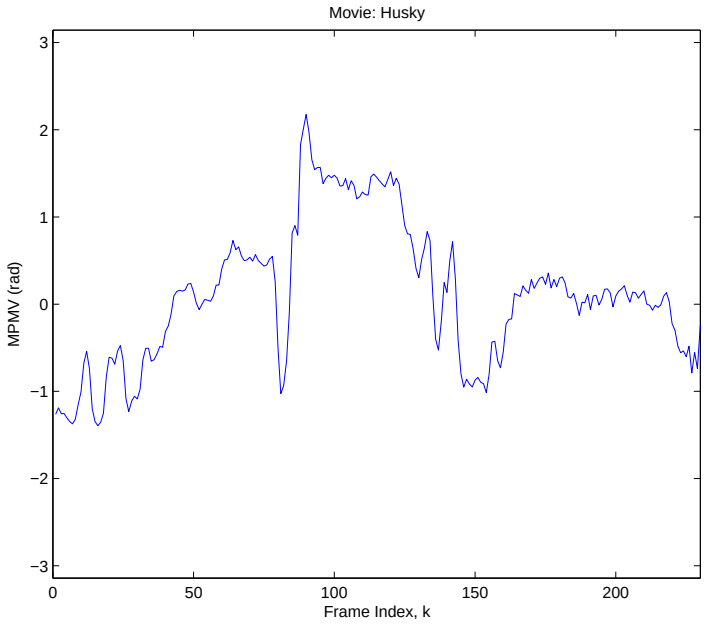


Figure 3.9: The MMMV plot of video "Husky". Since camera is tracking the running dog and the man, phase plot has a rise at frame 78 from -1 to 2 which is due to the changing flow direction of the camera.

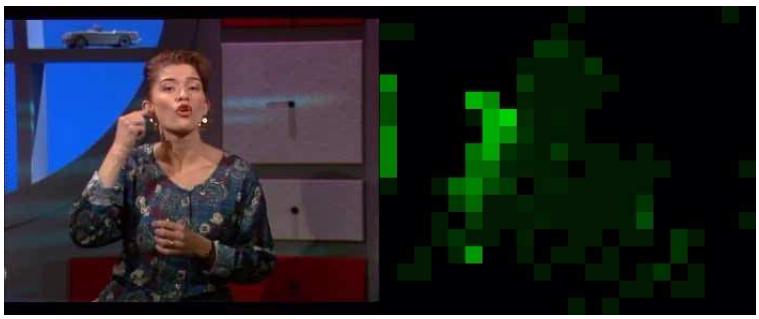
Different movies show different temporally and spatially motion vector patterns according to the camera motions or object movements in the movie. The MMMV gives information about how much there is a motion in frames and the MPMV gives information about which direction pixels tend to move in frames.

3.3 Effects of Using Modified MV Extraction Algorithm on MMMV and MPMV

If there are two videos where one of them has high motion activity and the other one has little motion activities, then its easy to distinguish them using motion vector information. In that case using MMMV for comparing them is advantageous because of the high difference of motion activities in the scenes.

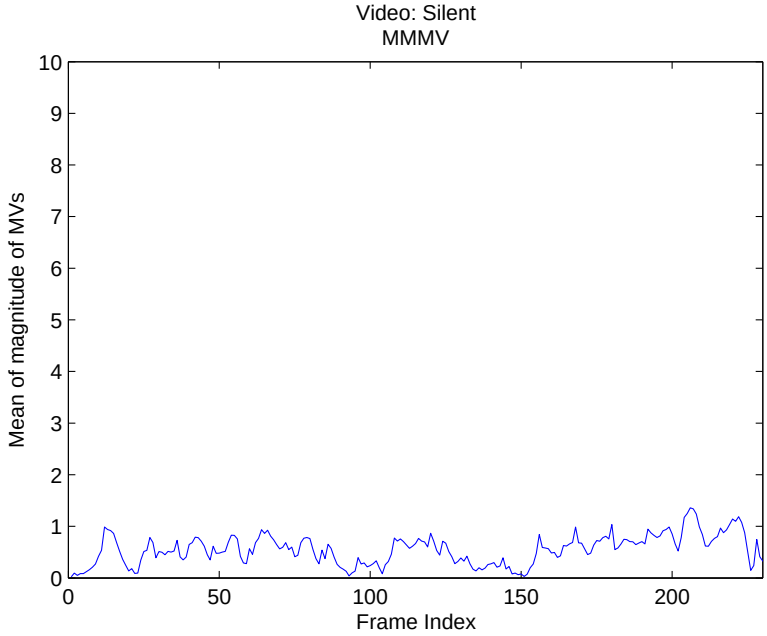
On the other hand, in the case of similar videos with respect to MMMV, such that both of them have a stationary background and slowly moving objects, it may be hard to distinguish the distorted version of the original video from the other similar candidate video. Similarity of the MMMV of two similar videos 3.10(a), 3.10(b) are shown in Fig. 3.11(a), 3.11(b).

(a)

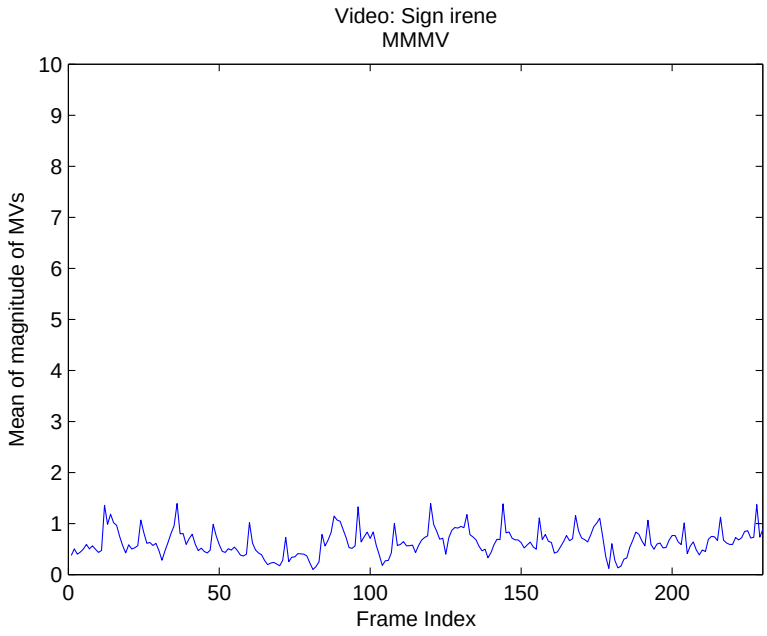


(b)

Figure 3.10: These two videos has similar motions and motion vector magnitudes are small. (a) A frame from the video "sign irene" (b) A frame from the video "silent".



(a)



(b)

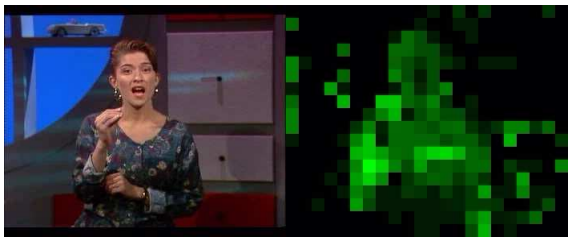
Figure 3.11: The MMMV plots of two similar movies: They have low motion activity. (a) The MMMV plot of the video "sign irene", (b) the MMMV plot of the video "silent".

In this case, increasing the amplitudes of the motion vector magnitudes will increase the difference which is a desired case for the CBCD problem. The motion vector extraction algorithm can be changed to give results with high amplitudes by the MVs from every n-th frame, $n > 1$. In general, human movements are slowly changing in one frame to next frame. 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. MMMV computed in consecutive frames in a 25 fps video may not provide robust information about a video as shown in Fig. 3.11(a) and 3.11(b). In addition, some of the 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 as shown in Fig. 3.21. Motion vectors of wall blocks appear to move in all directions in Fig. 3.21. By computing the MVs using every n-th frame ($n > 1$) it is possible to get more descriptive MMMV and MPMV plots representing a video as shown in Fig. 3.13(b) and 3.13(d). Instead of using two consecutive frames we use i^{th} and $(i + n)^{th}$ frames for MV computation and as a result, MV displacements in the video will be high. It is shown that when every 5^{th} frame is used in motion vector estimation, the moving objects are more emphasized in motion vector image as shown in Fig. 3.12 and the corresponding MMMV plots are compared in Fig. 3.13.

(a)

(b)

(c)



(d)

Figure 3.12: 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 higher than (a). (c) 51^{th} frame and its corresponding MV pattern of video "sign irene". The MVs are extracted using the next frame. The MV magnitudes are small. (d) 151^{th} frame of video "sign irene". MVs are extracted using every 5^{th} frame. MV magnitudes are higher than (c).

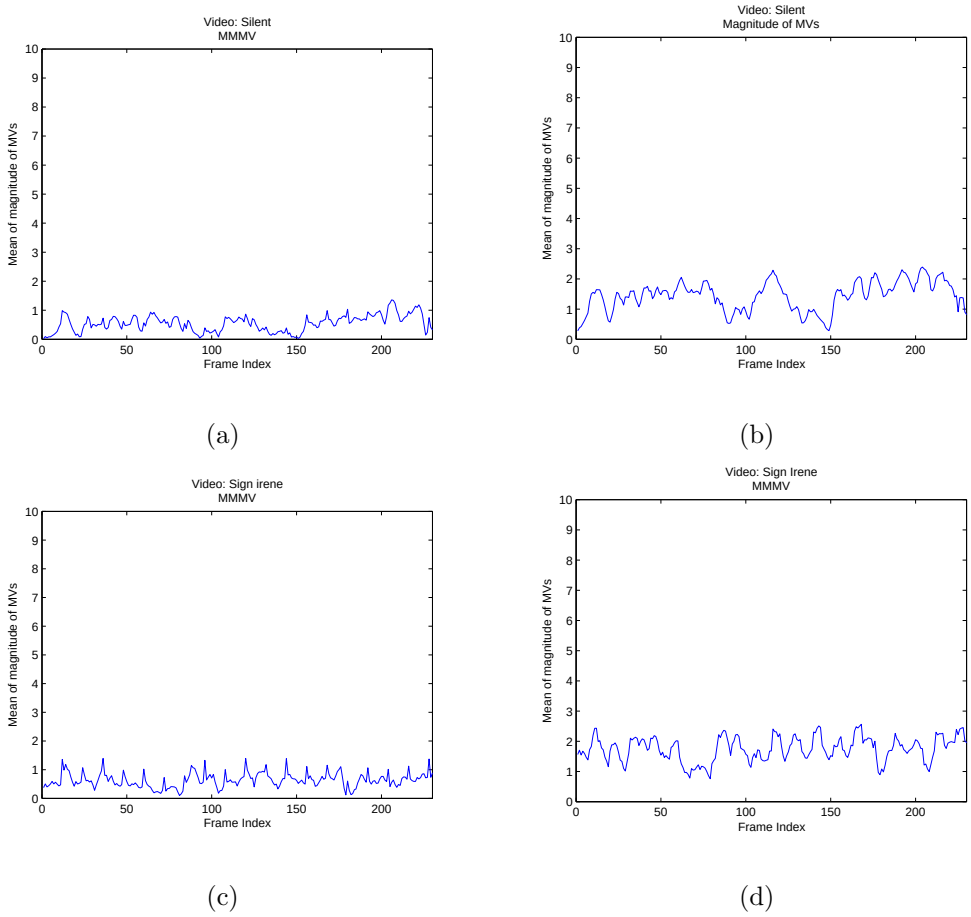
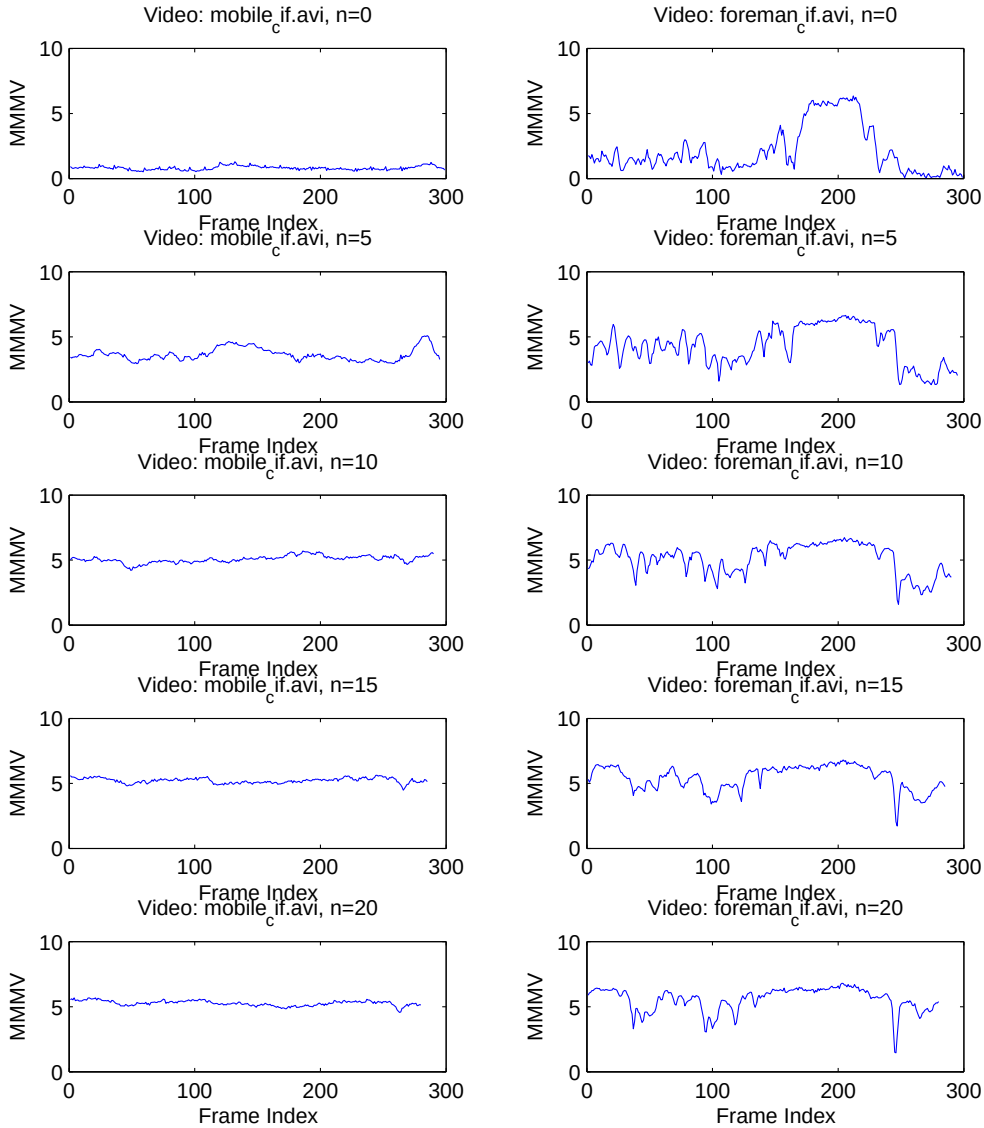


Figure 3.13: MVs are extracted using every 5^{th} frame. Thus, magnitudes of MVs are higher (a) MMMV plot of the video "Silent". MVs are extracted using next frame. (b) MMMV plot of the video "Silent". MVs are extracted using every 5^{th} frame. (c) MMMV plot of the video "Sign Irene". MVs are extracted using next frame. (d) MMMV plot of the video "Sign Irene". MVs are extracted using every 5^{th} frame.

Magnitudes of motion vectors of videos which have slow moving objects can be increased by using the modified motion vector extraction algorithm employing every n-th frame for MV computation. The MMMV of two videos which have slow moving objects for different n values are shown in Fig. 3.3.

(a) (b)



(c) (d)

Figure 3.14: Effect of using different n value in MV extraction step on the MMMV plots of two videos. (c) The MMMV of the video "Mobile.avi" (d) The MMMV of the video "Foreman.avi"

We experimentally observed that increasing n up to 5 also increases the magnitudes of MVs and MMMV of videos still represents the video well. On the other hand, MMMV of videos calculated for $n > 10$ approach to a constant value and does not represent the video because relevancy between compared frames in MV extraction step decreases as in Fig. 3.3. The average of the MMMV of several videos calculated are listed in Table 3.1. The average of MMMV of the video "Container.avi" is 0.13 for $n = 1$ which is a weak representation value for this video. It increases to 0.87 when MVs are extracted for $n = 5$. There is no point of the increasing the n value after 5 because the moving object may simply disappear from the view of the camera when large n values are used.

Table 3.1: Average values of the MMMV of some videos which have small motions. MVs are extracted for different n values.

n	Coast.avi	Container.avi	Flowers.avi	Foreman.avi	Mobile.avi
1	1.80	0.13	1.72	2.13	0.81
3	4.69	0.53	3.92	3.60	2.44
5	4.85	0.87	4.49	4.29	3.64
10	4.54	1.98	4.93	5.12	5.07
15	4.69	2.62	5.00	5.44	5.22
20	4.79	2.97	5.12	5.57	5.27

3.4 CBCD Using MMMV and MPMV

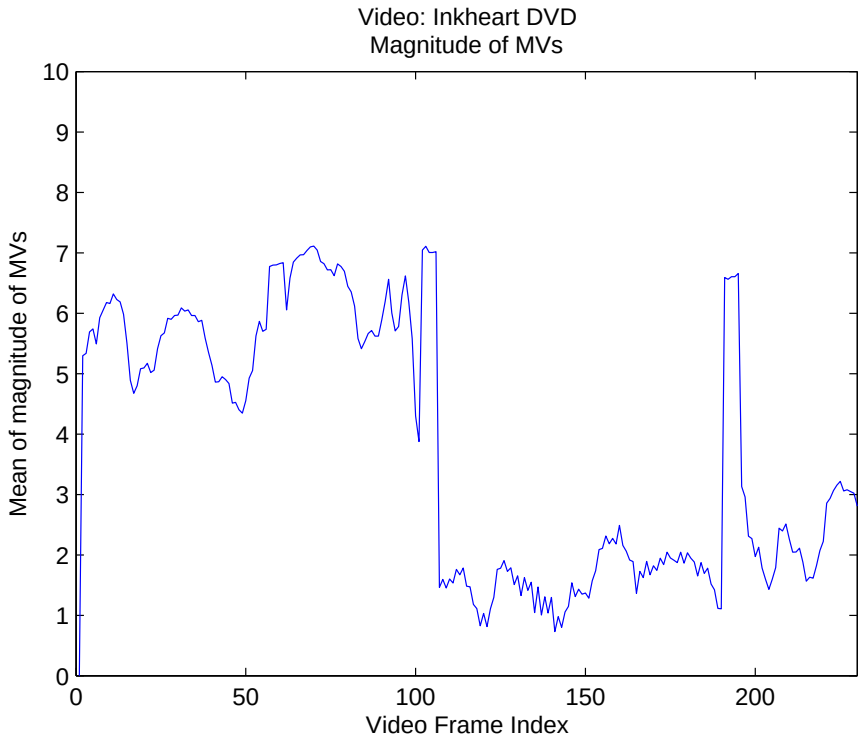
Searching and comparing the movies violating the copyright issues with official movies may not be a challenging problem if we know that the copied movie has exactly the same digital data as the original. However, in most of the cases unofficial movies are published with a small distortion or additions 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 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 based CBCD comparison methods have 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 3.2 shows the properties of the movies used in this section. Test videos have different size and fps. Videos in this section are the hardest ones in terms of matching. For more video comparison and detailed experiment results please refer to Sec. 3.7.

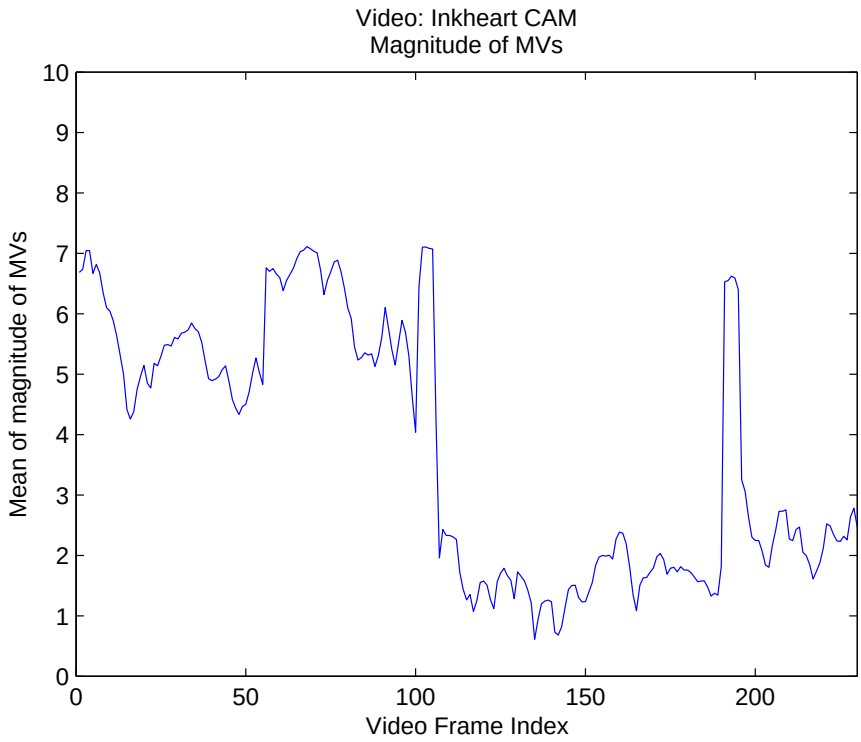
Table 3.2: Properties of original movies (with DVD extension) and the same movies recorded from a hand-held camera (with CAM extensions).

Movie Name	FPS	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

Although the original and hand-held camera recorded videos have different fps and size, they have similar MMMV plots as shown in Fig. 3.15. Original movie in Fig. 3.15(a) and its hand-held camera recorded version from a movie theater (Fig. 3.15(b)) show significant similarities. The MVs are computed with a frame difference of n=5.



(a)



(b)

Figure 3.15: Similarity of the MMMV plots of "Inkheart DVD" and "Inkheart CAM", (with n=5).

In order to obtain a value that gives information about how much two movies resemble each other, the absolute different 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.

The second reason is that frame 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:

$$\overline{MMMV} = \frac{MMMV - \mu_{MMMV}}{\sigma_{MMMV}} \tag{3.4}$$

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

The Sum of absolute values of difference of normalized MMMV values of each frame are calculated as the distance $D(a, b)$ as follows:

$$D(a,b) = \frac{1}{N} \sum_{t} \min_{|d| \leqslant W} |\overline{MMMV}_a(t) - \overline{MMMV}_b(t+d)| \tag{3.5}$$

where W is the search window width. Experimentally we select W as 2 because the fps of most commercial videos are between 20 and 30. In this thesis, unless it is stated, W is taken as 2. In Eq. 3.5, N is the number of frames in the movie $MMMV_a$. If the original and the candidate video has different fps, then their frame indices corresponding to the same time instance should be calculated

first. So, instead of comparing the frame index to frame index, the frames that correspond to same time are compared.

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. 3.16. 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. 3.17. Since the two different movies have different camera motions and object movements, their $\overline{MMMV}$ plots are not similar, $D(a, b) = 2.91$. However, distance of original video a and hand-held camera recorded video c is 0.35, $D(a, c) = 0.35$.

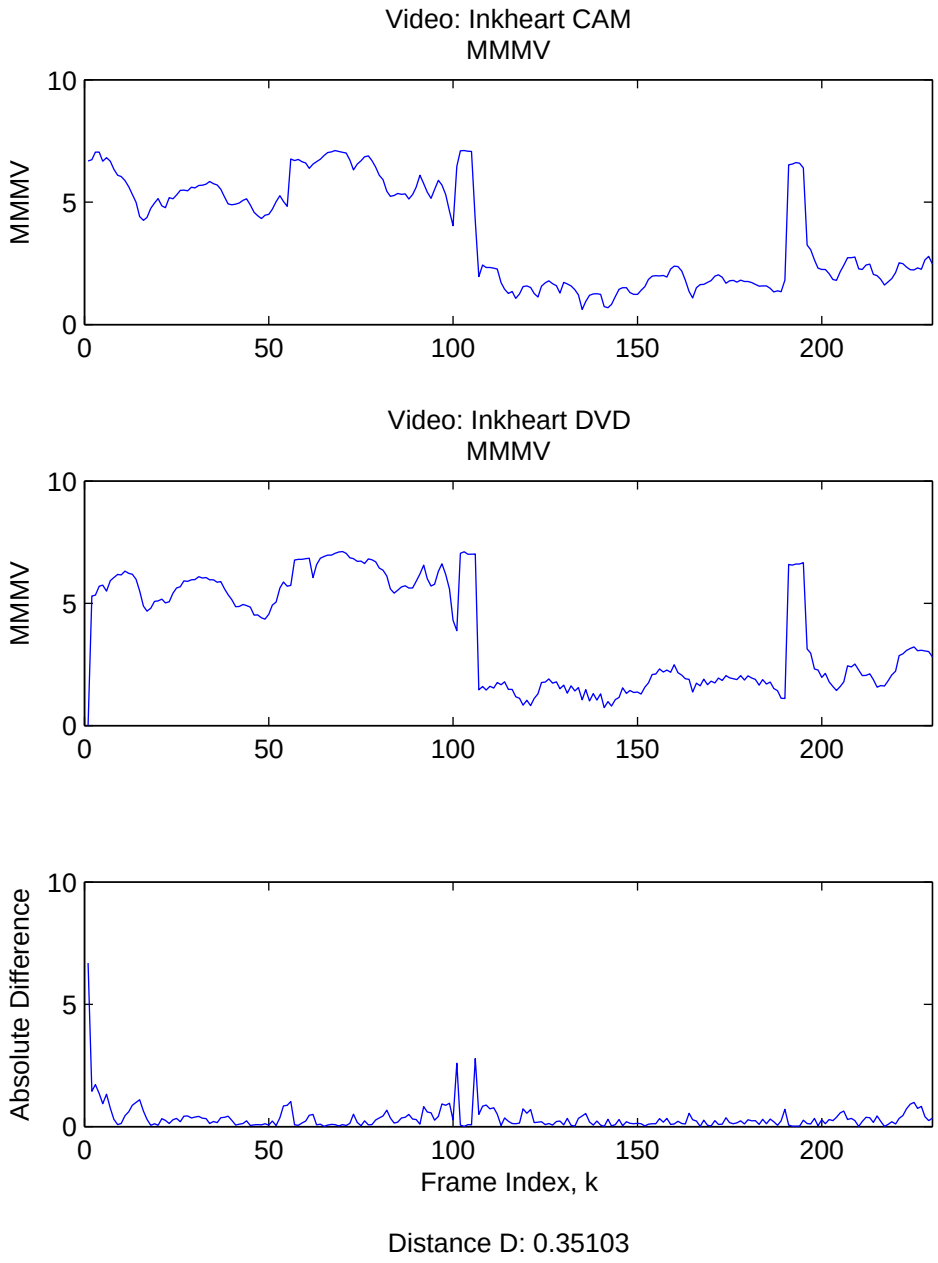


Figure 3.16: MMMV plots of videos "Inkheart DVD" and "Inkheart CAM" videos. $D(a, c) = 0.35$.

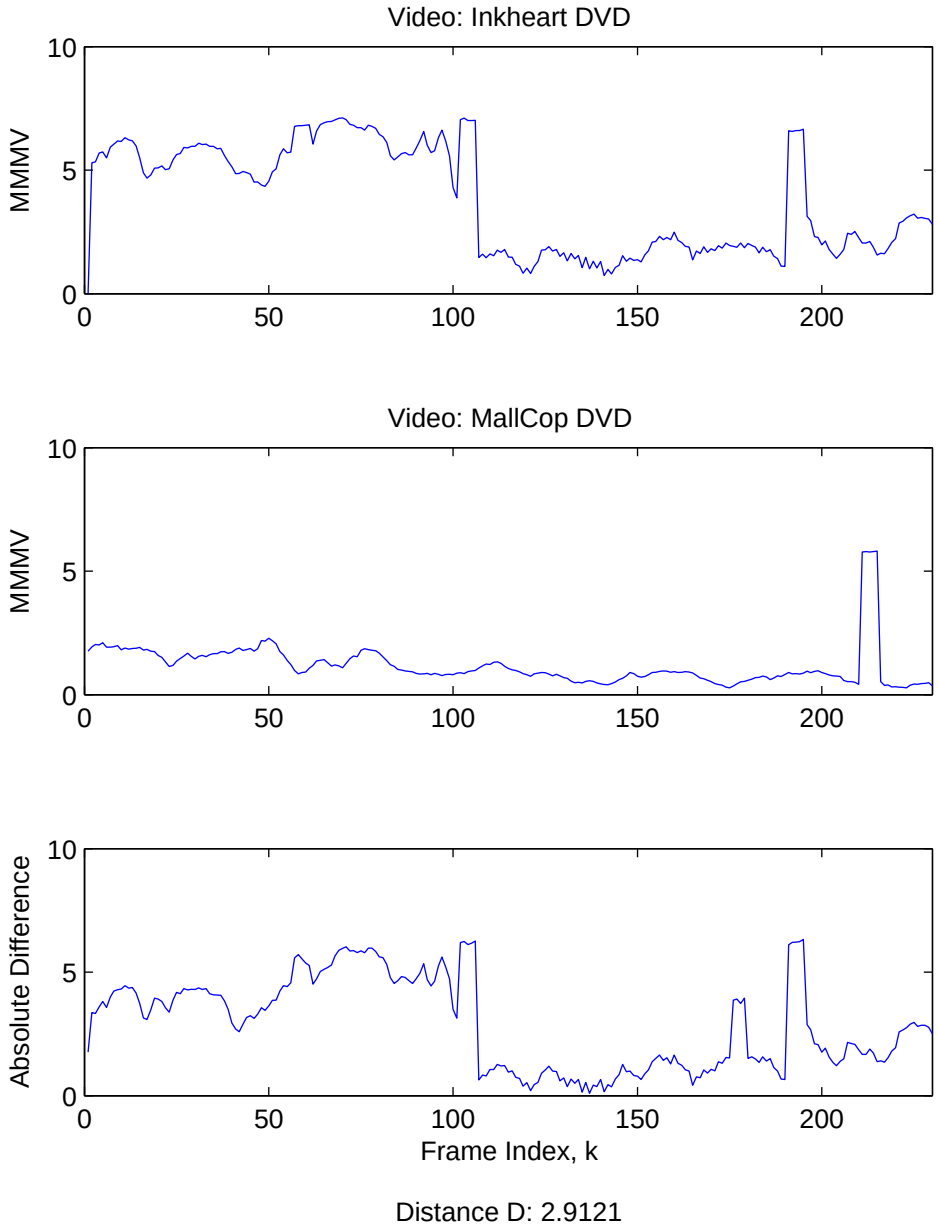


Figure 3.17: MMMV plots of "Inkheart DVD" and "Mallcop CAM" videos. $D(a, b) = 2.91$.

Comparison of distances of 8 test videos are listed in Table 3.3. Rows of Table 3.3 are original videos and columns are hand-held camera recorded versions. The diagonal elements of Table 3.3 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.

Table 3.3: Average of the distance D of $MMMV_N$ of test videos. Diagonal results show the distance of original and its copy.

Movie Name	Desperaux CAM	Inkheart CAM	Mallcop C.	Spirit C.
Desperaux DVD	**0.44**	1.23	0.9	0.86
Inkheart DVD	1.2	**0.08**	0.68	0.74
Mallcop DVD	0.85	0.54	**0.18**	0.75
Spirit DVD	1.06	0.76	0.67	**0.29**

The 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. 3.19. 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. 3.18.

(a)

(b)

Figure 3.18: 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.

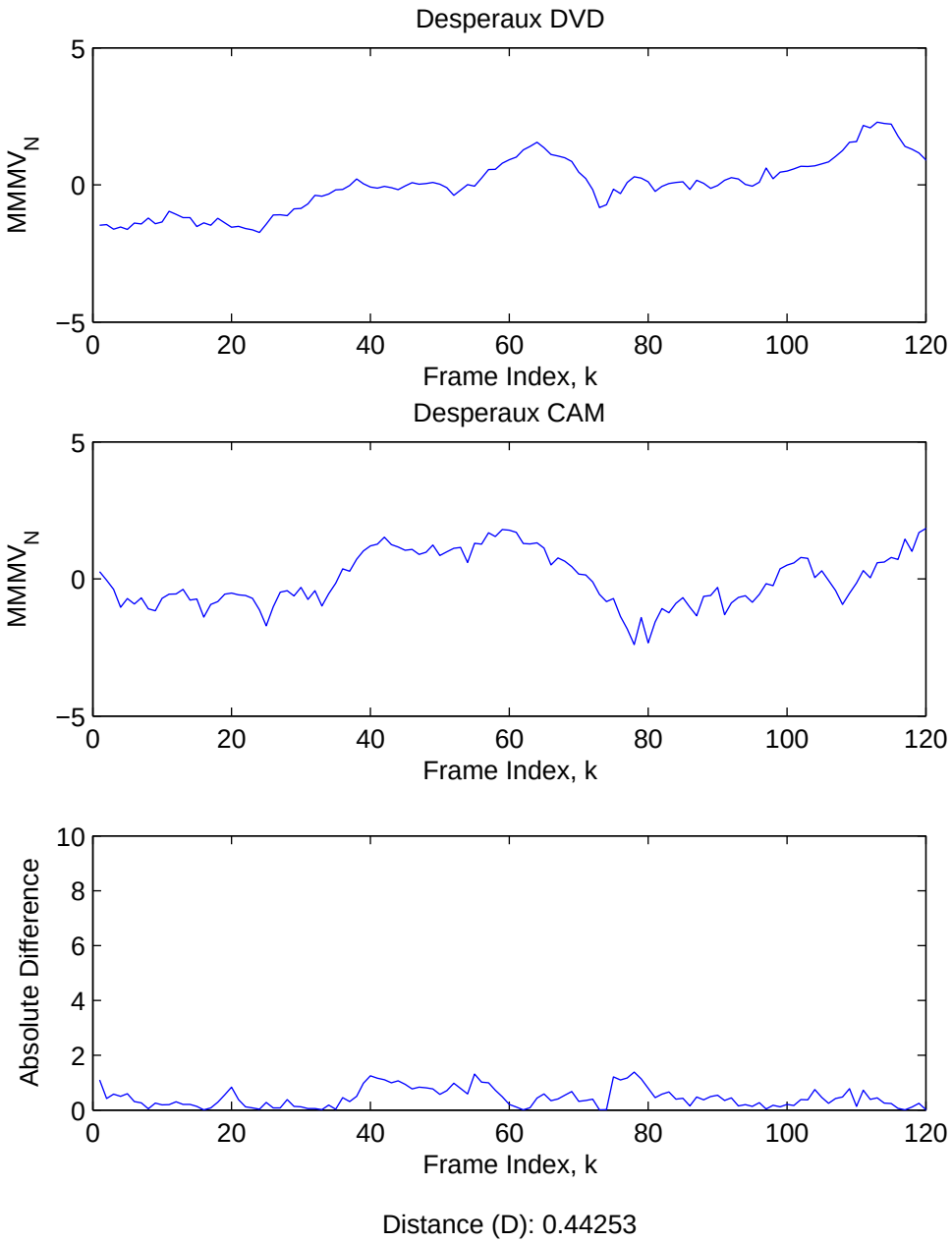


Figure 3.19: $\overline{MMMV}$ plots of "Desparaux DVD" and "Desparaux CAM" video clips. The distance between the MMMV plots, $D(a, b) = 0.44$.

As mentioned in Section 3.2 angle information of motion vectors can be used for comparison. The MPMV plots of "Inkheart DVD" and "Inkheart CAM" are

shown in Fig. 3.20. The original video and the recorded video have very similar MPMV plots. Comparison results of test videos are listed in Table 3.4.

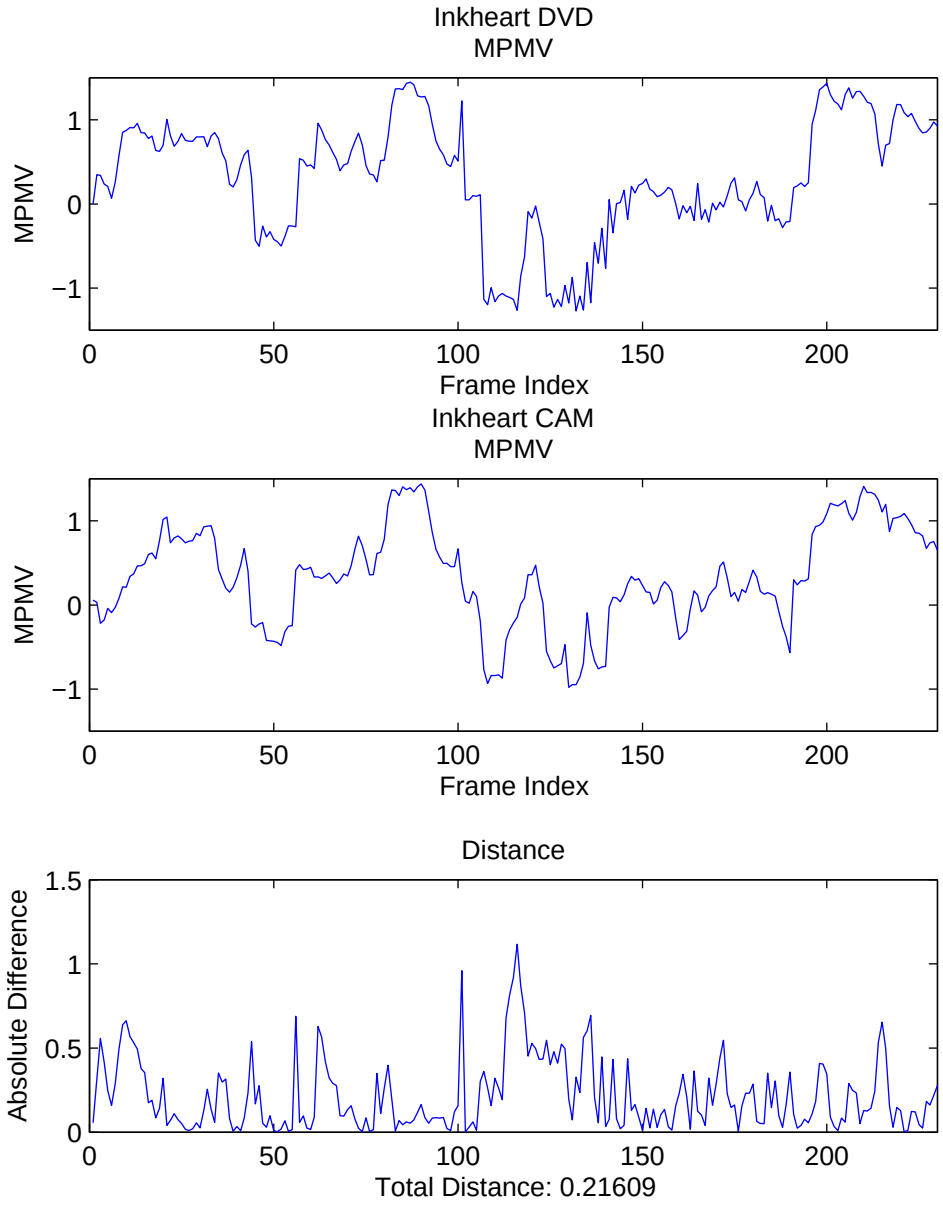


Figure 3.20: The MPMV plots of "Inkheart DVD" and "Inkheart CAM" video clips. The distance between the MPMV plots, $D(a, b) = 0.22$.

Table 3.4: Average distance D of $\overline{MPMV}$ data of test vidoes. Diagonal results show the distance between the original and its copy.

Movie Names	Desperaux CAM	Inkheart CAM	Mallcop C.	Spirit C.
Desperaux DVD	**0.29**	0.96	0.7	0.74
Inkheart DVD	1.03	**0.15**	0.85	0.86
Mallcop DVD	0.98	0.87	**0.4**	0.74
Spirit DVD	0.62	0.75	0.59	**0.24**

Diagonal elements of the Table 3.4 are the smallest elements in a given row in Table 3.4. 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.

3.5 Histogram of Motion Vectors

In Section 3.2 the phase angle or the magnitude of motion vectors are used for comparison. The phase angle and the magnitude of motion vectors contain different information about the videos. When only MMMV of videos are used for comparison MPMV information is neglected and vice versa. However, if both phase and magnitude information are used accuracy of the results are expected to be higher since more information will be used in the comparison step. So, in order to include both information, we propose a feature for videos, histogram of motion vectors (HOMV) described in Eq. 3.6. This section describes the proposed feature and investigates how well HOMV describe a video and uses it in comparison for CBCD.

HOMV contains both the phase angle and the magnitude information in a vector. The HOMV gives information about how strong objects tend to move in a given direction. The elements of the HOMV vector are weighted histogram of phase of motion vectors of macro blocks in an image frame. Each bin of the histogram contains sum of corresponding magnitude of phase values at specific

directions instead of the count of phase values at that direction. Phase angles are discretized during computation and a two-dimensional matrix is computed for a given video as follows:

$$HOMV(m,n) = \frac{L}{N} \sum_{\substack{i \\ \theta_{m-1} \leqslant \theta(n,i) < \theta_m}} r(n,i)\theta(n,i) \tag{3.6}$$

where n is the frame index and m is the histogram bin index, $m \in (0, M)$, L is the number of angle regions and N is the total number of MVs. The phase angle region $(-\pi, \pi)$ is divided into M equal subregions with boundaries θ_m , with $\theta_0 = -\pi$ and $\theta_M = \pi$. $HOMV(m,n)$ is a weighted histogram of $\theta(n,i)$. Weight of the $\theta(n,i)$ is the magnitude of the corresponding motion vector, $r(n,i)$. In this way, more emphasis is given to large motion vectors.

HOMV is a matrix. Rows of the matrix gives temporal information, columns of the matrix gives spatial information. Each column contains histogram of motion vectors in that frame. So, row count is equal to number of bins of HOMV and column count is equal to number of frames. HOMV of the 15th frame of the video "Foreman" is given in Fig. 3.22 as an example. Motion vectors of that frame is shown in Fig. 3.21. This is a spatial feature since it gives information about motion activities in one frame. In order to obtain a temporal feature of the video, it is extended to all frames as shown in Fig. 3.23.

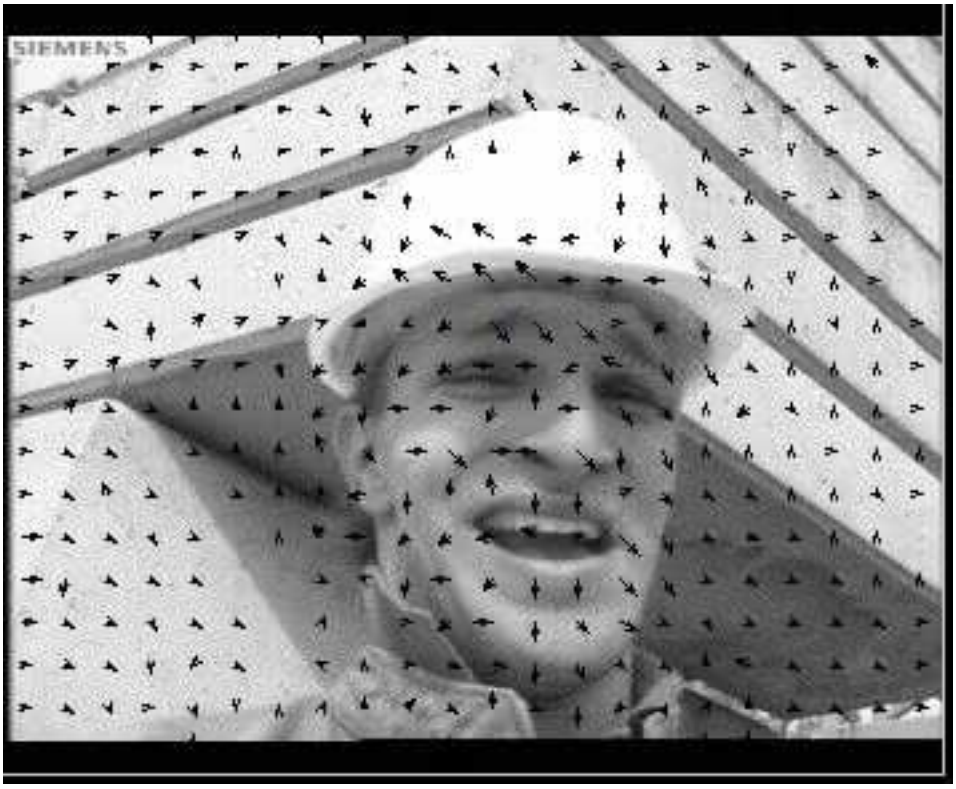


Figure 3.21: 15^{th} frame of video "Foreman" with motion vectors ($n = 5$).

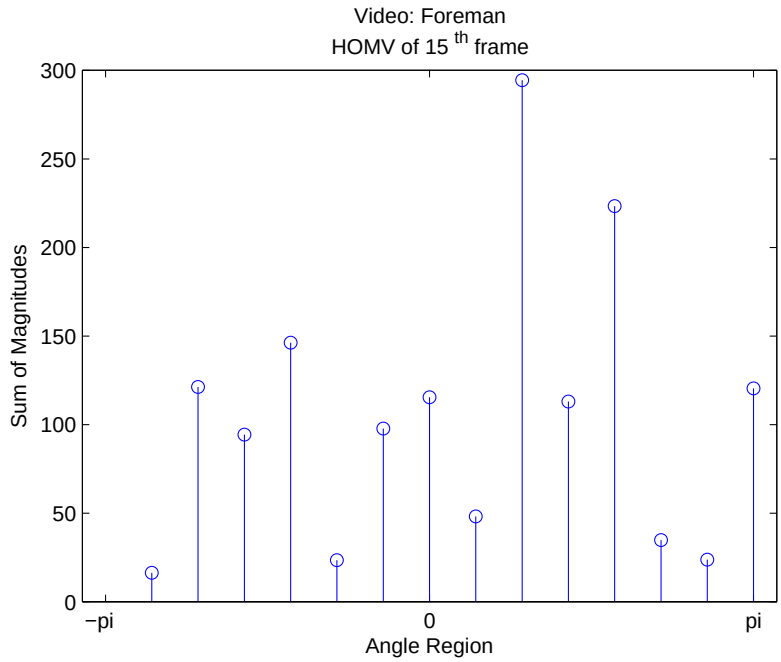


Figure 3.22: HOMV of 15^{th} frame of video "Foreman" which is shown in Fig.3.21

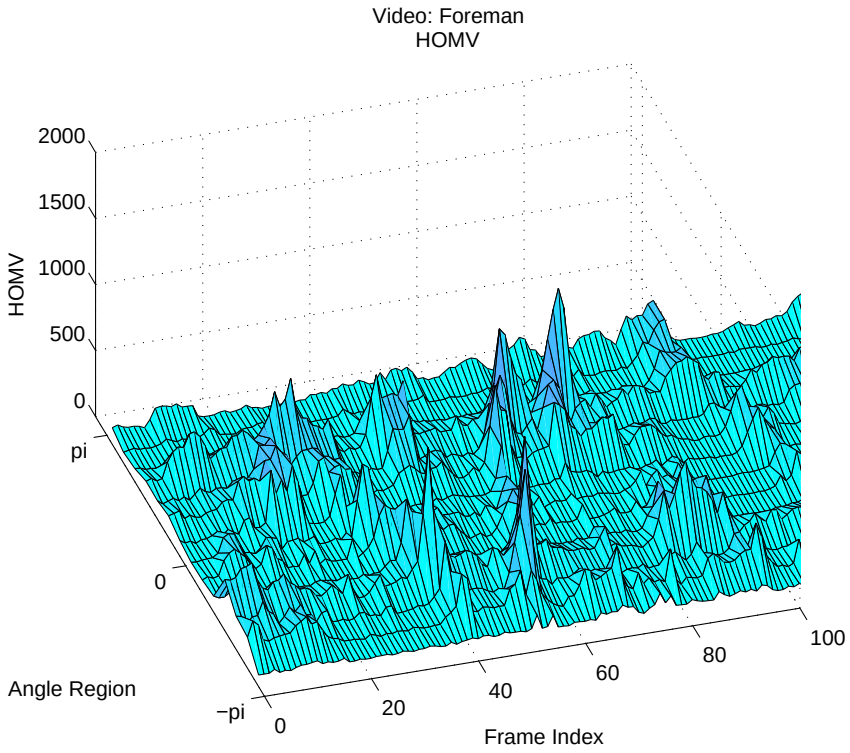


Figure 3.23: HOMV plot of video "Foreman".

HOMV feature of a video can be used in the comparison step as shown in Fig. 3.24. Original video "Inkheart DVD" and camera recording of same video "Inkheart CAM" has similar HOMV plots. Their distance is 86 which is a small value when compared with other distances as shown in Table. 3.5.

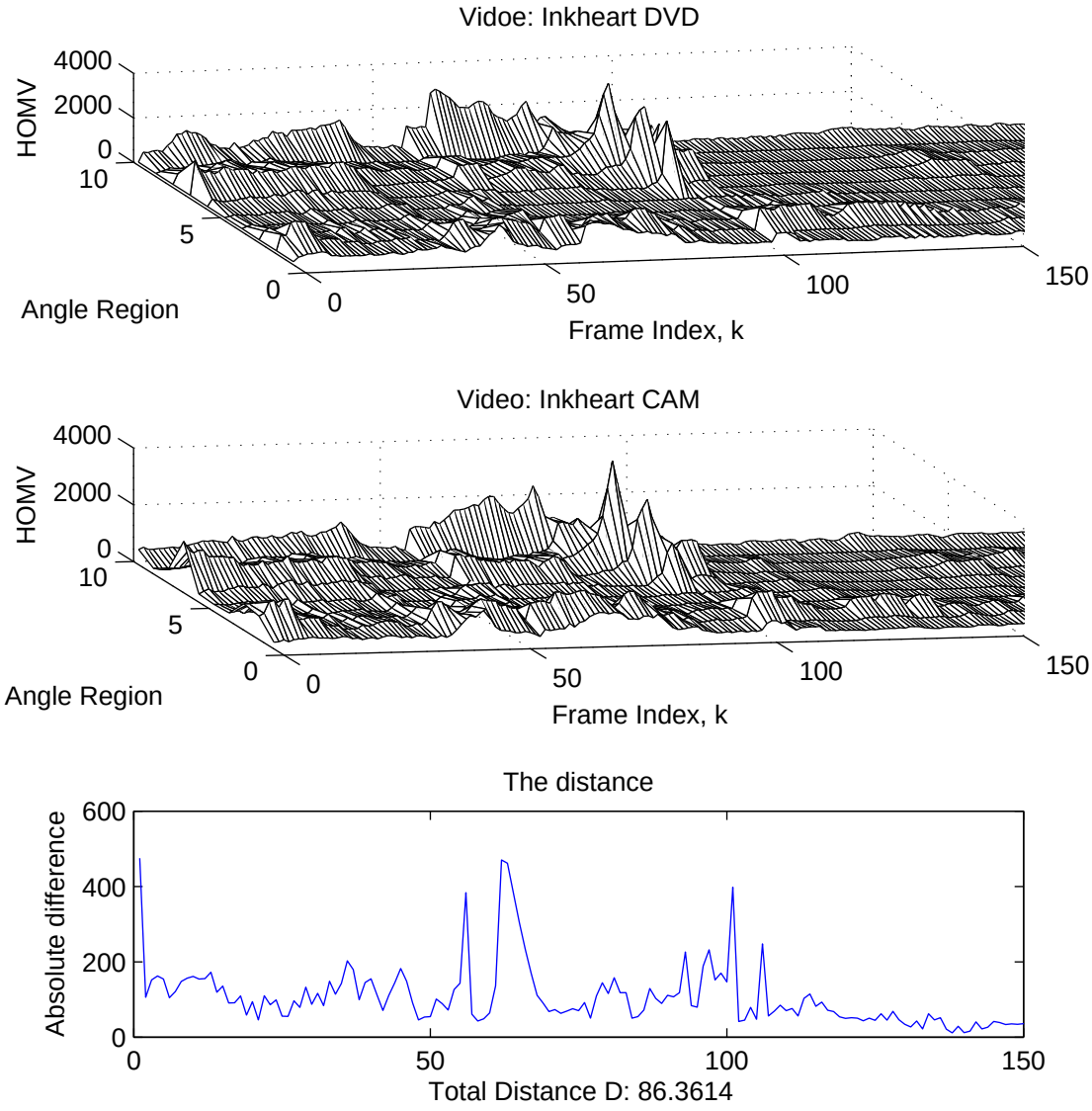


Figure 3.24: HOMV plots of video "Inkheart DVD" and "Inkheart CAM" videos and the distance between the HOMV plots, $D(a, b) = 86.36$

Generally diagonal elements of the Table. 3.5 are the smallest ones in corresponding rows. When it is compared with Table. 3.3 and Table. 3.4, the diagonal elements are more distinguishable than others. However, the first row of Table. 3.5 gives a false detection value. The smallest value, which shows the

Table 3.5: The distance D of HOMV data of test vidoes. Diagonal results shows the distance of original and its copy.

The distance	Desperaux CAM	Inkheart CAM	Mallcop CAM	Spirit CAM
Desperaux DVD	131.86	291.89	116.31	167.1
Inkheart DVD	294.07	86.36	226.94	245.45
Mallcop DVD	232.14	241.9	116.68	249.96
Spirit DVD	152.74	233.51	187.64	101.99

most similar videos, is at 3th element of the row which means that "Desperaux DVD" is more similar to "Mallcop CAM" rather than "Desperaux CAM". The reason is explained previously as some of the information is lost at sides of the video and the copy is a very blurred version of the original video as shown in Fig. 3.18.

HOMV, 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. 3.4, phase information is more resistant to loss of some information and significant deformations in the video. Even magnitude and HOMV data of the videos were not enough to detect the "Desperaux DVD" and "Desperaux CAM" as similar videos, phase data gave correct matching.

3.6 Using Most Active MBs In The Frame

Some MVs do not represent an actual motion, because in a moving object the vectors inside the object may point out arbitrary directions instead of the actual direction of the object. This is due to the fact that in an object pixel values of the neighboring macro blocks are almost the same. Therefore, we assume that the most meaningful information is in fast moving regions. Thus, we developed a method that takes the most active regions into account in a given frame instead of using all motion information as in Sections 3.4 and 3.5. We applied the same

algorithms in Equations 3.2, 3.3 except that we used most active α-percent of MVs where $\alpha \in (0, 100)$. $MMMV_s$ and $MPMV_s$ methods use first α-percent most moving of MVs and they are defined as

$$MMMV_{max}(k) = \frac{1}{\lceil N\frac{\alpha}{100} \rceil} \sum_{i=0}^{\lceil N\frac{\alpha}{100}-1 \rceil} r_s(k,i) \tag{3.7}$$

where $r_s(k,.)$ is the array of first α-percent of highest MV magnitudes of the frame k, N is the number of macro blocks and

$$MPMV_{max}(k) = \frac{1}{\lceil N\frac{\alpha}{100} \rceil} \sum_{i=0}^{\lceil N\frac{\alpha}{100}-1 \rceil} \theta_s(k,i) \tag{3.8}$$

where $\theta_s(k,.)$ is the array of first α-percent of highest MV angles of frame the k.

3.7 Experimental Results

A video database is available in [38]. Original videos in this database are compared with the transformed versions of the same videos. There are 47 original videos taken from [38]. Duration of the videos are 30 seconds. Each video has eight different transforms. The transformations are summarized in Table 3.6. 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 3.6: Video transformations

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, $\mu = 0, \sigma = 0.001$

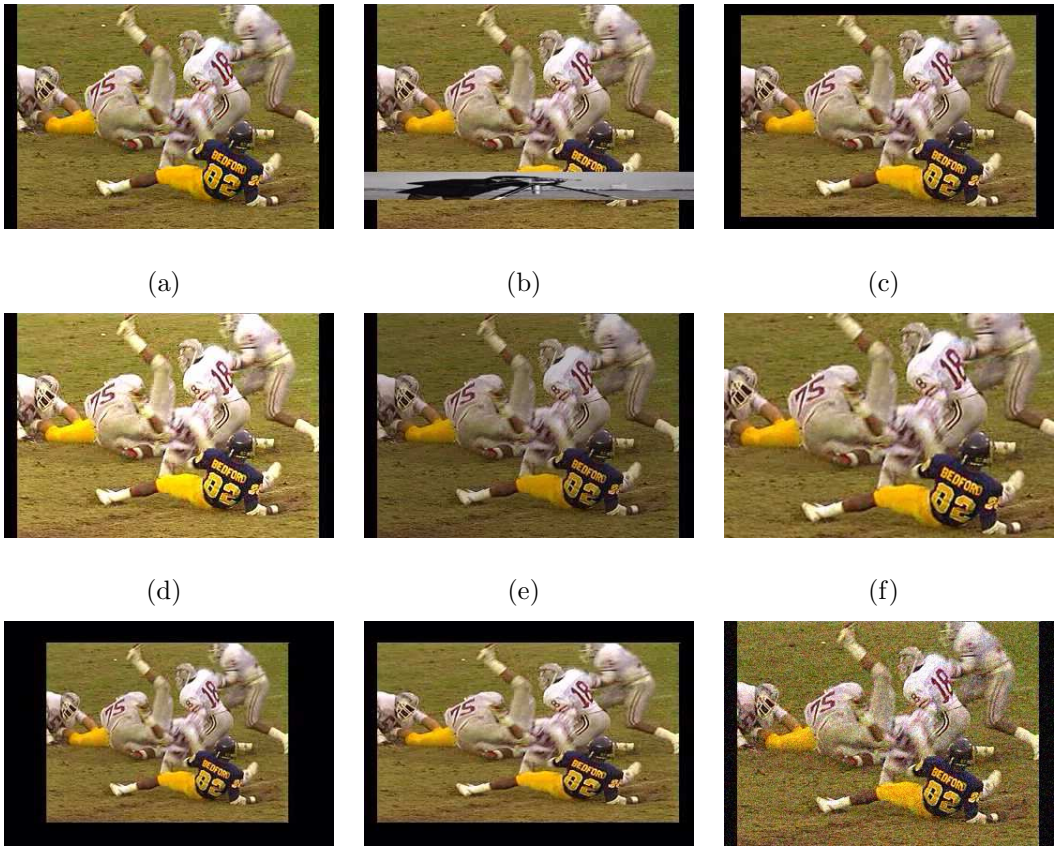


Figure 3.25: Transformations: (a) original frame, (b) a pattern is inserted, (c) crop 10% with a black frame, (d) contrast increased by 25%, (e) contrast decreased by 25%, (f) zoom by 1.2, (g) zoom by 0.8 with in the black window, (h) letter-box, (i) additive Gaussian noise with $\mu = 0 and \sigma = 0.001$.

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. 3.5 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}. Let F_p and F_n the number of false positives (clips that matched with a different video) and false negative (clips that should match, but

did not). False positive and negative rates are defined as

$$F_{pr}(\tau) = \frac{F_p}{N_p}, F_{nr}(\tau) = \frac{F_n}{N_n} \tag{3.9}$$

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%.

Effects of varying the frame skipping parameter n in motion vector extraction step is shown in Fig. 3.26. We can obtain more descriptive features of videos based on motion vectors if we use every 5^{th} frame instead of the current and the next frame in motion estimation step. As it is shown in Fig. 3.26(a) to 3.26(d) there is a dramatic increase in detection ratio with increasing n to 5.

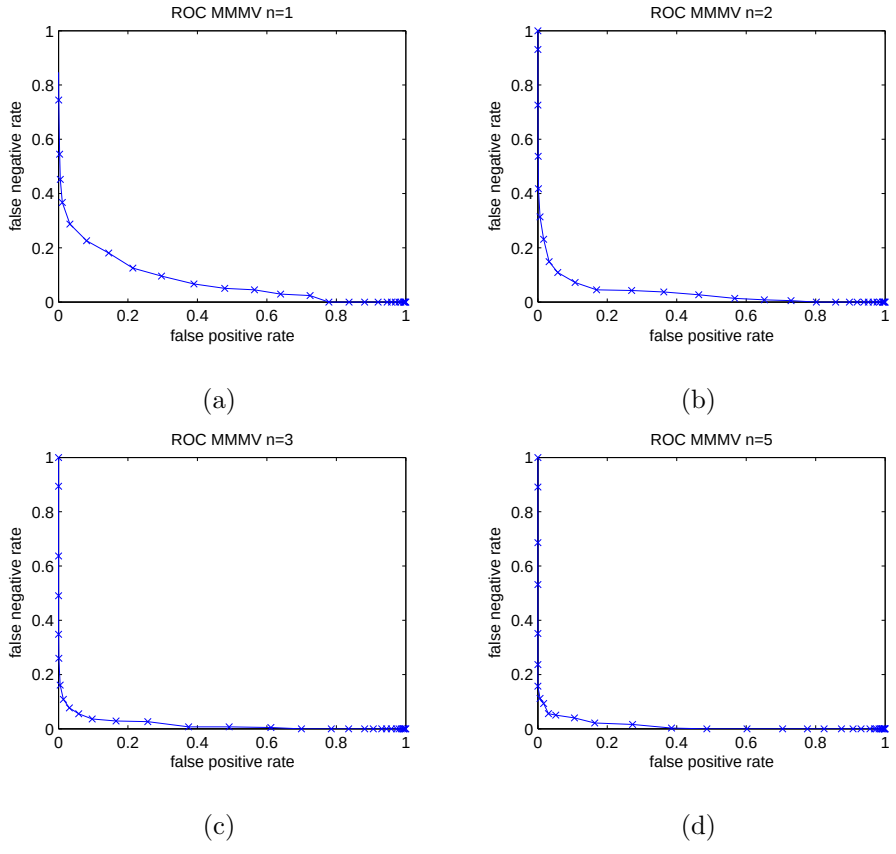


Figure 3.26: Effect of varying n on $MMMV$ plots. (a) $n = 1$ (b) $n = 2$ (c) $n = 3$ (d) $n = 5$.

We test the effects of using upper α% of magnitudes of motion vectors. As it is seen in Fig. 3.27 increasing α increases the detection rate of the tests. Fig. 3.27(d) and Fig. 3.26(d) are very similar to each other. The area under the ROC curve in Fig. 3.26(d) is 0.0115, and the area under the ROC curve in Fig. 3.27(d) is 0.0091. Therefore, the use of upper 50% of the MVs does not significantly effect the accuracy. Instead of using all MVs, upper 50% of the MVs can be used in the MMMV algorithm. In other words, only large motion vectors can be used in practice.

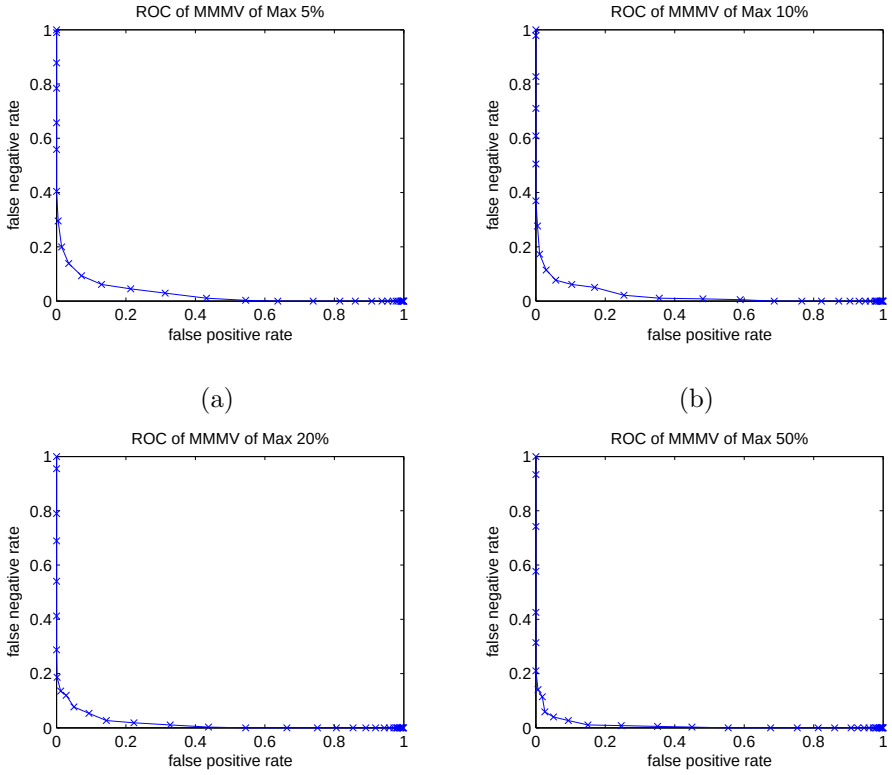


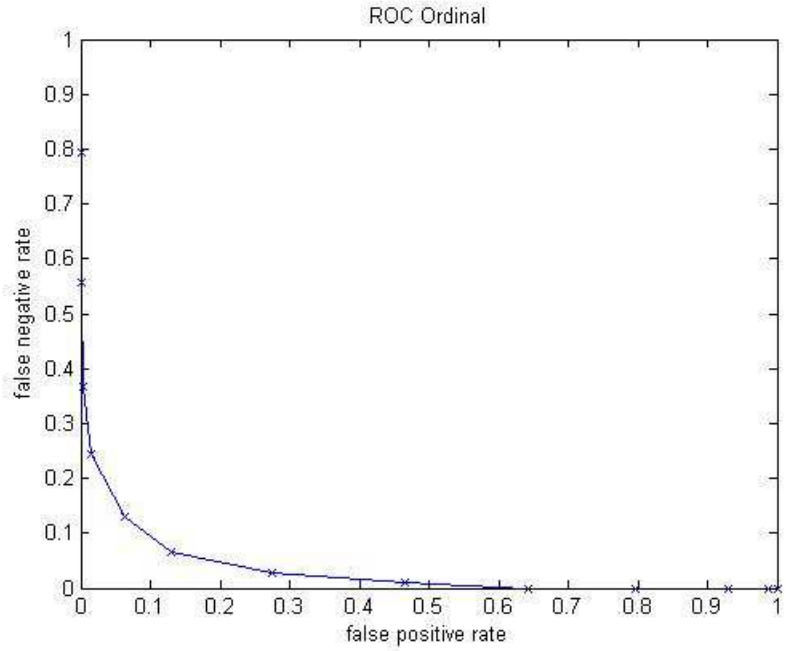
Figure 3.27: Effects of using different α for MMMV. (a) $\alpha = 0.05$ (b) $\alpha = 0.10$ (c) $\alpha = 0.20$ (d) $\alpha = 0.5$, $n = 5$

Table 3.7: The area under the ROC curves of MMMV for different α and n.

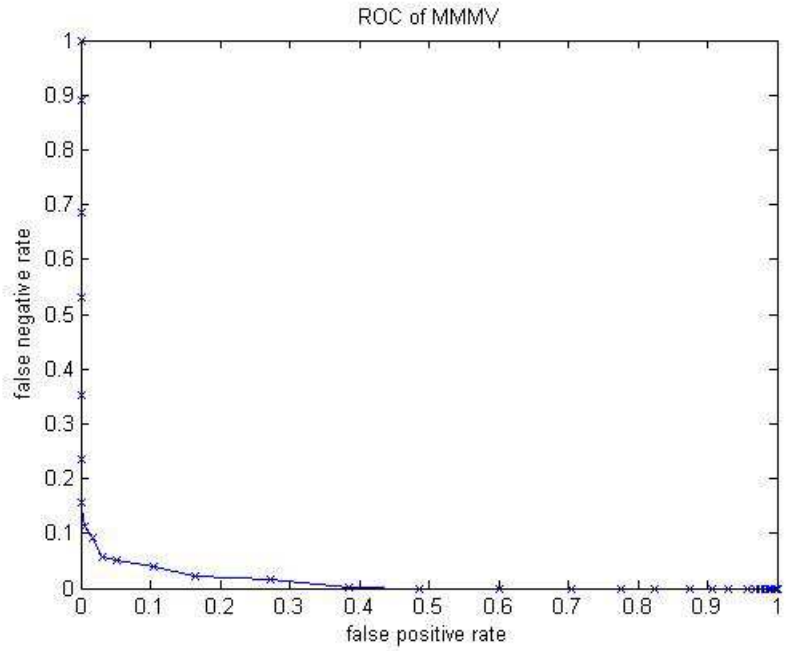
α	15%	25%	50%	100%
n=1	0.0611	**0.0577**	0.0599	0.0807
n=5	0.0205	0.0138	**0.0091**	0.0115

As shown in Table 3.7 using upper 25% for n=1 and 50% for n=5 is closer to the ideal case. Instead of using all MVs, using upper α% of MVs is more advantageous where α varies according to n.

In [7] it is stated that *Ordinal Signature* outperforms the *Motion Signature*. This is true when the motion vectors are extracted using the current and the next frame. On the other hand, if motion vectors of the 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 3.28.



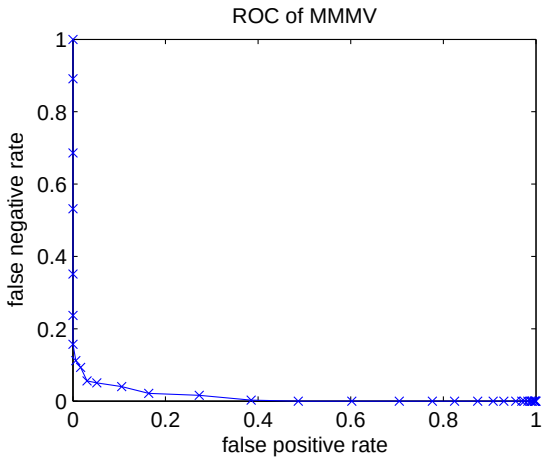
(a)



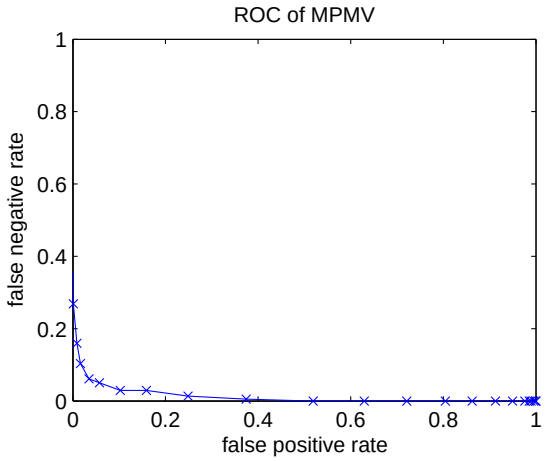
(b)

Figure 3.28: The ROC curves of Ordinal signature and MMMV signatures. MMMV is a better signature than the ordinal signature when n=5. (a) ROC curve of results of ordinal measurement, (b) ROC curve of MMMV, n=5.

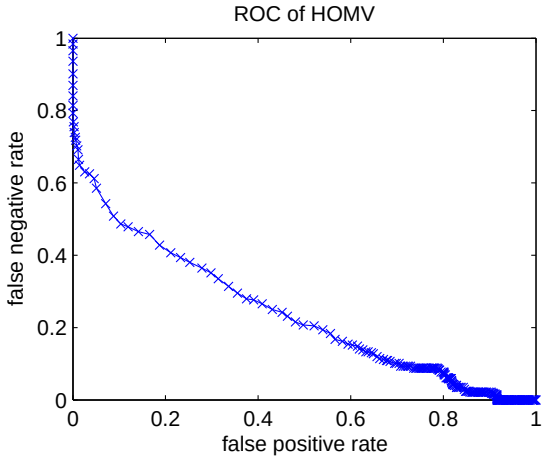
In this thesis, we proposed MMMV, MPMV and HOMV signature as motion vector based signatures of videos. Comparison of ROC curves of these methods are given in Fig. 3.29. ROC curves of the MMMV and MPMV are very close to each other. On the other hand the HOMV has a poor performance. It is experimentally shown that the MMMV and the MPMV are good descriptive features for videos. In this database the best results are obtained with alpha=50% and n=5.



(a)



(b)



(c)

Figure 3.29: Comparison of ROC curves of proposed methods, n=5. (a) MMMV, (b) MPMV and (c) HOMV

3.7.1 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 3.8 summarizes the feature per frame (FPF) values of several algorithms. The FPF values of algorithms except MMMV, MPMV and HOMV are taken from [5].

Table 3.8: Sizes of feature spaces

Technique	Features Per Frame
ViCopT [24]	7
AJ [22]	4.8
STIP [39]	73
Temporal [5]	0.09
Ordinal Meas. [7]	9
MMMV	1
MPMV	1
HOMV	14[a]

[a]It is equal to the number of bins used in the histogram. If 4 bin histogram is used this value will be 4.

Table 3.8 shows that MMMV and MPMV algorithms consume less space for signatures than the other algorithms except the method called "Temporal" [5].

Chapter 4

CONCLUSIONS

In this thesis, it is experimentally shown that motion vectors are unique signatures of videos. Motion vectors can be used in similar video detection or 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.

In this thesis, motion-vector based feature parameters for videos are defined. The proposed feature parameters depend on motion vectors of macro blocks of video frames.

Magnitude and phase 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. However the MPMV feature is robust because, usually, 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.

Using the upper 50% of MVs gives very similar results to the MMMV using all MVs in terms of accuracy. So, instead of using all MVs, the maximum 50% of the MVs can be also used. This reduces the computational cost a little bit.

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.

Bibliography

[1] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data. a state-of-the-art overview," *Signal Processing Magazine, IEEE*, vol. 17, pp. 20–46, Sep 2000.

[2] M. Swanson, B. Zhu, B. Chau, and A. H. Tewfik, "Object-based transparent video watermarking," in *Proc. IEEE Workshop on Multimedia Signal Processing*, pp. 369–374, 1997.

[3] G. Doërr and J.-L. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Communication*, vol. 18, no. 4, pp. 263 – 282, 2003.

[4] M. Swanson, B. Zhu, and A. Tewfik, "Data hiding for video-in-video," in *Image Processing, 1997. Proceedings., International Conference on*, vol. 2, pp. 676–679 vol.2, Oct 1997.

[5] J. Law-To, L. Chen, A. Joly, I. Laptev, O. Buisson, V. Gouet-Brunet, N. Boujemaa, and F. Stentiford, "Video copy detection: a comparative study," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, (New York, NY, USA), pp. 371–378, ACM, 2007.

[6] A. Hampapur and R. Bolle, "Comparison of distance measures for video copy detection," in *Multimedia and Expo, 2001. ICME 2001. IEEE International Conference on*, pp. 737–740, Aug. 2001.

[7] A. Hampapur, K. Hyun, and R. M. Bolle, "Comparison of sequence matching techniques for video copy detection," in *Storage and Retrieval for Media Databases 2002* (M. M. Yeung, C.-S. Li, and R. W. Lienhart, eds.), vol. 4676, pp. 194–201, SPIE, 2001.

[8] L. Teodosio and W. Bender, "Salient video stills: content and context preserved," in *MULTIMEDIA '93: Proceedings of the first ACM international conference on Multimedia*, (New York, NY, USA), pp. 39–46, ACM, 1993.

[9] Y. Tonomura and S. Abe, "Content oriented visual interface using video icons for visual database systems," in *Visual Languages, 1989., IEEE Workshop on*, pp. 68–73, Oct 1989.

[10] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Syst.*, vol. 1, no. 1, pp. 10–28, 1993.

[11] E. Ardizzone, M. L. Cascia, V. D. Gesú, and C. Valenti, "Content based indexing of image and video databases by global and shape features," in *In Proc. Int. Conf. Pattern Recognition*, 1996.

[12] M. M.-Y. Yeung, *Analysis, modeling and representation of digital video*. PhD thesis, Princeton, NJ, USA, 1996.

[13] V. Kobla, D. Doermann, K.-I. D. Lin, and C. Faloutsos, "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.

[14] R. Mohan, "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, May 1998.

[15] C. Kim and B. Vasudev, "Spatiotemporal sequence matching for efficient video copy detection," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 15, pp. 127–132, Jan. 2005.

[16] S. Satoh, "News video analysis based on identical shot detection," in *Multimedia and Expo, 2002. ICME '02. Proceedings. 2002 IEEE International Conference on*, vol. 1, pp. 69–72 vol.1, 2002.

[17] M.C. Yeh and K.T. Cheng, "Video copy detection by fast sequence matching.," in *Proceedings of the ACM International Conference on Image and Video Retrieval (ACM CIVR)*, April 2009.

[18] E. Ardizzone, M. L. Cascia, A. Avanzato, and A. Bruna, "Video indexing using mpeg motion compensation vectors," in *ICMCS '99: Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, (Washington, DC, USA), p. 725, IEEE Computer Society, 1999.

[19] A. Joly, C. Frélicot, and O. Buisson, "Robust content-based video copy identification in a large reference database," in *Proceedings of ACM International Conference on Image and Video Retrieval (CIVR)*, vol. 2728, pp. 511–516, 2003.

[20] A. Joly, O. Buisson, and C. Frelicot, "Statistical similarity search applied to content-based video copy detection," in *ICDEW '05: Proceedings of the 21st International Conference on Data Engineering Workshops*, (Washington, DC, USA), p. 1285, IEEE Computer Society, 2005.

[21] A. Joly, C. Frelicot, and O. Buisson, "Content-based video copy detection in large databases: a local fingerprints statistical similarity search approach," in *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, vol. 1, pp. I–505–8, Sept. 2005.

[22] A. Joly, O. Buisson, and C. Frelicot, "Content-based copy retrieval using distortion-based probabilistic similarity search," *Multimedia, IEEE Transactions on*, vol. 9, pp. 293–306, Feb. 2007.

[23] W.-L. Zhao, C.-W. Ngo, H.-K. Tan, and X. Wu, "Near-duplicate keyframe identification with interest point matching and pattern learning," *Multimedia, IEEE Transactions on*, vol. 9, pp. 1037–1048, Aug. 2007.

[24] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa, "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*, (New York, NY, USA), pp. 835–844, ACM, 2006.

[25] J. Law-To, V. Gouet-Branet, O. Buisson, and N. Boujemaa, "Local behaviours labelling for content based video copy detection," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, vol. 3, pp. 232–235, 0-0 2006.

[26] S. Poullot, O. Buisson, and M. Crucianu, "Z-grid-based probabilistic retrieval for scaling up content-based copy detection," in *CIVR '07: Proceedings of the 6th ACM international conference on Image and video retrieval*, (New York, NY, USA), pp. 348–355, ACM, 2007.

[27] R. Lienhart, C. Kuhmünch, W. Effelsberg, U. Mannheim, P. I. Iv, D-Mannheim, R. Lienhart, C. Kuhmnch, and W. Effelsberg, "On the detection and recognition of television commercials," pp. 509–516, 1996.

[28] J. M. Sanchez, X. Binefa, J. Vitria, and P. Radeva, "Local color analysis for scene break detection applied to TV commercials recognition," in *Visual Information and Information Systems*, vol. 1614 of *Lecture Notes in Computer Science*, Springer Berlin / Heidelberg, 1999.

[29] A. Hampapur and R. Bolle, "Feature based indexing for media tracking," in *Multimedia and Expo, 2000. ICME 2000. 2000 IEEE International Conference on*, vol. 3, pp. 1709–1712, 2000.

[30] G. Indyk and N. Shivakumar, "Finding pirated video sequences on the internet," tech. rep., Stanford Infolab Technical Report, Feb 1999.

[31] M. R. Naphade, M. M. Yeung, and B.-L. Yeo, "Novel scheme for fast and efficent video sequence matching using compact signatures," vol. 3972, pp. 564–572, SPIE, 1999.

[32] O. Küçüktunç, "Content-based video copy detection using multimodal analysis," Master's thesis, Bilkent University, Department of Electrical and Electronics Engineering, Ankara, Turkey, 2009.

[33] Y.-F. Ma and H.-J. Zhang, "A new perceived motion based shot content representation," 2001.

[34] T. Liu, H. Zhang, and F. Qi, "A novel video key frame extraction algorithm," in *Circuits and Systems, 2002. ISCAS 2002. IEEE International Symposium on*, vol. 4, pp. IV–149–IV–152 vol.4, 2002.

[35] J. Watkinson, *MPEG Handbook*. Newton, MA, USA: Butterworth-Heinemann, 2001.

[36] J. Lu and M. Liou, "A simple and efficient search algorithm for block-matching motion estimation," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 7, pp. 429–433, Apr 1997.

[37] E. Chan and S. Panchanathan, "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, Sep 1993.

[38] Internet Archive, "Internet archive movie database," 2009. [Online; accessed 10-Agust-2009].

[39] I. Laptev and T. Lindeberg, "Space-time interest points," in *IN ICCV*, pp. 432–439, 2003.