Introduction to Video Fingerprinting

Wei-Lun Chao

Graduate Institute of Communication Engineering, NTU

Abstract

This report presents a brief overview of video fingerprinting, a popular and powerful technology for copy detection and copyright management. I'll start with the basic concept of video fingerprinting, give a compact introduction to its procedure of execution and then take a look at different kinds of proposed methods of video fingerprinting also their advantages. At the end, I'll have a summary about my project work and result in UIUC with professor, Pierre Moulin and two undergraduate students, Julien Dubois and Ryan Rogowski.

1. Introduction

With the fast development of technology and increasing use of the widespread availability of ADSL and the World Wide Web, people can easily find and upload tons of videos on the Internet. There exist too many duplicated and transformed video clips online and some of them may be illegally copied or broadcasted, so database and copyright management have become big issues nowadays. There are two main approaches for solving these issues, one is the well-known "Watermarking", and one is the topic of this report, "video fingerprinting". Figs 1.1, 1.2 and 1.3 show these techniques.

Watermarking relies on inserting a distinct pattern into the video stream, while copy-detection techniques match content-based signatures to detect copies of video. The primary thesis of content-based copy detection (CBCD) is ***"the media itself is the watermark"***, the media contains enough unique information that can be used for detecting copies. Content-based copy detection schemes extract a small number of pertinent features from the original media, called ***"fingerprints"*** or ***"signatures"*** of the video. The same signatures are extracted from the test media stream and compared to the original media signature according to a dedicated ***voting algorithm*** to determine if the test stream contains a copy of the original media.

The bottleneck of watermarking is that the inserted marks are likely to be destroyed or distorted as the format of the video get transformed or during the transmission, while the video signature extraction of the content-based copy detection can be done after the media has been distributed. That's why video fingerprinting attracts more and more attention recently and in this report we'll take a look at how it has been used to detect copies.

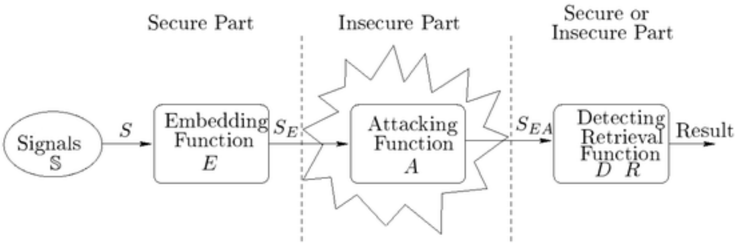


Figure 1.1: General watermark life-cycle phases with embedding-, attacking- and detection/retrieval functions

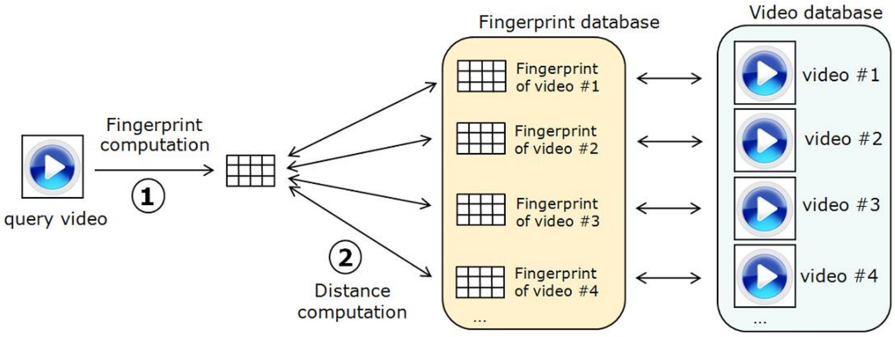


Figure 1.2: Video retrieval in a database in two steps

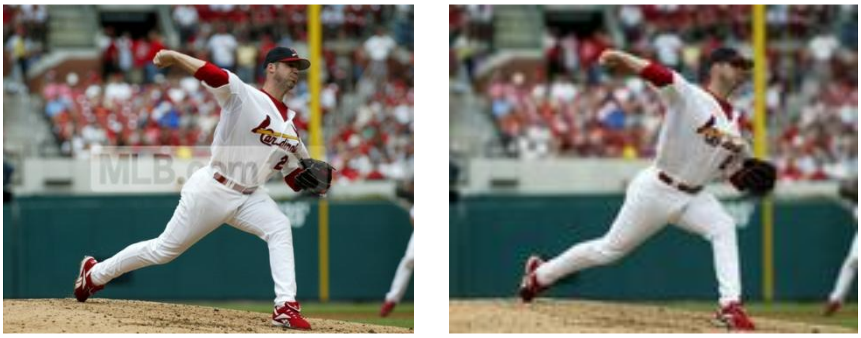


Figure 1.3: Examples of watermarking. On the Major League baseball website, you have to pay for high-quality photos, or you can only get high-quality photos with "watermarks" inside or get blurring photos.

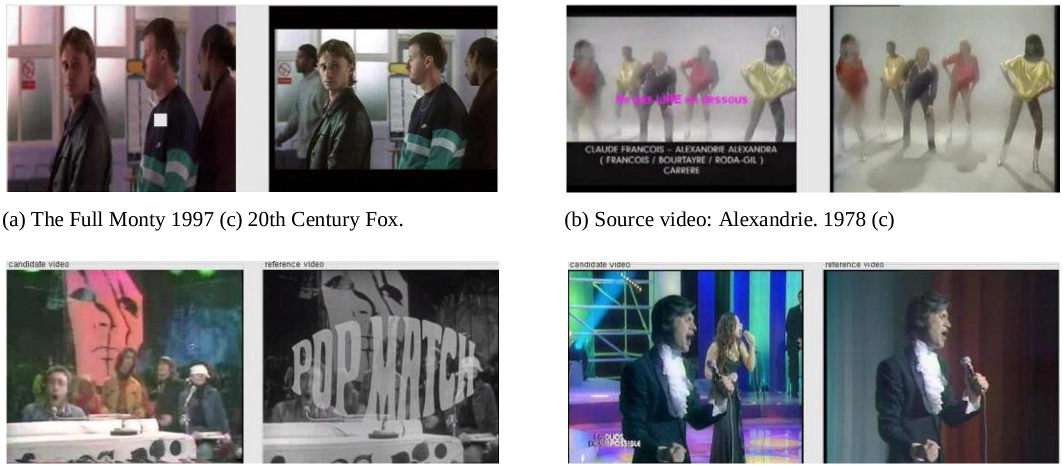
Section 2 discusses challenges in video copy detection, while section 3 discusses concepts and previous works of video fingerprinting. Section 4 shows the procedure and stages of the whole CBCD system, also details of interesting and powerful methods are included. My project is briefly introduced in section 5, and section 6 gives a conclusion and discusses future work. Besides, two reports, ***"Comparison of Video Copy Detection techniques"*** and ***"The Core of Video Fingerprinting"*** are written as supplement of this report.

2. Challenges in Video Detection

People can easily determine whether the test video is a copy from the video in the database by eyes, while there are some difficulties for computers. A video clip can be encoded in different formats depending on the purpose (e.g. AWG uses less memory storage than DVD but has worse quality). Different formats can give rise to several distortions, such as change in brightness, shift in hue, change in saturation and spatial shift in the picture. Besides these digital artifacts, lossy encoding processes introduce artifacts like the blocking effects in MPEG. There are kinds of signature extraction methods depend on the color and image information in the videos, such as histograms and color coherence vectors, and due to the artifacts above a wrong detection probably occurs [1].

Besides distortions from different formats, there still other kinds of factors make the copy detection difficult, such as frame drops, noise during transmission and storage, blurring are common distortions [2,4,5]. Hardly detected factors come from the building of a copy video, for example, just cutting a small part of a movie, zooming or changing the contrast, inserting words or logos, and changing the background of the original video or even combining several video clips into a new video. Some cases are shown in fig 2.1.

At the beginning, CBCD is used only to detect whole movies, but now, more and more complicated video fingerprinting methods are created to solve newborn strong cases [4].



(a) The Full Monty 1997 (c) 20th Century Fox.

(b) Source video: Alexandrie. 1978 (c)

(c) Source video: Samedi et Compagnie 1970 (c)ORTF.

(d) Source video: Gala du Midem. G. Ulmer 1970 (c) INA

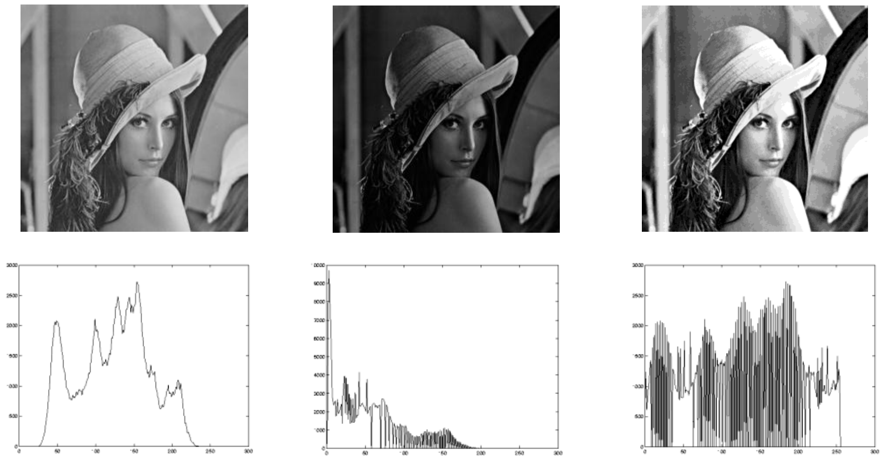
Figure 2.1: Examples of some strong distortion cases. (a) shows the zooming case, while (b) shows both zooming and logos inside. (c) changes the color video into only gray-level and also logos inside, while (d) changes the background.

3. Previous Works

The most crucial part in CBCD is the feature extraction. Features for fingerprinting should be carefully chosen since they directly affect the performance of the entire video fingerprinting system. What information in a video can be used to build a well-identified and robust fingerprint? In the proposed paper of S. Lee and C.D. Yoo [3], three important qualities have been proposed.

- **Robustness:** The fingerprints extracted from a degraded video should be similar to the fingerprinting of the original video.
- **Pair-wise independent:** Two videos, that are perceptually different, must have different fingerprints.
- **Database search efficiency:** Fingerprints must suitable for fast database search.

Many features have been proposed for the video fingerprinting [1,3,4], e.g. color (luminance) histogram, mean luminance and its variant, dominant color, etc. But in section 2, we have briefly state that color or gray-level image is not robust enough against color or luminance variation. Fig 3.1 gives an example how gamma correction and histogram equalization affect the gray-level histogram.



(a)original Lena image (b)gamma correction (gamma=2.5) (c)histogram equalization

Figure3.1: Histogram comparison of 256-gray-level Lena image. Histograms here are 256-bin, although the luminance of these three images is different, we can still tell that they have the same content inside. But from the three histograms, it’s rarely hard to identify that they come from images with same content.

From this example, even the same-content images can have totally different features, and with the histogram equalization operation on different images, a wrong detection probably occurs.

Feature extraction techniques based on more information of video content have been proposed. For example, in the paper published by A. Hampapur and R. Bolle [1], CBCD with motion direction, ordinal intensity, color histogram are introduced and compared. Let’s go back to the intrinsic quality of a video, what’s the difference between it and an image? A video can be seen as an image stream, usually containing more than 24 images (called frames) in one second. Then a native and important feature for videos is the information among frames. Features can be classified into three dimensions.

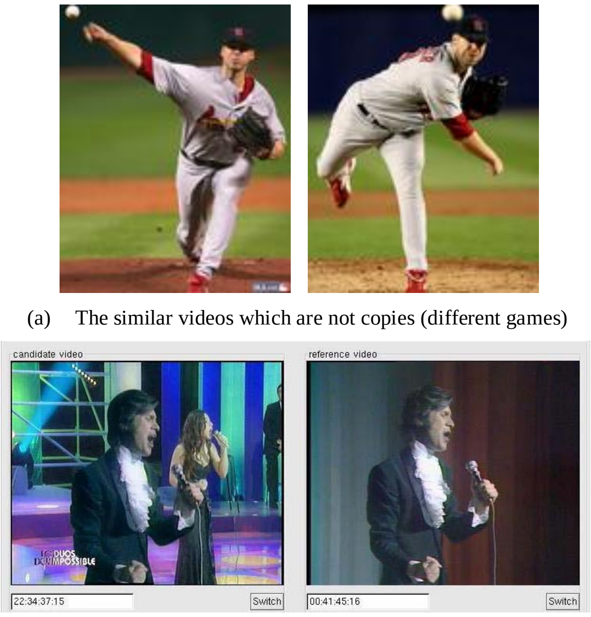
- **Color dimension:** This dimension depends on the color or gray-level properties of frames, such as histogram, hue, saturation, etc.
- **Spatial dimension:** This dimension extracts the distribution of color or arrangement of objects inside frames, such as ordinal signature [1] in space domain and Centroids of Gradient Orientations [3], etc. The key idea of this group is that it treats each pixel of a frame different according to the pixel location.
- **Temporal dimension:** Changes among frames or the order of frames are the key concepts of this dimension, motion detection [1] and ViCopT [2,4,6] are popular examples exploiting the temporal information.

A brief introduction about these three dimensions is included in proposed [1], and the experiment results of this paper shows that more dimensions exploited in the signature extraction technique, more robust it is. Ordinal intensity signature [1] separates each frame into 9 blocks, compares their average intensities then assigns a number for each block. This technique contains both color and spatial dimensions and its matching function also has the properties of the temporal dimension, surely it dominates over color histogram. And why ordinal intensity performs better than motion detection is because it uses the relative distribution rather than the exact information, which makes the signature immune to global changes in the quality of the video that are introduced by the digitization/encoding process.

But what will happen if treating background changing or logo insertion with these three techniques? The result is fairly unreliable. The main reason is that these three approaches use “all information” in frames, which is called “global feature”. To be detailed, a zooming 0.8 videos will leaves the edge with black columns and severely results in changes of histograms and even the averagely intensities. Here comes another classification of features introduced by J. Law-To and V. Gouet-Brunet [2,4,6] as below.

- **Global descriptor:** A descriptor just means the method to extract signatures. The key word “global” here shows that signatures of this class coming from the whole image.
- **Local descriptor:** The signature exploits just parts of the whole image, such as the Harris interest point detector [8] and the key point detection of SIFT [5].

The result of using the local descriptors on changing of background or logo insertion is much better than using global descriptors.[2,4,6]. The dominant advantage of local descriptors is that they only focus on some points of interest rather than the whole image, so even if the background which occupies a big space has been changed, as long as the signatures only exploit the information of key points, we can still detect the copies.



(a) The similar videos which are not copies (different games)

(b) Two videos which are copies (one is used to make the other)

Source video: *Gala du Midem.* G. Ulmer 1970 (c) INA

Figure 3.2: Similarity/ Copy

This key property of local descriptor can also be used to solve another big issue, Content-Based Video Retrieval (CBVR). A crucial difficulty comes from the fundamental difference between a copy and the notion of similar images encountered in CBVR: a copy is not an identical or a near replicated video sequence but rather a transformed video sequence. These photometric or geometric transformations (gamma and contrast transformations, overlay, shift, etc) can greatly modify the signal. From fig 3.2, we can see why recognition between CBCD and CBVR is so important. Using the global feature may detect fig3.2 (a) as the same video but fig 3.2(b) as different videos, but with the key points setting on the moving objects (in fig 3.2 are pitchers and singers), it’s more likely to make a correct detection.

Widely-used local feature techniques are introduced in some reference papers [2,4,6,7,8,9,10]. Different combinations of key-point extraction methods and local description methods yield different results, I’ll give more detail about the technique in [7] by A. Joly st. al. and the ViCopT [2,4,5] later in section 4.

4. Procedure, Steps & Symbolic Techniques

We have gone through several concepts and cases of video fingerprinting techniques, now let me show the procedure, steps of the whole CBCD system, symbolic and powerful techniques of each step are included.

Fig 4.1 gives an outline of the whole system. Here I separate the whole works into three steps: ***Database Operation***, ***Matching***, and ***Testing Query Operation***. And there are four kinds of blocks inside these three steps: ***Data***，***Operation***，***Strategy block***，and ***Final Decision***. Following are the introductions. The reason why I introduce the whole system rather than just the “feature extraction” part is because not only this step but other steps can affect the robustness of a technique.

4.1 Three steps

4.1.1 Database Operation:

Prepared work for copy detection, or it can be called the out-line work. Time consuming here is beyond consideration because we can execute the work at any time even when copy detection is not in use.

4.1.2 Testing Query Operation:

This is the starting point of copy detection. In this part, a video or just a small clip is being tested to find if it’s a copy from videos of the database or not. Time consuming here is considered.

4.1.3 Matching:

This is the decision part of the system. A powerful matching function/voting algorithm should be created for signature matching. Time consuming here is also considered.

4.2 Four kinds of blocks

4.2.1 Data:

There are three different data blocks in fig 4.1. “***Videos in database***” means the original, genuine videos collection. The database may be stored in the institution of copyright management, such as the one used in [2], INA (the French Institut National de l Audiovisuel). ***“Test Clip”*** presents a suspicious copy we want to check. The ***“Signatures Database”*** contains signatures extracted from videos in database.

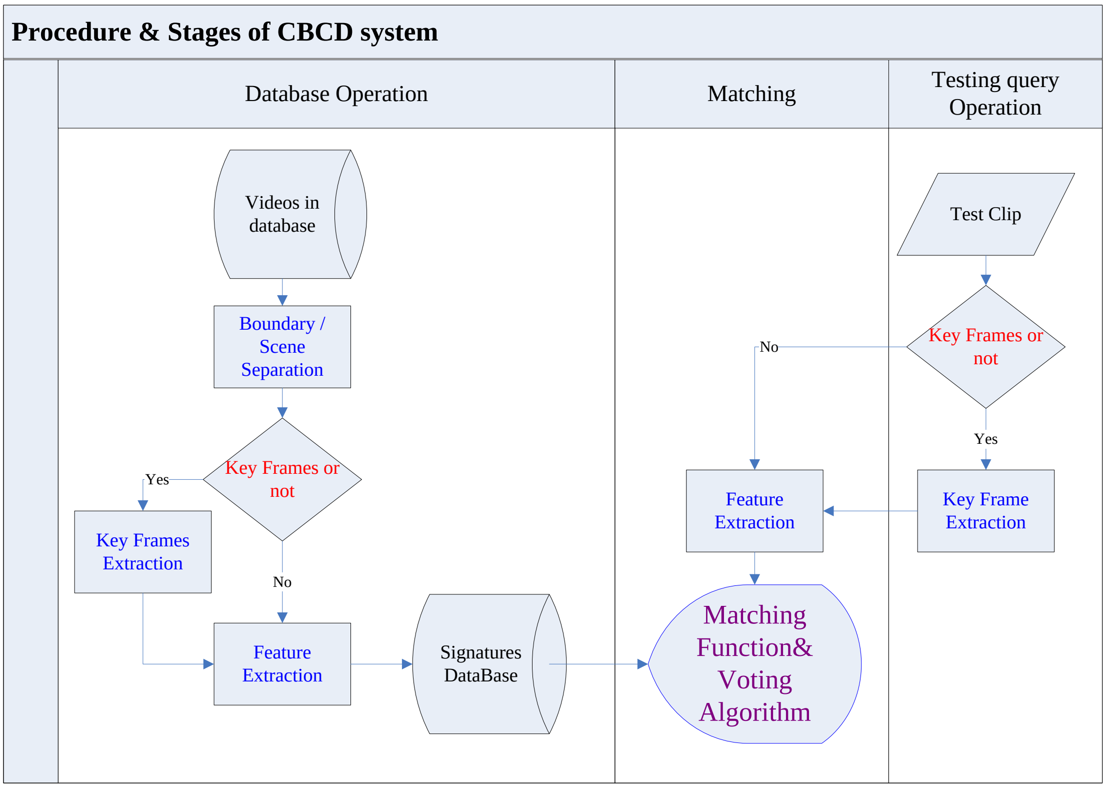


Figure 4.1: The procedure and stages of the CBCD system. **The blocks of cylinder-shaped or parallelogram-shaped are data, while rectangular blocks are operations. Strategy blocks are diamond-shaped and the only one special-shape block in the matching operation symbolizes the final work, decision.**

4.2.2 Strategy blocks:

Strategy blocks here mean different procedures choosing to execute copy detection. We’ll see that with different techniques, the signatures can stand for a whole movie, whole scene, or just key frames in videos. There is only one strategy blocks in fig 4.1: ***"Key frames or not"***.

- **Key frames or not:**

From the paper I read, the signature extraction can executed on all the frames of a video/ a scene, or on several key frames of it. Key frames are simple yet effective form of summarizing a long video sequence, for example, the official website of a new released movie provides trailers or sets of pictures which give a compact description of the movie.

Why don't all the signature extraction techniques get features from every frame? The first reason is to make the process fast and fingerprints compact, we'll see that a technique which does complicated computation on a frame prefers to use key frames rather than all the video sequence for building the fingerprints. And the second one is to avoid frame drops. We'll discuss more later in 4.2.3.

Examples of techniques without key frames extraction are ***ViCopT*** from [2,4,6], ***Centroids of Gradient Orientations*** from [3], and ***Ordinal Histogram Feature***, ***YCbCr Histogram Feature from*** [5]. Examples of techniques with key frames extraction include ***Compact Fourier-Mellin Transform(CFMT)***, ***Scale Invariant Feature Transform(SIFT)*** from [5], and the technique from [7] published by A. Joly, etc.A special case comes from ***Motion Detection, Ordinal Intensity Signature,*** and ***Color Histogram Signature*** from [1], they do need key frames, but these frames can be randomly selected. Signatures are extracted from all the frames but only a part of them will be used in the matching step.

4.2.3 Operation:

Operation blocks here are the algorithms for extracting signatures. We can separate the work into three different steps: ***Boundary/ Scene Separation***, ***Key Frames Extraction***, and ***Signature Extraction***. These are the most technical steps of the whole CBCD system and I'll give several examples about them.

- **Boundary/ Scene Separation:**

This step focuses on detecting the boundary of scenes. There are kinds of CBCD methods based on signatures of scenes, the reasons are that frames in a scene are similar or related, and a copy clip may just a scene cut. A scene can be seen as an event in the movie, while another term, shot, means frames coming from the same camera or the same angle, and it's hard for computer to distinguish these two situations. Generally speaking, the histogram or color properties change a lot at boundaries of scenes, while sometimes these properties also change strongly at boundaries of shots. The basic method to detect boundaries of scenes is the ***measurement of histogram change***, but in movies, some boundaries of scenes has been processed to have little and slow histogram change, and even a scene can contain some sharp-histogram-changing points, which both make scene-boundary detection difficult. An effective algorithm is too complicated for us to generate, so in our experiment, we just use a software found online to do scene separation.

• Key Frames Extraction:

From 4.2.2, we have known that key frames are necessary for kinds of proposed techniques, and now let's talk more about the properties and extraction techniques. A video key frame is the frame that can represent the salient content of a video shot or scene. Key frames provide a suitable abstraction for video indexing, browsing, and retrieval. The extraction of key frames maps an entire video segment to a small collection of representative images, and should be automatic and content based so that they could maintain the salient content of the video while avoiding the redundancy.

A great set of key frames should give clear description of a video, and same key frames are expected to be extracted from the original video and its copy. To solve the frame drop situation, we assume even the selected key frame get lost, frames of similar content around it are probably to be selected and maintain the detection accuracy; while techniques without key frame may hardly to solve this problem.

Current key-frame-extraction techniques can be classified according to their various measurement of visual content complexity of a video shot or sequence, and the explanation is presented in [12].

• Feature Extraction:

This is the core of video fingerprinting. Selections of feature extraction methods can directly affect the copy detection performance. Here I'll just explain the idea of using information in a video to construct video fingerprints rather than introduce the proposed techniques, which will be included in another report, ***"The core of video fingerprinting"***. It's hard to clearly explain what kinds of features are suitable to construct video fingerprints since the development of this technique is used to solve existent video transformations or attacks. When a new attack is created, techniques proposed before may not deal with it. Thus, a new technique is needed. But the basic utilities of information of videos are worth discussing, and now I'll go back to the three basic dimensions, color, spatial, and temporal:

Color: simplest, the pixel values. Histograms won't consider where the pixel value comes but just a statistical case, and there're still similar methods such as hue, saturation, or even counting the mean and variance of a frame. Improved methods contain the concept of blocking, which means partitioning a frame into several blocks and store them in order to show the information of location.

Video fingerprinting has an interesting point that an arrangement of information of a frame is necessary, or we can say it needs mapped to a new space. Like using the YCbCr histogram, one method give 5 bins for each component while maintain the relation, so finally a 3D space is created; while the other one is a separated case, mapping a pixel into three histograms. The so-called color property is the nature of an image, but obviously, a histogram can map to different images, and that will be a disaster.

Spatial: going deep than the color property, this property includes relations among pixels. We can say if color is equivalent to the magnitude part of 2D-FT, then spatial is equivalent to the phase part.

One of the simplest ways to exploit spatial information is blocking. For each block, not only the histogram is usable, mapping it to a specific color space and use 3 colors as a symbol is also available. And for histograms, giving pixels near the center of a block higher weight is another way to bring in spatial information. More the spatial information is included, stronger the methods is. For example, using blocking for an image and calculating the average gray level, recording by histogram is worse than recording in order of the block locations.

Besides blocking, gradient around pixels is another way to get spatial information, and this computation can perform on each pixel or just the center pixel of each small-size block. Similarly, during storing, storing the data as histograms (small memory storage) or as sequences will affect the performance. In [3], gradient of each pixel in each block is computed and the magnitude part is used to weight the orientation. Finally a center-of-gravity-like quantity is reached for each block, and the frame signature is a vector with each element the block quantity arranged in order.

Local descriptors themselves exploit the spatial information, while a complicated issue is how to deal with the relation among each descriptor. For example, how to know that an image is just a rotation version of another image by local descriptors is really a hard question. Gradient or information among points is frequently used as description of points of interest, which means not only the location but also the environment have to be recorded.

Weak spatial information is just by blocking, while a stronger one is to use the relation among points, or even combine these two methods. Here the techniques of computer vision or object recognition can be used to enhance the performance. All techniques for image recognition and identification are worth trying on exploiting spatial information, while the computational time and data memory should be concerned.

Temporal: the difference between an image and a video, and the core of discussing video fingerprints. A simple methods compute difference images, but the difference of corresponding pixels between two frames can't pertinently present difference between two frames. Another simple method is to get each frame a vector then a video is recorded as a vector sequence. Stronger methods employ the motion vector [1] like the one used in video compression, and it's more symbolic for the frame difference. Other methods compute the gradient in both spatial and temporal dimensions [4], which is used for description of points of interest.

Techniques with key frames seems only exploit features from several frames, while key frames themselves contain the temporal information. They are important locations in the temporal axis, and a great key frame extraction technique can raise the distinguishability of videos. Key frame extraction is a little bit like prediction whose technique is self-defined and evaluated by its performance.

A simple method using frame extraction is to record the time period between each shot boundary and the temporal location of each boundary. The contents of these boundaries are worthless (usually black images), so only temporal information is reliable. And for short videos, it's likely to get same time period sequence of shot boundaries between two different videos, that's why this method is used for matching entire movies.

Other concepts:

(1) More dimensions (color, spatial, temporal), more robust.

(2) Local descriptor is stronger than global descriptor especially for logo and word insertion.

(3) Ordinal recording is better than actual value recording.

(4) It's better to record the direction of an image by some algorithms in order to deal with image rotation.

4.2.4 Final decision:

Final decision is the last step of video matching and copy detection, composed of ***searching***, a ***voting algorithm*** (***matching function***), and a ***threshold***. Searching means to find some candidate videos in the database which may contain the same content of the test clip, and the voting function is to decide which one is the best matching, finally the result is compared with a threshold to determine if the test clip is a copy or not.

Here I will introduce a concept about the storage of signatures. At first, two terms should be distinguished: ***storing*** and ***registration***. Storing means to arrange the information into a certain form; while registration means during the matching step, a candidate from database will be indexed with temporal, spatial position and variation.

In the signature extraction step, the information of an image is exploited, arranged, transformed to make a compact while identifiable description, and this description is stored for future copy detection. For global descriptions, the stored signature is an arranged form, which means it's hard to tell what the original video looks like just from the signature; while for local description, it's may be a different case. Some local description methods like SIFT [5] do arrange the information of interest points of a frame into a histogram form, but some methods still record the position of interest points at the end of signature extraction step, like ***ViCopT***.

For different storing situations (arrangement or not), the final decision steps are different. On searching step, for arranged signatures (especially global description), the result will be several candidate videos with temporal position, while for non-arranged signatures (points of interest), each point descriptor will have several candidate points in database which make the decision step complicated.

Voting function is to determine the best match. In [4], a general model is used to deal with rotation, zooming, translation and also slow/fast motion from the temporal point of view:

$$\begin{pmatrix} \mathbf{x}' \\ \mathbf{y}' \\ \mathbf{t_c}' \end{pmatrix} = \begin{pmatrix} \mathbf{r\cos\theta} & \mathbf{-r\sin\theta} & \mathbf{0} \\ \mathbf{r\sin\theta} & \mathbf{r\cos\theta} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{a_t} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \\ \mathbf{t_c} \end{pmatrix} + \begin{pmatrix} \mathbf{b_x} \\ \mathbf{b_y} \\ \mathbf{b_t} \end{pmatrix} \qquad (4.1)$$

where (x', y', t_c') and (x, y, t_c) are the spatio-temporal coordinates of two matching points.

The transformation model parameters are estimated for each retrieved video clip V_h using the random sample consensus algorithm [4]. Once the transformation model has been estimated, the final similarity measure *m(V_h)* related to a retrieved clip V_h consists in counting the number of matching points that respect the model according to a small temporal (for global features) and to a spatio-temporal precision for local feature.

For arranged signatures, general methods for similarity measure are distance measure, such as L1, L2 distance, or normalized correlation. And the smallest distance or the largest normalized correlation is determined as the best matching. While for non-arranged signature, we have to get the combinational result of each point descriptor to determine a possible candidate video sequence in the database, and the number of point matched will be a clue for determining the best match.

5. Brief Introduction of My Project

My project in UIUC is to generate some methods and compare their performance. There are few steps we have to work on: ***database creation, key frame extraction, feature extraction, matching function, performance measure, and the result.***

5.1 Database creation

Our test is based on clip by clip, so a video cutting algorithm is needed. We use a software "HandySaw DS" [16] to separate a movie into scenes and generate a database with 2600 scenes. Later we select 100 scenes of them to build duplicated videos, and the transformation we used are:

- **Frame drops:** 0%, 20%, 40%, 60% of frame drops.
- **Blurring:** two linear averaging filters to create two blurred videos. The size of

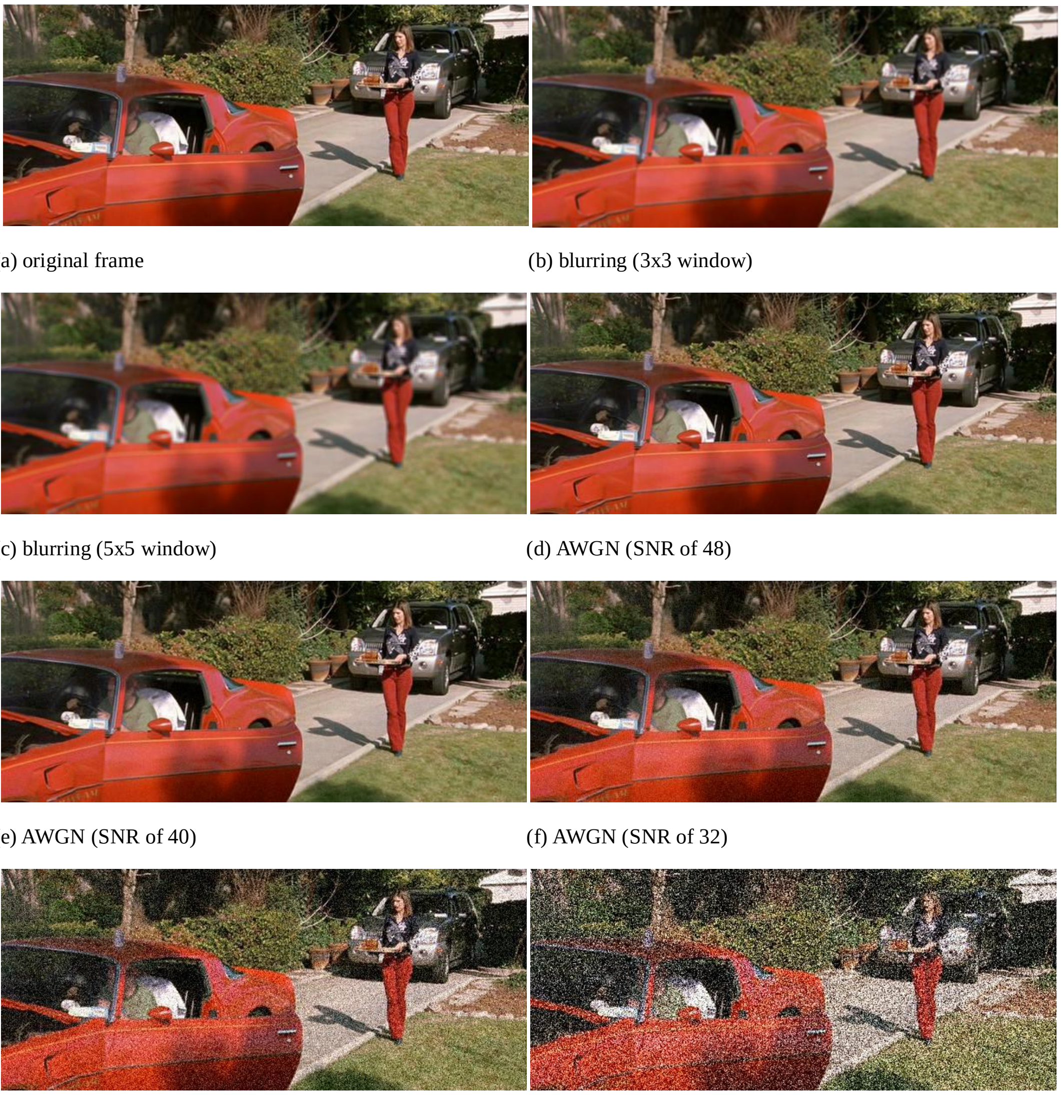
the squared averaging window is respectively 3 by 3 and 5 by 5 pixels. The effect of this filter is to attenuate the edges in the luminance component each frame.

- **Additive Gaussian noise:** SNRs of 8, 16, 24, 32, 40, and 48 are used in luminance component.
- **Gamma correlation:** +20 and -20 are used in the luminance component. The formula goes below:

$$\boldsymbol{I_{corrected}(x,y) = 255} \times (\frac{\boldsymbol{I(x,y)}}{\boldsymbol{255}})^{\boldsymbol{1\pm0.2}} \qquad (5.1)$$

- **JPEG compression:** quality factors of 10, 30, 50, 70, and 90 are used for the luminance component.

Samples are shown in figure 5.1. Then a scene is duplicated into 4(different cases of frame drop) × [1(original) +2+6+2+5] -1= 63 clips, then finally there will be 6300+2600 = 8900 clips in the database.



(a) original frame

(b) blurring (3x3 window)

(c) blurring (5x5 window)

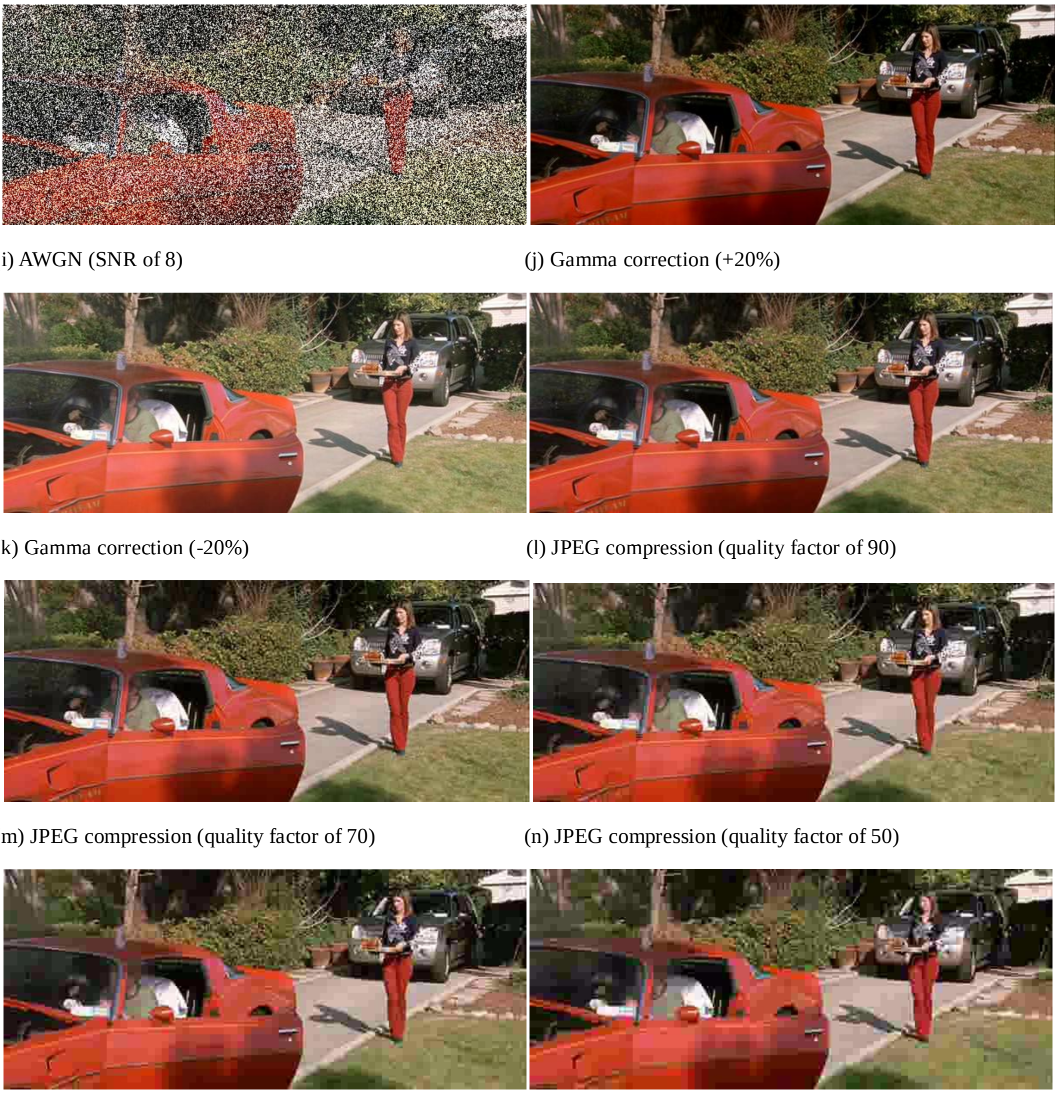
(d) AWGN (SNR of 48)

(e) AWGN (SNR of 40)

(f) AWGN (SNR of 32)

(g)AWGN (SNR of 24)

(h) AWGN (SNR of 16)



(i) AWGN (SNR of 8)

(j) Gamma correction (+20%)

(k) Gamma correction (-20%)

(l) JPEG compression (quality factor of 90)

(m) JPEG compression (quality factor of 70)

(n) JPEG compression (quality factor of 50)

(o) JPEG compression (quality factor of 30)

(p) JPEG compression (quality factor of 10)

Figure 5.1: Examples of our transformation methods.

5.2 Key frame extraction

Some methods of our project need key frame extraction, and the technique introduced in [16] is selected and modified to our requirement: ***8 key frames*** for each clip.

5.3 Feature extraction

Five techniques are used in our project:

- **Ordinal histogram**
- **YCbCr histogram**
- **CFMT**
- **SIFT**
- **Centroids of gradient orientation (CGO)**

And in the ***"The core of video fingerprinting"***, those techniques will be explained in detail.

5.4 Matching function

In our project, the matching function is the one described in [5], signature distance computation and its improved form. The basic formula is shown below:

$$d(\boldsymbol{X},\boldsymbol{Y}) = \sum_{i=1}^{K}\{min_{1\leq j\leq K} \ \| \boldsymbol{X}(i) - \boldsymbol{Y}(j) \|_1\} \qquad (5.2)$$

where $\| \boldsymbol{X}(i) - \boldsymbol{Y}(j) \|_1$ refers to the L1 distance computed between the ith frame of $\boldsymbol{X}$ and the jth frame of $\boldsymbol{Y}$. Thus, the distance relation is not symmetric: $d(\boldsymbol{X},\boldsymbol{Y}) \neq d(\boldsymbol{Y},\boldsymbol{X})$ in general, for each frame of $\boldsymbol{X}$ the best match from $\boldsymbol{Y}$ is selected. Thus, if frame drops occur or some video frames are corrupted by noise, they will not adversely affect the distance between two duplicated video, which would have been the case if an F-norm like distance is used, which compares the two signatures component-wise (assuming that the two videos follow the same temporal sequence). This is a closest-overlap distance rather than a sequential frame-to-frame distance.

This method is used SIFT and CFMT, while for ordinal, YCbCr histogram and CGO, a general L1 distance is used.

5.5 Performance measure

After all the ***Matlab*** scripts and the database have been generated, we are ready to evaluate the accuracy performance. I'll first describe our testing procedure then introduce two evaluation criteria used in our project, ***Precision recall curve*** and the ***F1 score***.

5.5.1 Our testing procedure

It's really an important part in our project. For practical cases, there are two kinds of detection procedures, one uses the signatures of videos in database to test if there is any copy on the Internet, and the other one tests a video from Internet with the database to see if it's a copy or not. The second one is more popular in practical because it aims at the possible copy rather than to test if there is any copy of a certain video.

In the second procedure, the practical case is that official, noiseless videos are used to generate the database, and the transformed videos form the Internet are tested to see if they are copies or not. While in our testing, the direction is opposite, duplicated videos are used to build the database and the original video is for testing. We have 8900 video clips in the database, and 100 original clips used to generate duplicated videos are tested with the database. For each original video, the best 100 matches are found and put in order (higher similarity to lower).

5.5.2 Precision recall curve

This is the most popular criterion use in video copy detection evaluation. The formula is:

$$\boldsymbol{Recall} = \frac{\boldsymbol{N_{TruePositive}}}{\boldsymbol{N_{\ AllTure}}} \qquad (5.3)$$

$$\boldsymbol{Precision} = \frac{\boldsymbol{N_{TruePositive}}}{\boldsymbol{N_{AllPostive}}} \qquad (5.4)$$

where $\boldsymbol{N_{\ AllTure}}$ is the number of duplicated videos of a scene in the database, and for our test it's 64. $\boldsymbol{N_{AllPostive}}$ is the number of retrievals done, in our case, we have already done 100 matches and put them in order, and from higher similarity to lower, we can see 1 to 100 retrievals together. For example, if the number of retrieval is 1, we only focus on the highest similarity match. $\boldsymbol{N_{TruePositive}}$ means the number of true detections, for example, if the number of retrievals is 10, then we look at the highest 10 matches and see how many matches are really copies of the test clip. If the number is 8, then the recall is 8/64 = 1/8 and the precision is 0.8 = 8/10 at this $\boldsymbol{N_{AllPostive}}$. A special case occurs when there are more than one precision values for a certain recall value, and we use the ***average precision*** [fig 5.2] for each recall value except for recall =1.0, at which only the highest precision value is recorded. Finally, 64 P-R values are computed and the P-R curve can be plotted.

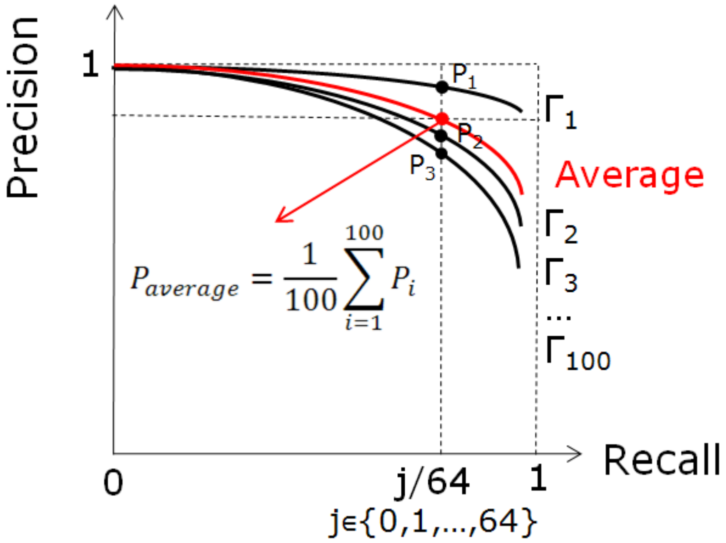


Figure 5.2: Classical shape of a precision-recall curve.

5.5.3 F1 score

The F1 score (also known as F-score) is a measure that simultaneously takes into account the precision and the recall to compute an "average" of these two criteria. It is defined as:

$$\boldsymbol{F} = \mathbf{2} \times \frac{\boldsymbol{P}\times\boldsymbol{R}}{\boldsymbol{P}\times\boldsymbol{R}} \qquad (5.5)$$

The value of F ranges from 0 to 1. The higher it is, the better the global accuracy is of a method. For each of the resulting 64 P-R pairs (average precision), the F1 score is computed and the highest point is found. From the highest F1 score, we can get the ***best retrieval times*** for a certain technique.

5.6 Result

Due to insufficient time of testing, only four methods are tested at the final result, except for CGO. And the trade-off concept between compactness and accuracy will be introduced in this section.

- **CFMT:** The fingerprint of a video scene is a ***matrix of 8 ×d coefficients***, where *d* is the number of dimension (number of eigenvectors) selected in feature extraction step. With higher *d*, the performance is better, which shows the trade-off between compactness and accuracy. Lower-dimensional signatures take short time and smaller memory while perform worse than higher dimension. Our result shows AWGN with SNR = 8 and 16 are the hardest cases to deal with. The best F1 score is with *d*=36.

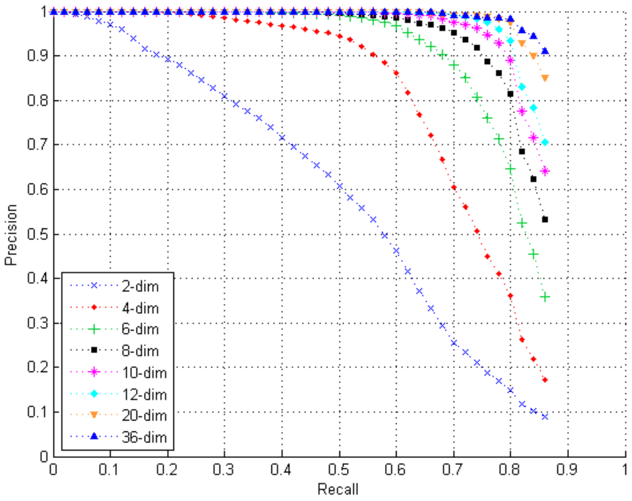
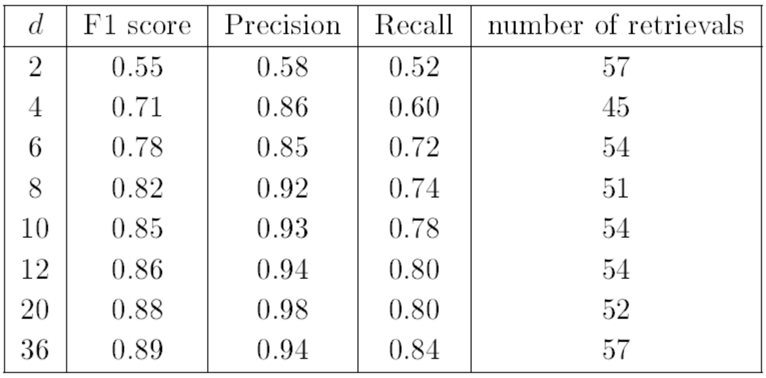


Figure 5.3: Precision –recall values of CMFT feature.



d	F1 score	Precision	Recall	number of retrievals
2	0.55	0.58	0.52	57
4	0.71	0.86	0.60	45
6	0.78	0.85	0.72	54
8	0.82	0.92	0.74	51
10	0.85	0.93	0.78	54
12	0.86	0.94	0.80	54
20	0.88	0.98	0.80	52
36	0.89	0.94	0.84	57

Table 5.1:F1 scores of CMFT feature

- **SIFT:** The fingerprint of a video scene is a ***matrix of*** $\boldsymbol{8 \times d}$ ***elements***, where *d* is the number of clusters selected in feature extraction step. Also the trade-off is shown in the P-R curve, and AWGN with SNR = 8 and 16 are still the hardest cases. Besides, SIFT performs worse than CMFT in JPEG and blurring cases while better in Gamma correction case. There is a special case that 120-dim signatures perform better than 320-dim signatures, and that's possibly because during the training process, due to the limitation of memory provided by ***Matlab***. The 320-dim case uses fewer samples for training the clusters so the result is worse. The best F1 score of SIFT is lower than CMFT.

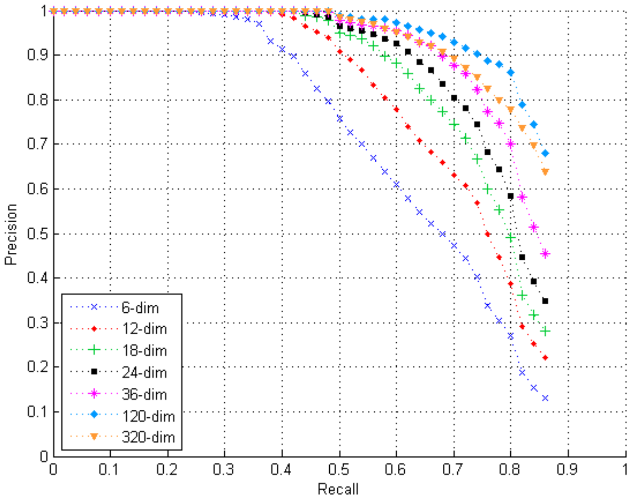
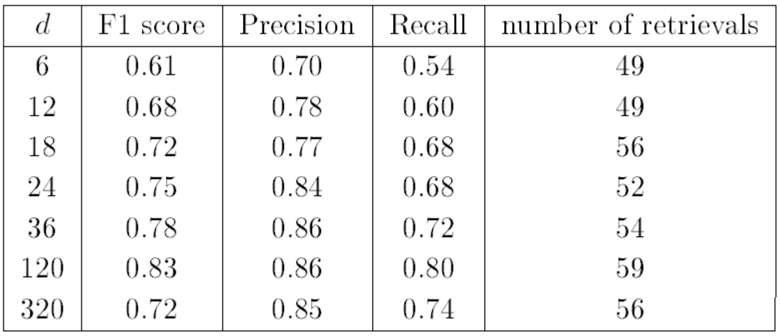


Figure 5.4: Precision –recall values of SIFT feature.



d	F1 score	Precision	Recall	number of retrievals
6	0.61	0.70	0.54	49
12	0.68	0.78	0.60	49
18	0.72	0.77	0.68	56
24	0.75	0.84	0.68	52
36	0.78	0.86	0.72	54
120	0.83	0.86	0.80	59
320	0.72	0.85	0.74	56

Table 5.2:F1 scores of SIFT feature

- **Ordinal histogram:** The fingerprint is a ***72-dimensional vector*** and depends on the length of the scene. The shape of the P-R curve is different from the one of CMFT or SIFT. The steps shown at recall =0.5 and 0.75 result from the frame drops of 40% and 60%, and this technique is extremely robust against other transformations. The F1 score is the highest among our testing.

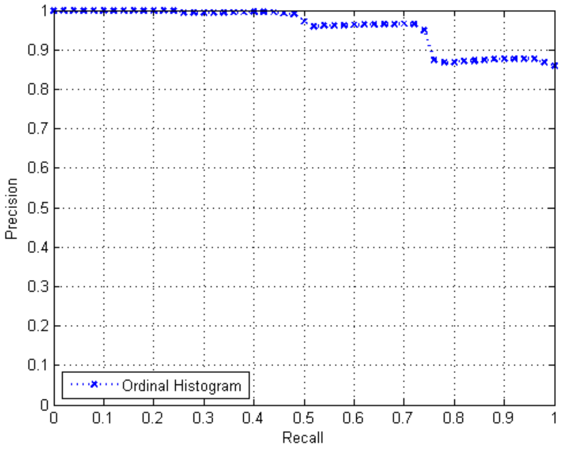
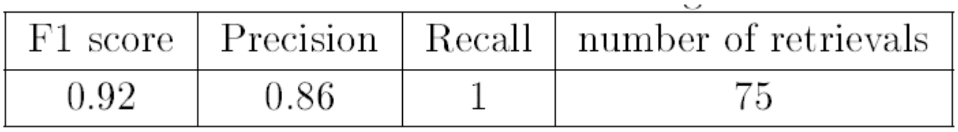


Figure 5.5: Precision –recall values of ordinal histogram feature**.**



F1 score	Precision	Recall	number of retrievals
0.92	0.86	1	75

Table 5.3:F1 scores of ordinal histogram feature

- **YCbCr histogram:** The fingerprint is a ***matrix of*** $\mathbf{8 \times 125}$ ***coefficients*** and depends on the scene length. This method is extremely sensitive to frame drops and also AWGN with SNR=8 and 16, so the performance is the worst in our testing. A possible improvement is to normalize the summation of the 125 coefficients to resist frame drops.

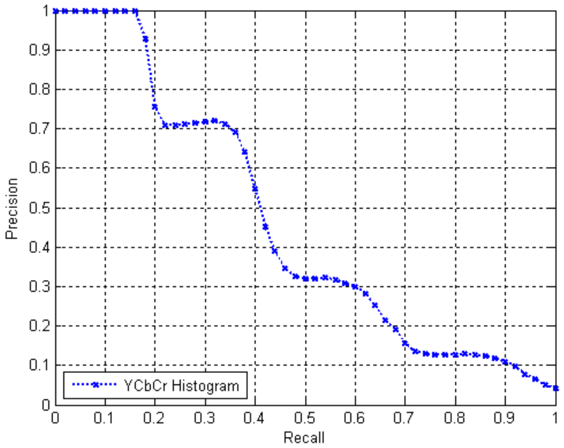
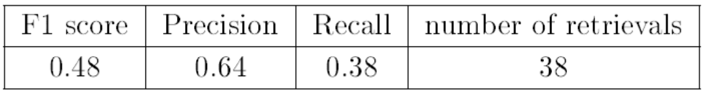


Figure 5.6: Precision –recall values of YCbCr histogram feature**.**



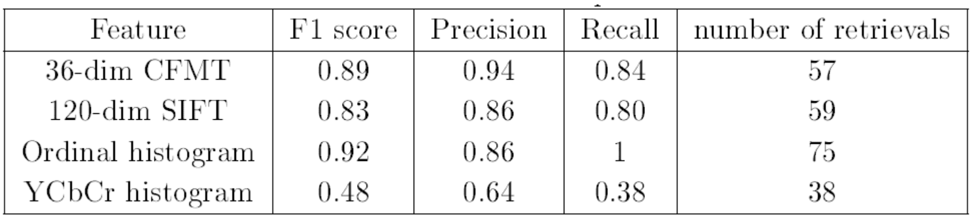
F1 score	Precision	Recall	number of retrievals
0.48	0.64	0.38	38

Table 5.4:F1 scores of YCbCr histogram feature

- **Comparison:**

Here we compare the best case of each technique: 120-dim feature for SIFT, 36-dim feature for CMFT, and ordinal and YCbCr histograms. The result shows that the two best features for duplicate video detection are CFMT and ordinal histogram depending on the recall values we consider. CFMT is better for recall values lower than 0.86 and ordinal histogram is better for recall values between 0.86 and 1. For key-frame-based signatures CFMT performs better than SIFT. This observation is also made in the research work of Manjunath et al [5]. This confirms thus the validity of our results.

The results obtained for ordinal histogram feature are better than the result presented in [28] but we use a smaller database (about 2600 scenes against 3800). It seems that the performance of this feature decreases more quickly than the performance of CFMT or SIFT when the size of the database increases.



Feature	F1 score	Precision	Recall	number of retrievals
36-dim CFMT	0.89	0.94	0.84	57
120-dim SIFT	0.83	0.86	0.80	59
Ordinal histogram	0.92	0.86	1	75
YCbCr histogram	0.48	0.64	0.38	38

Table 5.4:F1 scores of the best case of each technique

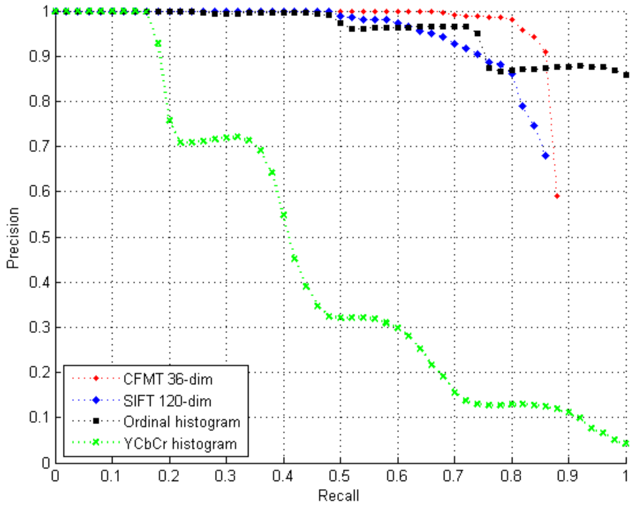


Figure 5.7: Precision –recall values of the best case of each technique.

6. Conclusion & Future Works

After the previous 5 sections, we have obtained the basic understanding about video fingerprinting. Now I'll present some questions I faced and some ideas of mine.

6.1 My question and answer

At first, almost all the feature extraction techniques are frame-based, which means exploiting information frame by frame. Later, features of frames are combined and modified to be the video fingerprints. Therefore, when studying ViCopT and writing this report, I always have a question, "Video fingerprinting uses the same idea of frame features, then why it cannot be seen as an application of researches of ***content-based-image-identification***?". In UIUC, I took lessons in Computer Vision for two weeks, and I'm surprised that the development of image identification is so comprehensive, and skills like object recognition, computer vision, corners and edges detection all can enhance the performance of identification. Then, why not just use these techniques for videos? Why the frame-feature-extraction techniques or video fingerprinting look relatively simpler?

These days, I gradually find the answer of my question. Image identification only focus on an image, while a 20-second-long video is likely to contain near 1000 frames, then the speed of matching should be taken into consideration. Besides, features of image identification can be extracted during the matching operation, while video copy detection should build a signature database for videos in the database. Therefore, techniques with highly complicated computation and great amount of memory requirement are not suitable for video fingerprinting.

There have been thousands and thousands of images and videos you can find on-line, for image identification, an image is matched with multiple images, so a detailed description is needed; while the video copy detection can be seen as multiple to multiple, where a frame sequence is compared with several frame sequences. And a video is unlikely to be composed of randomly picked images, therefore the relation between frame and frame is an important key for video copy detection. To make good use of it, then the computation cost on each frame could be significantly reduced.

Generally speaking, video fingerprints can be extracted from all the frames (sequence-based) or key frames, and the concept of trade-off between compactness and accuracy is necessary. Sequence-based extraction first extract features from each frame, then either combine these features into a vector sequence (here I call a frame feature a vector), or into a single vector. The advantage of a single vector is simple, fast matching, while it abandons the relation and order among frames, and a vector sequence remains it. Key-frame-based methods seem to lose the correlation among frames, while this relation has been used during the key-frame-extraction process, and the order among key frames can also be important information. It’s interesting to see that the frame-feature-extraction techniques of key-frame-based CBCD can also be used on frames of sequence-based CBCD, vice versa. Usually the techniques used on key frames are more complicated, since only features of key frames need computed and recorded, and then the cost on each key frame can be raised.

6.2 My ideas

There still differences between image matching and video matching. First, there is no temporal variation but spatial variation of pixel values in an image. Second, we may not store a slightly unclear image, while a slightly unclear video can still be tolerated due to its freshness and timeliness, for example, several bootleg videos and videos on the ***“Youtube”*** website are of poor quality but still popular. Therefore, when dealing with video copy detection, more cases should be considered.

6.3 Future works

Besides finding new features of videos and combining proposed features to enhance the performance, we still have to face new transformation challenges. In UIUC, professor Moulin gave me an example, “If a video or movie is cut into two pieces in space domain not time domain, which means separating each frame into two or more parts and transmitting them, how could we deal with it?”. And this may be a good topic for future researching.

Reference

[1]A. Hampapur, K. Hyun, and R. M. Bolle. Comparison of sequence matching techniques for video copy detection. volume 4676, pages 194_201. SPIE, 2001.

[2] J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Video copy detection on the Internet: the challenges of copyright and multiplicity. In ICME'07: Proceedings of the IEEE International Conference on Multimedia and Expo, pages 2082_2085, 2007.

[3] Sunil Lee and Chang D Yoo. Video fingerprinting based on centroids of gradients orientations. In ICASSP '06: Proceedings of the 2006 IEEE International Conference on Acoustics, Speech and Signal Processing, pages 401_404, Washington, DC, USA, 2006. IEEE Computer Society.

[4] 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, pages 371_378, New York, NY, USA, 2007. ACM.

[5] A. Sarkar, P. Ghosh, E. Moxley, and B. S. Manjunath. Video fingerprinting: Features for duplicate and similar video detection and query-based video retrieval. In SPIE – Multimedia Content Access: Algorithms and Systems II, Jan 2008.

[6]J. Law-To, O. Buisson, V. Gouet-Brunet, and N. Boujemaa. Robust Voting Algorithm based on labels of behavior for video copy detection. In ACM MM, 2006.

[7] A. Joly, C. Frelicot, and O Buisson. Robust content-based video copy idetification in a large reference database. CIVR 2003, LNCS 2728, pp. 414-424, 2003.

[8]Chris Harris and Mike Stephens. A combined corner and edge detection. In 4thAlvey Vision Conference, pages 153-158, 1988.

[9]S. Eickeler, S. Muller. Content-based video indexing of TV broadcast news using hidden markov models. In ICASSP, 1999.

[10]A. Joly, C. Frelicot and O. Buisson. Feature statistical retrival applied to content-based copy identification. In International Conference on Image Processing, 2004.

[11]Xinying Wang,and Zhengke Weng. Scene abrupt change detection.

[12]Tianming Liu, Hong-Jiang Zhang, and Feihu Qi. A novel video key-frame extraction algorithm based on perceived motion energy model.

[13]T.M. Liu, H.J. Zhang, F.H. Qi. A novel key frame extraction algorithm.

[14]L.J. Latecki, D.d. Wildt, and J. Hu. Extraction of key frames from videos by optimal color

composition matching and polygon simplification.

[15]Z. Sun. K. Jia, H. Chen. Video key frame extraction based on spatial-temporal color distribution.

[16] Video scene detection software Handysaw DS, June 2009.
http://www.davisr.com/cgi-bin/content/products/handysaw/description.htm