

CONTENT IDENTIFICATION USING VIDEO TOMOGRAPHY

By

Gustavo A. Leon

A Thesis Submitted to the Faculty of

The College of Engineering and Computer Science

in Partial Fulfillment of the Requirements for the Degree of

Master of Science

Florida Atlantic University

Boca Raton, Florida

August 2008

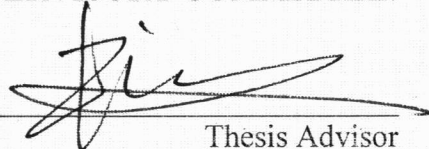
CONTENT IDENTIFICATION USING VIDEO TOMOGRAPHY

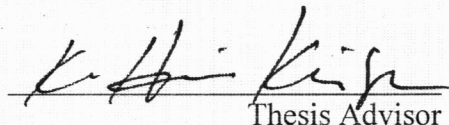
By

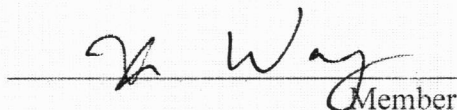
Gustavo A. Leon

This thesis was prepared under the direction of the candidate's thesis advisor, Dr. Hanqi Zhuang, Department of Electrical Engineering, and Dr. Hari Kalva, Department of Computer Engineering and has been approved by the members of his supervisory committee. It was submitted to the faculty of The College of Engineering and Computer Science and was accepted in partial fulfillment of the requirements for the degree of Master of Science.

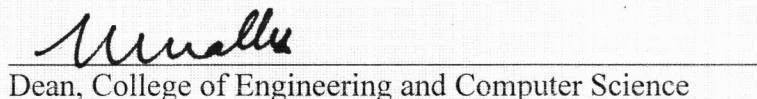
SUPERVISORY COMMITTEE:

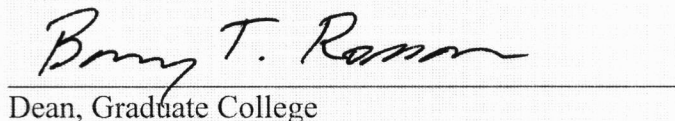

Thesis Advisor

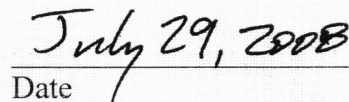

Thesis Advisor


Member


Chairperson, Department of Electrical Engineering


Dean, College of Engineering and Computer Science


Dean, Graduate College


Date

ACKNOWLEDGMENTS

I would like to thank my thesis advisor, Dr. Hanqi Zhuang his encouragement and help made this work possible. Also I wish to express sincere appreciation to Dr. Hari Kalva for his assistance in suggesting the research topic and supervising the research work. This effort would not have been made possible without his guidance and support, in addition to his familiarity with the needs and ideas on the topic throughout this. Dr. Xin Wang deserves my gratitude as a committee member of this thesis. Last but not least, the support of my family is what has defined me and the goals I have reached so far.

ABSTRACT

Author: Gustavo A. Leon
Title: Content Identification Using Video Tomography
Institution: Florida Atlantic University
Thesis Advisor: Dr. Hanqi Zhuang and Dr. Hari Kalva
Degree: Master of Science
Year: 2008

Video identification or copy detection is a challenging problem and is becoming increasingly important with the popularity of online video services. The problem addressed in this thesis is the identification of a given video clip in a given set of videos. For a given query video, the system returns all the instance of the video in the data set. This identification system uses video signatures based on video tomography. A robust and low complexity video signature is designed and implemented. The nature of the signature makes it independent to the most commonly video transformations. The signatures are generated for video shots and not individual frames, resulting in a compact signature of 64 bytes per video shot. The signatures are matched using simple Euclidean distance metric. The results show that videos can be identified with 100% recall and over 93% precision. The experiments included several transformations on videos.

CONTENTS

Chapter 1	INTRODUCTION	1
1.1	Overview and Motivation	1
1.2	Problem Statement and Objective	3
1.3	Contribution	4
1.4	Organization	5
Chapter 2	GENERAL BACKGROUND AND RELATED WORK	6
2.1	Introduction	6
2.2	General Definitions	6
2.3	State of the art	7
2.3.1	Digital Watermarking Detection	7
2.3.2	Content Based Copy Detection	8
Chapter 3	VIDEO SIGNATURE BASED ON VIDEO TOMOGRAPHY	11
3.1	Introduction	11
3.2	Video Tomography	11
3.2.1	Background	11
3.2.2	Video tomography to generate Digital Signatures	12
3.3	Canny Edge Detector	17
3.4	Distance Concept	20
3.5	Signature Generation	21
3.6	Shot Boundary Detection	24
3.6.1	Background	24
3.6.2	Motion and Tomography for shot Detection	26
3.7	A possible scenario	29
3.8	System Description	30
Chapter 4	IMPLEMENTATION	32
4.1	Introduction	32
4.2	Software Used	32
4.3	Obtaining Video Frames	33
4.4	Video Tomography	35
4.5	Signature Generation	38
4.6	Complete Process	39

4.7	Search Queries	41
Chapter 5	EXPERIMENTS AND RESULTS	42
5.1	Introduction.....	42
5.2	Boundary Detection	42
5.3	Content Identification	43
5.3.1	Description	43
5.3.2	Performance with shot boundary detection	45
5.3.3	Performance with constant shot lengths	46
5.4	Content Identification with Transformation.....	47
5.4.1	(CIVR2007) Competition	47
5.4.2	(CIVR2007) Competition Participants	48
5.4.3	(CIVR2007) Competition Database	49
5.4.4	(CIVR2007) Competition Task	49
5.4.5	Performance of the Proposed Solution	52
5.5	Signature Generation Complexity.....	55
Chapter 6	CONCLUSIONS AND FUTURE WORK.....	56
6.1	Introduction.....	56
6.2	Conclusions.....	56
6.3	Future Work.....	58
BIBLIOGRAPHY	59

TABLES

Table 1 - Results on boundary detection on TrecVid 2007 Database.....	42
Table 2 – Video Characteristics.....	43
Table 3 – Content Detection Results	46
Table 4 – (CIVR 2007) query videos description.....	51
Table 5 – Results on CIVR content	54
Table 6 – Performance Evaluation compared to other CIVR competitors	54

FIGURES

Figure 1 – Video Tomography for a video shot of S frames with dimensions WxH	13
Figure 5 – horizontal tomography (720x180) (pattern 1)	15
Figure 6 – the edges on horizontal tomography.....	15
Figure 7 – left diagonal tomography (720x180) (pattern 3)	15
Figure 2 – vertical tomography (180x480) (pattern 2)	15
Figure 3 – Snap shot of Shrek (720x480), 180 frames	15
Figure 4 – edge in vertical tomography	15
Figure 8 – edges in left diagonal tomography	16
Figure 9 – right diagonal tomography (720x180).....	16
Figure 10 – edges in the right diagonal tomography	16
Figure 11 – Canny Detector, Original gray level image	17
Figure 12 – Canny Detector, Binary image after Algorithm	17
Figure 13 - Canny Edge Detector blocks diagram.....	19
Figure 14 - Euclidean Distance Concept	20
Figure 15 – composite of horizontal and vertical tomography (180x180)	22
Figure 16 – composite of left and right diagonal edges (720x180)	22
Figure 17 – Level changes measured at eight equally spaced horizontal and vertical positions	23
Figure 18 – Spatio – Temporal patterns for different camera breaks	26
Figure 19 - edge pattern of a scene change (hard cut)	28
Figure 20 - edge pattern of a scene change (spin)	28
Figure 21 – Video Identification Scenario.....	29
Figure 22 – Signature Generation Process.....	30
Figure 23 – SIMULINK “from Multimedia File” configuration box”	34
Figure 24 - SIMULINK model for signature generation (Tomography Detail part 1).....	35
Figure 25 - SIMULINK model for signature generation (Tomography Detail part 2).....	37
Figure 26 - SIMULINK model for signature generation (signature generation).....	38
Figure 27 – SIMULINK model for signature generation (complete)	39
Figure 28 – shot distance against the entire database	44
Figure 29 – shot distance histogram	45
Figure 30 – (CIVR 2007) Video Stream Query.....	50
Figure 31 – (CIVR 2007) query 1	51
Figure 32 – (CIVR 2007) query 10.....	52
Figure 33 – (CIVR 2007) query 14.....	52

Chapter 1 INTRODUCTION

1.1 Overview and Motivation

Video detection also referred as video identification is an important problem that impacts wide applications including video copy detection, video indexing, online content distribution and video search. The main problem is determining whether a given video clip belongs to a known set of videos. There are several scenarios for video identification

With ever more popularity of video web-publishing more and more digital videos are available on the web and in multimedia databases, this content is being mirrored, re-formatted, modified and republished, resulting in excessive content duplication. An efficient algorithm to identify similar video clips can be beneficial to many applications.

Multimedia search engines are widely used; however, the retrieval efficiency is significantly hampered since a large number of search results are essentially copies of one another. It is detrimental to have all “near-duplicate” entries cluttering the top retrievals. Rather, it is advantageous to group together similar entries before presenting the retrieval results to users. In order to realize a useful browsing experience, one needs to detect and remove copies from the retrieval results before displaying the search results.

When a particular Web video becomes unavailable or suffers from slow network transmission, users can opt for a more accessible version among similar video content identified by the video search engine.

Similarity-detection algorithms can also be used for content identification when conventional techniques such as watermarking are not applicable. For example, multimedia content brokers may use similarity detection to check for copyright violation, as they have no right to insert watermarks into original material.

One related application is determining whether copyrighted videos or part of them are uploaded to video sharing websites. Be able to detect copies of digital media (in our case video) is a basic requirement in handling digital contents and protecting intellectual property rights (IPR). The IPR issue is also one of the main driving forces behind the proposed MPEG-21 standards. One scenario is movie studios interested in monitoring whether any of their content is used without authorization

A related problem is determining the number of instances a clip appears in a given source/database. Media tracking is the problem of keeping track of when and where a particular known piece of media has been used. In this scenario advertisers would be able to monitor how many times an advertisement is shown. For example, monitoring a particular TV commercial for market research is a specific application of the media tracking. In detail, one might want to know when and how many times, and on which channel a competitor's commercial is aired. By doing so, the competitor's marketing

strategy can be apprehended. It is also important for right management and royalty payments.

Video identification problem is challenging and the solutions fall into two classes:

The first set of solutions are based on digital watermark, digital watermarking based solutions assume an embedded watermark that can be extracted anytime in order to determine the video source.

The second set of solutions are based on content, content based copy detection has received increasing interest lately as this approach does not rely on any embedded watermarks and uses the content of the video to compute a unique signature based on various video features, the signature extraction is not required to be conducted before the media is distributed and this is the main advantage of this method over digital watermark.

1.2 Problem Statement and Objective

Most of the content based video identification methods operate with video signatures that are computed using extracted features from individual frames. Most of the time, the complexity of these video signatures are very high, therefore they are very expensive computationally speaking, furthermore there is no one method which can cover all the possible video transformations at the same time.

It is required to develop a video signature compact, unique and robust, unique implies that videos with different content should have distinct signatures, robustness

indicates the capability of change tolerance, which means that two videos with the same content should have identical or near the same signatures even if they have suffered the most commonly video transformations. This video signature is preferred to be computationally inexpensive not only to compute but also to compare.

1.3 Contribution

The following are the main contributions of this work:

- Design and implementation of a robust video identification system based on video tomography.
- Design and implementation of a simple and unique digital video signature, which contains enough information for later identification.
- Design and implementation of a low complexity metric to compare video signatures based on Euclidean distance.
- Establishment of a framework for future works in the area of video identification based on tomography.

1.4 Organization

The remaining chapters of this thesis are structured as follows:

- Chapter 2: Background and related work, a general description of the state of the art in video identification area as well as some basic definition in the video processing area.
- Chapter 3: Video Signature Based on Video Tomography, the tomography process and the video signature generation process are explained in this chapter.
- Chapter 4: Implementation, description of the algorithms used in this work and implementation details.
- Chapter 5: Experiments and results, shows the methodology of the experiments as well as the results obtained.
- Chapter 6: Conclusion, the conclusion and the future work are drawn in this chapter.

Chapter 2 GENERAL BACKGROUND AND RELATED WORK

2.1 Introduction

In this chapter the state of the art in the video identification area is given as well as an introduction to the general concepts and algorithms associated with this thesis are provided.

2.2 General Definitions

Video Shot: a video shot is created of a series of consecutives frames within a video. Video shots run for an uninterrupted period of time. Shots are generally filmed with a single camera and can be of any duration.

Video Scene: A consecutive series of frames that constitutes a unit of action in a video. It is usually composed of several shots.

Video Transcoding: video transcoding is the process to convert digital video signal from one codec/format to another, usually has three steps the first one is decoding the original format, then apply the required process, and the third step is re-encoding the video in the desired format. Video transcoding is necessary for those cases when bandwidth is limited, or resolution is lower (cell phone i.e.).

Video Cropping: refers to the removal of the outer parts of the frames of a video, video cropping is used, for example, when the screen size is smaller than frame size

Digital Watermarking: is the name given to the process which include information into a digital signal such as video, than later can be recognized, mostly is used for identification purposes.

2.3 State of the art

The solutions for video identification problem fall into two classes 1) digital watermark based video identification and 2) content based video identification.

2.3.1 Digital Watermarking Detection

Digital watermarking based solutions assume an embedded watermark that can be extracted anytime in order to determine the video source. Digital watermarking for video and images has been proposed as a solution for identification and tamper detection in video and images by G. Doer et al. [1]. While digital watermarking can be useful in identifying video sources, they are not usually designed to address the problem of identifying unique clips from the same video source. Even if frame-unique watermarks are embedded, the biggest obstacle of using watermarking is the embedding of a robust watermark in the source. Another issue is that large collections of digital assets without watermarks already exist.

The drawbacks of digital watermarking are being addressed in an emerging area of research referred to as *blind detection* [2] and [3]. Blind detection based approaches, like digital watermarks; address the problem of tampering detection and source identification. Unlike watermarks, blind detection uses characteristics inherent to the video and capture devices to detect tampering and identify sources. Nonlinearity of capturing sources, lighting consistency, and camera response function are some of the features used in blind detection. This is still an emerging area and some doubts persist about the robustness of blind detection [4]. Like watermarks blind detection approaches are not intended to identify unique clips from the same video. Both digital watermarking and blind detection are more suitable for tamper detection and source identification and are not suitable for video copy detection or identification.

2.3.2 Content Based Copy Detection

Content based copy detection has received increasing interest lately as this approach does not rely on any embedded watermarks and uses the content of the video to compute a unique signature based on various video features. A survey of content based video identification systems is presented by X. Fang et al. [5] and by J. Law-To et al. [6].

A content based identification system for identifying multiple instances of similar videos in a collection was presented in [7]. The system identifies videos captured from different angles and without any query input. Since the system is designed to identify

similar videos this is not suitable for applications such as copy detection that require identification of a given clip in a data set.

A solution for copy detection in streaming videos is presented in [8]. The authors use a video sequence similarity measure which is a composite of the frame fingerprints extracted for individual frames. Partial decoding of incoming video is performed and DC coefficients of key frame are used to extract and compute frame features. This method requires a lot of computing resources.

A copy detection system based on the bag-of-words model of text retrieval is presented in [9]. This solution uses SIFT descriptors as words to create a SIFT histogram that is used in finding matches. The use of SIFT descriptors makes the system robust to transformations such as brightness variations. Each frame has a feature dimension of 1024 corresponding to the number of bins in the SIFT histogram. But the system fails in cropping videos.

A clustering technique for copy detection was proposed in [10]. The authors extract key frames for each cluster of the query video and perform a key frame based search for similarity regions in the target videos. Similarity regions as short as 2x2 pixels are used leading to high complexity.

A content based video matching scheme using local features is presented in [11]. This approach extracts key frames to match against a database and then matches the local spatio-temporal features to match videos.

Most of these content based video identification methods operate with video signatures that are computed using features extracted from individual frames. These frame based solutions tend to be complex as they require feature extraction and comparison on frame basis.

Another common feature of these approaches is the use of key frames for temporal synchronization and subsequent video identification. Determining key frames either relies on underlying compression algorithms or requires additional computation to identify key frames. An important characteristic of video identification solutions is a robust and compact video signature that is computationally inexpensive to compute and compare.

Chapter 3 VIDEO SIGNATURE BASED ON VIDEO TOMOGRAPHY

3.1 Introduction

In this chapter the novel approach to the video identification problem is given. The video identification system proposed in this thesis uses spatio-temporal signatures based on video tomography. Video tomography captures the spatio-temporal changes in videos and is a measure of local and global motion in videos. The proposed video identification system is based on the hypothesis that the combination of local and global motion in a video clip can uniquely characterize and identify videos. Since the system is based on spatio-temporal changes the luminance signal is required to video to generate the signature.

3.2 Video Tomography

The definition for tomography is imaging by sections or sectioning. So sections of the object are taken to create a new object. Video tomography is the process of generating from a video sequence a new object, in this case an image.

3.2.1 Background

The proposed method of video identification is based on video tomography. Video tomography was first presented in ACM Multimedia '94 by Akutsu and Tonomura for camera work identification in movies [12]. Since then this approach has been

primarily explored for summarization and camera work detection in movies [13]. The images of video tomography in [12] and [13] reminded us of flow patterns of ridges in human fingerprints and thus began our exploration of video tomography for identification.

The initial thought was to exploit the work done in fingerprint analysis to extract signatures from video tomography. During the course of development we discovered the simple and elegant structure in video tomography and developed a video signature based on easily computable features.

The experiments conducted in this thesis verify that these video signatures are robust and uniquely identify video shots. This approach is robust to transformations such as recompression and is independent of the compression algorithms used. The video tomography is also referred to as spatiotemporal slices in the subsequent work [14]. The spatio-temporal slices were explored for applications in shot detection [15] and segmentation [16].

3.2.2 Video tomography to generate Digital Signatures

Video tomography is the process of generating tomography images for a given video shot. A tomography image is composed by taking a fixed line from each of the frames in a shot and arranging them from top to bottom to create an image.

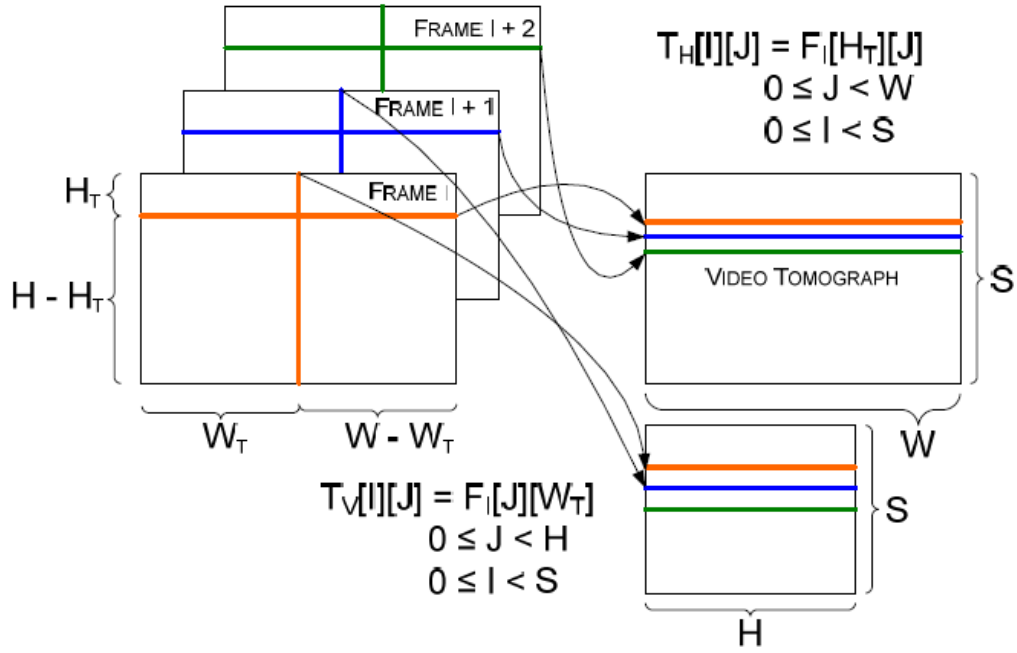


Figure 1 – Video Tomography for a video shot of S frames with dimensions WxH

Figure 1 illustrates the concept for a video shot of S frames. The figure shows horizontal tomography image, T_H , created at height H_T from the top-edge of the frame and a vertical tomography image, T_V , created at position W_T from the left-edge of the frame. The height of the tomography images is equal to the number of frames in a shot.

Others lines patterns can be used in addition to the vertical and horizontal tomography patterns shown in Figure 1; e.g., left and right diagonal patterns and any other arbitrary patterns.

The image obtained using the composition process shown in Figure 1 captures the spatio-temporal changes in the video. The position of the scan line (H_T or W_T) strongly affects the information captured in the video tomography. When scan lines are close to

the edge (e.g., $HT < H/5$) the tomography is likely to cut across background as most of the action in movies is at the center of the frame.

Any motion in a tomography that mainly cuts a static background would be primarily due to camera motion. On the other hand, with scan lines close to the center (e.g., $HT = H/2$) the tomography is likely to cut across background as well as foreground objects and the information in the tomography is a measure spatiotemporal activity that is a combination of local and global motion.

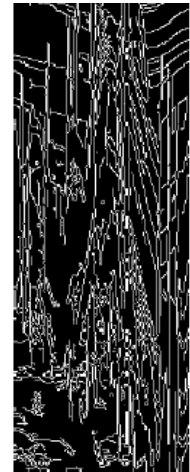
For video identification, capturing the interactions between global and local motion are critical and scan lines at the center of the frame are used. Horizontal and vertical tomography for a 180 frame shot from the movie Shrek is shown in Figure 4 to Figure 10. The tomography images are created using only the luminance component; this has the side effect of making the system robust to color variations.



Figure 3 – Snap shot of Shrek (720x480), 180 frames



**Figure 2 –
vertical
tomography
(180x480)
(pattern 2)**



**Figure 4 – edge
in vertical
tomography**

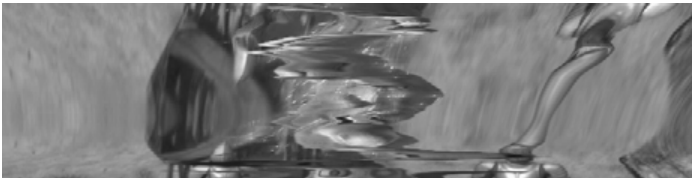


Figure 5 – horizontal tomography (720x180) (pattern 1)



Figure 6 – the edges on horizontal tomography



Figure 7 – left diagonal tomography (720x180) (pattern 3)



Figure 8 – edges in left diagonal tomography

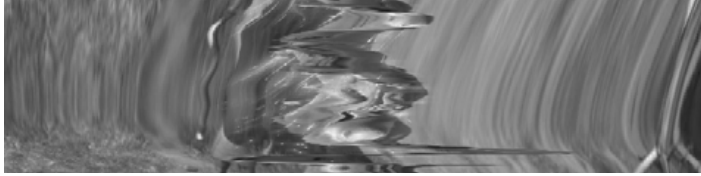


Figure 9 – right diagonal tomography (720x180)



Figure 10 – edges in the right diagonal tomography

The video resolution is (720x480). Figure 2 shows the vertical tomography and the corresponding edge image is shown in Figure 4. Figure 5 and Figure 6 show horizontal tomography as well as its edges, Figure 7 and Figure 8 are showing left diagonal tomography and its edge, Figure 9 and Figure 10 show tomography and edge image for right diagonal.

The edge image was created using the Canny edge detector. The edge image clearly reveals the structure of motion in the tomography. These edge images contain surprisingly rich information that can be used to understand the structure of the video sources. Such edge images are used to identify camera work in [12, 13]. These edge images are used in this video identification system for generating video signatures.

3.3 Canny Edge Detector

Canny edge detection algorithm is a multi-stage algorithm to detect a wide range of edges in images [17].



Figure 11 – Canny Detector, Original gray level image



Figure 12 – Canny Detector, Binary image after Algorithm

Figure 11 shows a gray image. Figure 12 shows the edges of the original image after canny edge detector algorithm.

Canny's aim was to discover the optimal edge detection algorithm. In this situation, an "optimal" edge detector means:

1. Good detection - the algorithm should mark as many real edges in the image as possible.
2. Good localization - edges marked should be as close as possible to the edge in the real image.
3. Minimal response - a given edge in the image should only be marked once, and where possible, image noise should not create false edges.

To satisfy these requirements Canny used the calculus of variations - a technique which finds the function which optimizes a given functional. The optimal function in Canny Detector is described by the sum of four exponential terms, but can be approximated by the first derivative of a Gaussian.

Canny edge detector blocks diagram is shown in Figure 13. The algorithm smoothes the image to eliminate and noise then finds the image gradient to highlight regions with high spatial derivatives using a Gaussian filter (in this case 3x3 pixels), after that the algorithm tracks along these regions and suppresses any pixel that is not at the maximum (non maximum suppression).

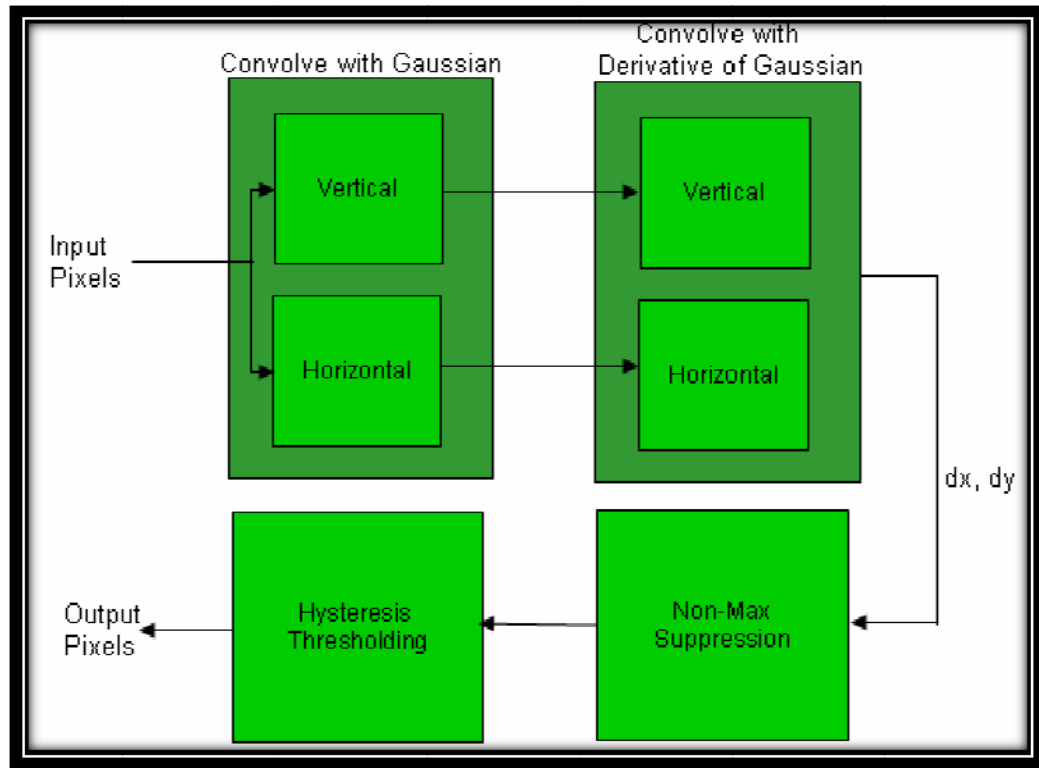


Figure 13 - Canny Edge Detector blocks diagram

Now using hysteresis the gradient array is reduced. Hysteresis is used to track along the remaining pixels that have not been suppressed. Hysteresis uses two thresholds and if the magnitude is below the first threshold, it is set to zero (made a non edge). If the magnitude is above the high threshold, it is made an edge. And if the magnitude is between the 2 thresholds, then it is set to zero unless there is a path from this pixel to a pixel with a gradient above the second threshold.

The Canny algorithm contains a number of adjustable parameters, which can affect the computation time and effectiveness of the algorithm.

1. The size of the Gaussian filter: the smoothing filter used in the first stage directly affects the results of the Canny Detection Algorithm.

2. Thresholds: the use of two thresholds with hysteresis allows more flexibility than in a single-threshold approach, but general problems of thresholding approaches still apply.

3.4 Distance Concept

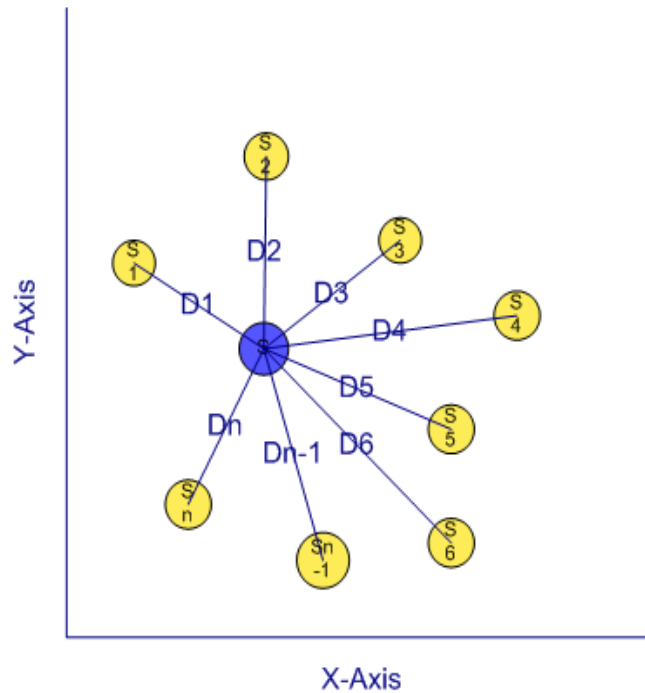


Figure 14 - Euclidean Distance Concept

Distance is a numerical description of how far apart objects are, in order to compare two videos signatures we need to use a distance in order to establish similarity. We can define the Euclidean distance for 2 vectors in two dimensions (x,y) as the length of the line segment that connects two points.

$$D = \sqrt{(Px - Qx)^2 + (Py - Qy)^2} \quad (3.1)$$

We can also extend the concept to a higher dimension n:

$$D = \sqrt{(P_x - Q_x)^2 + (P_y - Q_y)^2 + \dots + (P_n - Q_n)^2} = \sqrt{\sum_{i=1}^n (P_i - Q_i)^2} \quad (3.2)$$

This is the metric we are using to compare signatures once they have been created.

3.5 Signature Generation

Video signatures are designed to identify video clips uniquely. A clip can be a well defined shot that is S frames long or any continuous set of S frames. Video tomography for four scan patterns in a clip were analyzed 1) horizontal pattern at 50% (HT = H/2) 2) vertical pattern at 50% (WT = W/2) 3) left diagonal pattern and 4) right diagonal pattern.

The tomography images extracted from these four patterns have a complex structure reminiscent of fingerprints as shown in Figure 3 to Figure 10. The initial plan was to exploit tools in fingerprint analysis to extract signatures.

Fingerprint analysis uses combination of ridge endings and ridge bifurcations to match fingerprints [17]. To be able to use fingerprint analysis tools we needed to create enough artificial ridges and bifurcations in video tomography. Ridges and bifurcations in tomography are formed when lines representing motion flows intersect. One simple way of accomplishing this to combine tomography images created from different scan patterns.

The horizontal and vertical patterns were combined using an OR operation to create a composite image. A second composite image was created by combining the left and right diagonal patterns. The two composite images thus created form the basis for the video signatures. The composite images are visually as complex as a fingerprint as shown in Figure 15 and in Figure 16.

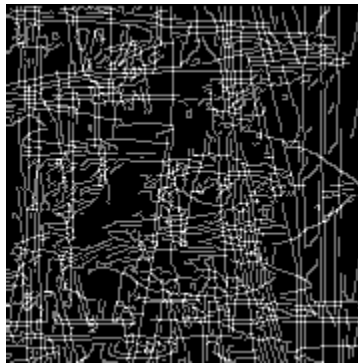


Figure 15 – composite of horizontal and vertical tomography (180x180)



Figure 16 – composite of left and right diagonal edges (720x180)

Before well known fingerprint analysis was applied, a simpler metric inspired by the *minutiae* in fingerprint analysis was developed.

The key constraint here is the ability to extract the features from exactly the same position in the composite image irrespective of the distortion a clip may suffer due to compression and other transformations. The metric used was the number of level changes

at discrete points in the composite images. The level changes were measured along horizontal and vertical lines at predetermined points in composite images. The number of such points determines the complexity and length of a signature.

Figure 17 shows eight horizontal and vertical positions used. At each of these positions on a tomography edge image, the number of level changes is counted; i.e., the black to white transitions representing the number of edges crossed along the line. This count can be as high as half the width of an image and is stored as a 16 bit integer.

The 16 counts on the horizontal-vertical composite and the other 16 edge counts on the diagonal composite form a 32 short integer signature for each shot. The signature size is always 32 bytes irrespective of the number of frames in the shot.

Since signatures are not created for individual frames, this solution results in a compact signature and the computational cost of finding a match is very low.

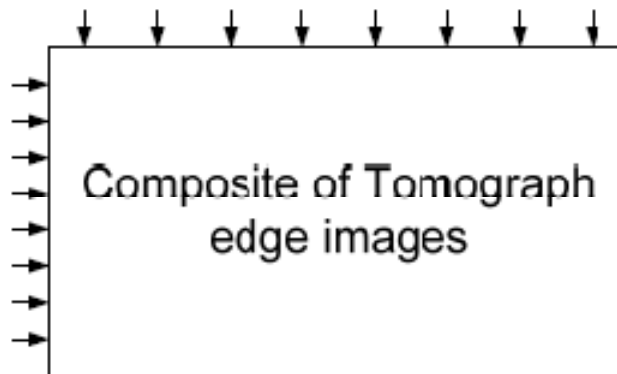


Figure 17 – Level changes measured at eight equally spaced horizontal and vertical positions

3.6 Shot Boundary Detection

To characterize video and uniquely identify it (or a portion of it), the original video needs to be segmented into respected scenes. Video identification typically requires shot detection to identify shot boundaries when signature databases are created by content providers as well as when a long video needs to be identified.

Video tomography has the potential to identify shot boundaries reliably. Since video tomography is generated for video signatures, shot boundary detection adds negligible complexity to the system.

3.6.1 Background

Video scene change detection or shot detection has been well studied problem over the last decade. It involves segmenting a given video into constituent shots. This the first step involved in video identification because each scene in a video is often unique and thus has unique motion vectors. Several approaches have been proposed for scene boundary detection. Most of the older methods use color histograms for identifying distance between frames [19].

Other methods include Pixel differences [20, 21], Statistical Methods [22], Compression Differences [23, 24], Edge tracking [25]. Most of the older methods have been studied and compared with experimental results in [27].

In [15], [16] and [26] the authors describe a scene change detection method based on motion. A scene, is described as a set of shots where as a shot is a single camera motion inside a video. A scene is comprised of similar shots.

For detecting scene change, they segment the video into various shots and then run a similarity analysis on the shots to group them into various shots. Spatio-temporal slices (similar to tomography) are used along with color information to segment the video into shots extracting motion fragments from each shot.

Each shot could have multiple motion units. They consider the fragment with the highest motion and reconstruct the background image. Based on the similarity measure of this they try to compare various scenes in a video. [15] Describes their similarity measure and video representation which is a color based similarity measure of the reconstructed background.

In [16], the authors introduce unique patterns for each camera breaks. They concentrate on cuts, wipes and dissolves and show how the Spatio-Temporal slices of a video show motion representation of a video into unique image patterns for each camera breaks. The Figure 18 shows the image patterns used.



















Camera Break	H	V	D
cut			
wipe (<i>l-to-r</i>)			
wipe (<i>r-to-l</i>)			
wipe (<i>t-to-b</i>)			
wipe (<i>b-to-t</i>)			
dissolve			

Figure 18 – Spatio – Temporal patterns for different camera breaks

3.6.2 Motion and Tomography for shot Detection

Tomography analysis as discussed previously provides valuable information as to what is happening in the video. The edge pattern of a tomography image of a video reveals easily comprehensible information related to shot boundaries. During a scene change in a video, the motion pattern (tomography) changes beyond a threshold. Due to this the edge pattern contains unique patterns for each scene transitions [15], [16].

Figure 19 below, shows one such pattern where a hard cut occurs. The position of the cut (each line in this pattern refers to a frame, as discussed previously), is encircled. For hard cuts, one can clearly observe a horizontal line at the shot boundary. Looking for horizontal lines in a tomography of a video is a simple way to detect hard cut scene boundaries in the image. Other such patterns are also obtained for different transitions. Figure 20 shows such a transition (spin – where the old frame spins out into a new

frame).The pattern can be clearly seen as a triangular flow. This makes tomography a very efficient method to detect even complex scene transitions using the same analysis algorithm. Also as the algorithm runs on the pixel domain data, it is independent of the compression method used.

Scenes detected using this scene detection algorithm based on tomography can serve as the basis for generating unique signatures for each shot.

The algorithm follows creating snapshots of such edge patterns for a video taking a constant number of frames per snapshot and looking for transition patterns (horizontal line in the case of a hard-cut). After a match is found with a sufficient threshold at a frame that frame is marked for signature generation. The threshold used here is the number of white pixels in a line measure as a percentage of the video width. A threshold of 65% implies that a line with more than 65% of white pixels corresponds to a hard cut. With varying thresholds we obtained considerable accuracy in our scene detection algorithm. In this version of the system only hard cuts are implemented.



Figure 19 - edge pattern of a scene change (hard cut)



Figure 20 - edge pattern of a scene change (spin)

3.7 A possible scenario

Figure 21 shows a use case where video owned by a content provider (e.g., Viacom) is distributed to users through one or more service providers (e.g., YouTube). A content provider creates a database of signatures for shots in videos.

When video is uploaded to video service providers, the service provider can extract signatures and query the content provider system for matches. Similarly, shot signatures can be generated while users are playing the video and content provider can be contacted for a match.

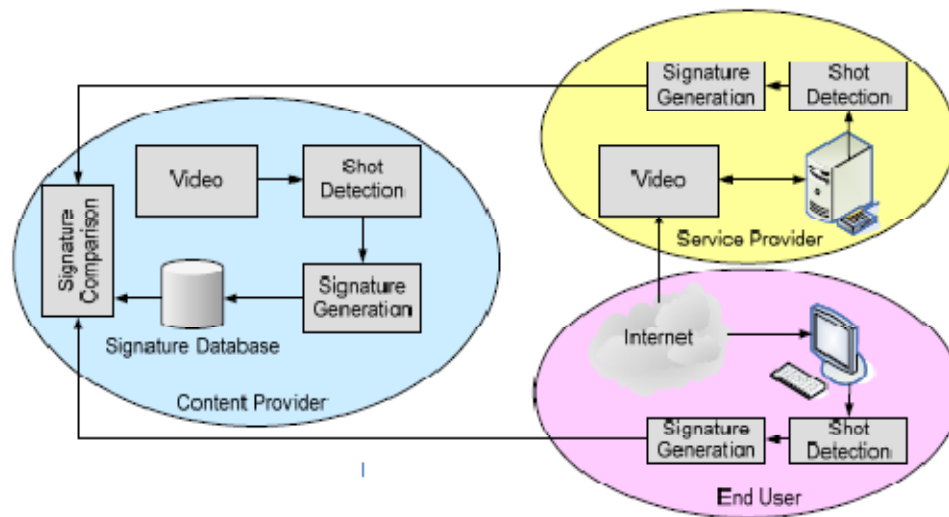


Figure 21 – Video Identification Scenario

This system can be used to identify unauthorized use of video or to monitor the consumption of certain videos (e.g., adverts), as has been said in chapter 1. When shot detection is used during signature generation, the same shot detection system is necessary

at the user side for reliable performance. It is also possible to bypass the shot detection and use clips of constant length for generating signatures.

3.8 System Description

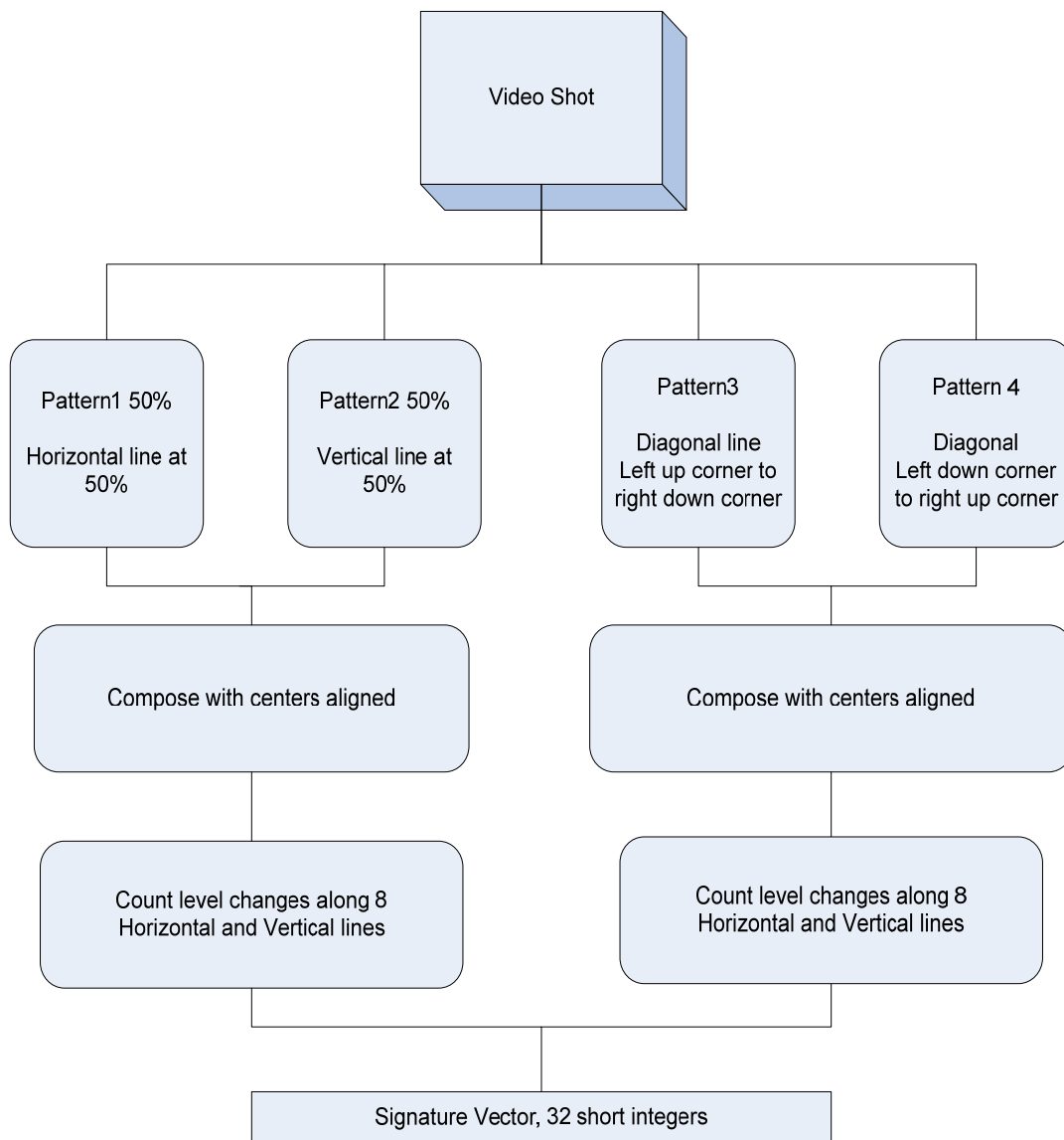


Figure 22 – Signature Generation Process

The signature generation has two principal stages video tomography for features extraction and signature generation.

- Video Tomography

In this process four images are created from a shot, four patterns are read in other words, from a constant number of frames four new images or patterns are created. Pattern 1 is created taking horizontal line at 50% of the height of the frame, if the frame has 480 pixels height the pattern 1 is the horizontal line located at 240 pixel height. Pattern 2 is created taking vertical line at 50% of the weight of the frame, if the frame has 640 pixels weight the pattern 2 is the vertical line located at 320 pixel weight. Pattern 3 is created taking the diagonal line, starting from the left up corner and end in the right down corner. Pattern 4 is created taking the diagonal line, starting from the left down corner and end in the right up corner. Then the canny edge detection algorithm is applied to these patterns, finally we have four binary images.

- Signature Generation

After the patterns are generated, they are composed as follows, pattern1 is compose with pattern 2 aligned by the center, pattern 1 and pattern 2 do not have the same size, so it is necessary to crop both images to make the composition. Pattern 3 is composed with pattern 4, they have the same size. The composition method is a logical OR. After this process we have two binary images. Count level changes, 8 horizontal and 8 vertical lines changes are registered for each image, the final video signature is a vector of 32 short integers.

Chapter 4 IMPLEMENTATION

4.1 Introduction

This chapter describes the algorithms used in the thesis, the details and considerations using to design the tools. All of the programs used in the course of this thesis were made in Microsoft Visual C++ [28], and in MATLAB [29], Microsoft Excel was used for trivial task like data analysis and plot generation but not for major tools. The complete process for signature generation and comparison is implemented in visual C++ and in MATLAB.

4.2 Software Used

MATLAB/SIMULINK is a numerical computing environment and programming language. It offers the advantage of a fast implementation, it has a lot of tools for video and image processing, including edge detection algorithms, also the matrix manipulation is one of the advantages compared to C++, and we must not forget the graphics tools for visualization and analysis. The main disadvantage is that the programs only run in MATLAB.

Visual C++ has the advantage of being able to create stand alone applications. We should mention that it is one of the most popular programming languages for multimedia applications.

Intel C++ Compiler [30] is a compiler which can be integrated with Microsoft Visual C++; it is an amazing tool that improves the performance of any program written for C++, for example using the dual processing capabilities of new processors.

Intel's Integrated Performance Primitives [31] (Intel IPP) is a library of multi-core-ready, optimized software functions for multimedia and data processing applications, one of the libraries includes the Canny edge detector algorithm, and decode/encode for different video formats.

4.3 Obtaining Video Frames

The first step to start working with video is being able to have the video frames in plain format, this is, the YUV format of the video, to be extracted we only need Y. The proposed solutions only need the intensity of the video signal to create the signature.

At the beginning of this project a YUV file was created using several applications for video processing, every video we want to analyze had to be converted to YUV, that means huge size files, just to have an idea, 20 Giga bytes only contains 40 minutes of video.

I was trying to integrate the video signature generation process to a player, in order to read frames in the original format, MPG i.e., but it becomes a very hard task and we consider it out of scope for this thesis purposes.

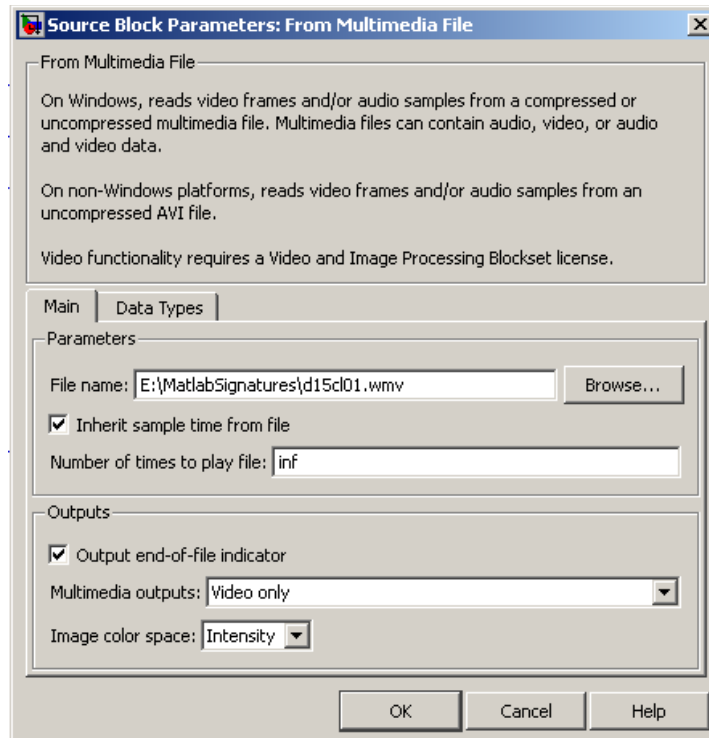


Figure 23 – SIMULINK “from Multimedia File” configuration box”

MATLAB has a set of function to manipulate the most common video formats; it has been decide to complete the final experiments using MATLAB, due to the large number of video to be processed and the nature of the analysis. SIMULINK only requires one block called “From Multimedia File” to manipulate any video file. Figure 23 shows the dialog box for the block.

4.4 Video Tomography

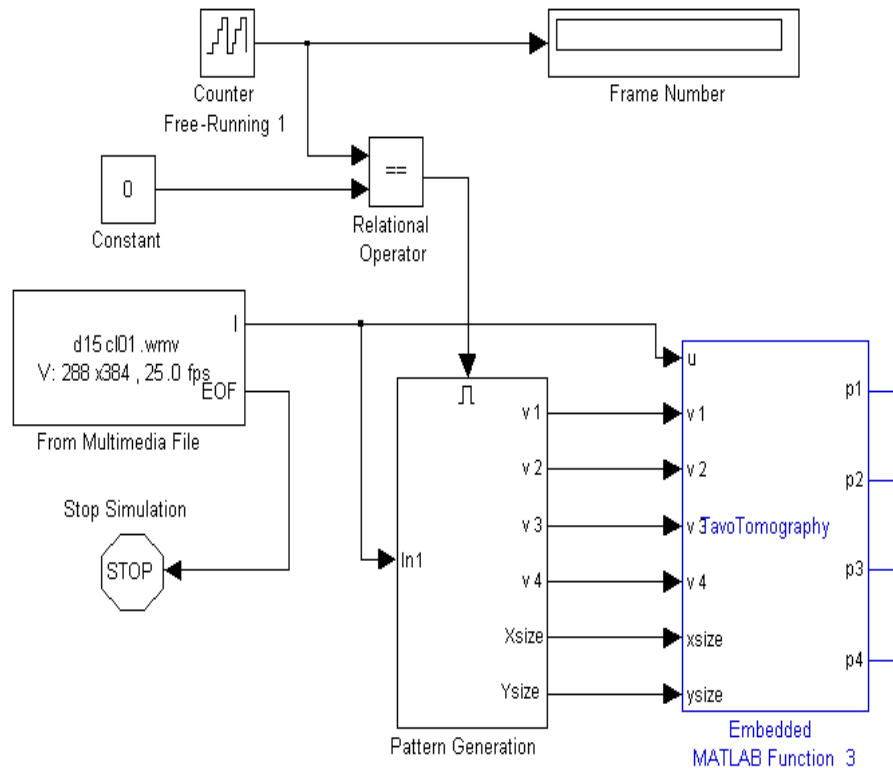


Figure 24 - SIMULINK model for signature generation (Tomography Detail part 1)

This process is relatively easy to implement, just reading some pixel values, for description purposes I will use the MATLAB/SIMULINK code to explain the process.

Once the intensity frame is read, it goes to the “TavoTomography” block which calculates the four patterns. “Pattern Generation” block is enabled only during the first frame; it is used for configuration, after the first frame the block is turned off. “Counter” block show the frame number.

The following code is the one within the “TavoTomography” block. The first while loop reads pattern1, pattern 3 and pattern 4. The second while loop reads pattern 2.

```
function [p1,p2,p3,p4] =
TavoTomography(u,v1,v2,v3,v4,xsize,ysize)

p1=u (1, :);
p2=u (:,1);
p3=u (1, :);
p4=u (1, :);

i = uint16 (1);
while i <= xsize,
    p1(i) = u(v1(i),i);
    p3(i) = u(v3(i),i);
    p4(i) = u(v4(i),i);
    i=i+1;
end

i = uint16(1);
while i<=ysize,
    p2(i) = u(i,v2(i));
    i=i+1;
end
```

The patterns creates 4 new images, in Figure 25 the buffers save the pattern until complete S – frames (Pattern 2 needs a transposition because is vertical), when S frames are read they go to the edge detector (one block in MATLAB).

The buffers initially are empty, they are being filled every frame which is read in the video, when the buffer completes its capacity of S – vectors. They put in the output line the matrix. The edge detector receives a matrix every S frames.

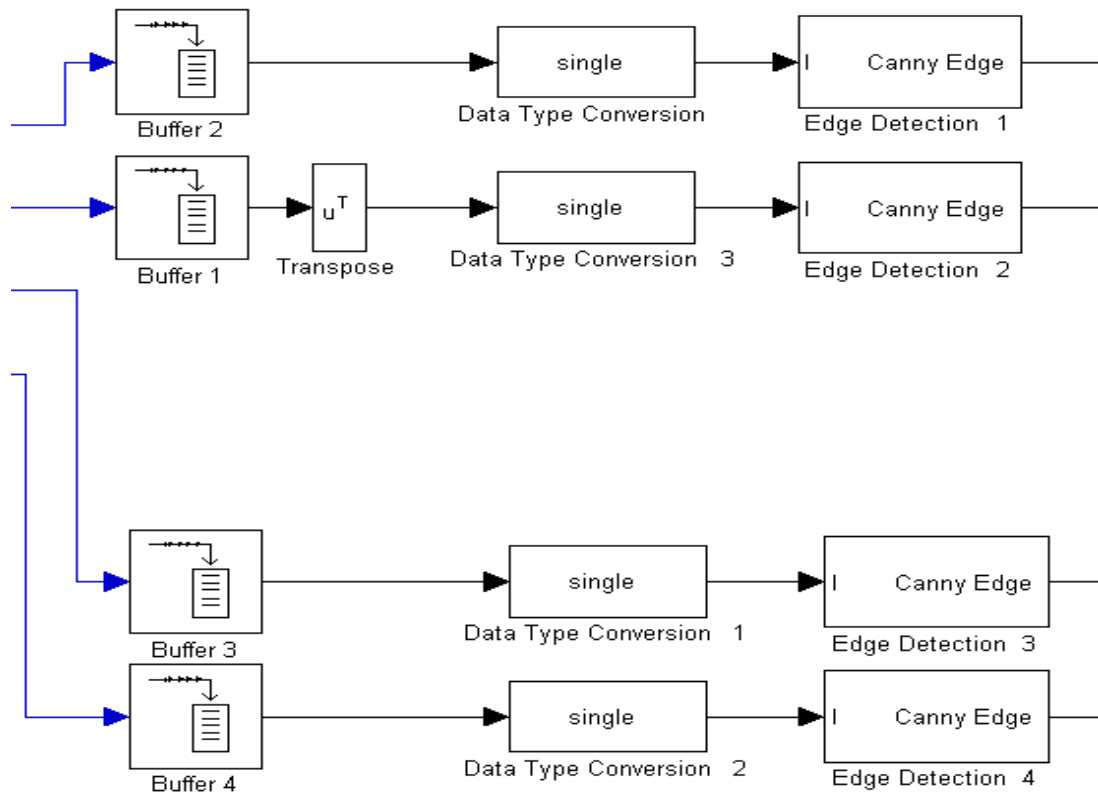


Figure 25 - SIMULINK model for signature generation (Tomography Detail part 2)

A data conversion block is required because the input to the edge detector block has to be double, and the data read from the video is unsigned integer of 8 bits. Edge detector in MATLAB is calculated using “edge” function which is included in the image processing toolbox. Edge detector in C++ is calculated using “edge” function which is included in the IPP library.

4.5 Signature Generation

The OR for pattern1 and pattern 2 has to make a crop before the logical or.

Pattern 3 and pattern 4 go to an OR block.

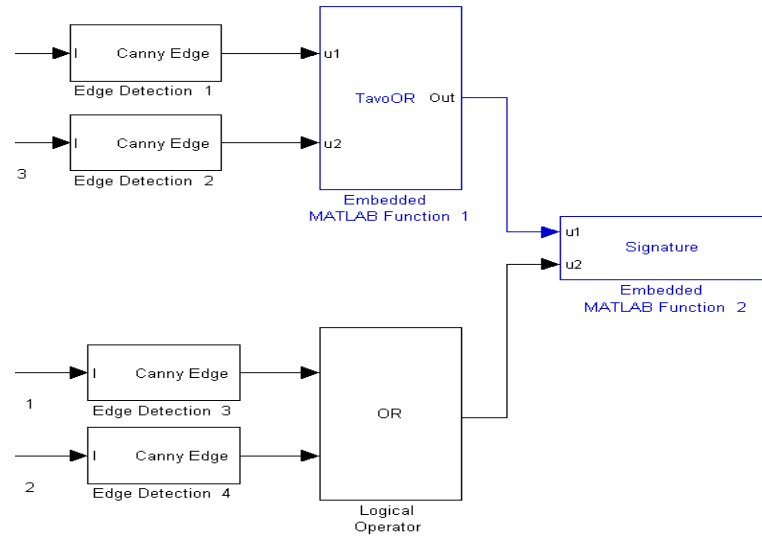


Figure 26 - SIMULINK model for signature generation (signature generation)

In the “Signature” block the counting of level changes is made and the signature vector is created. The signature vector (32 short integers) is saved into a file in binary format.

4.6 Complete Process

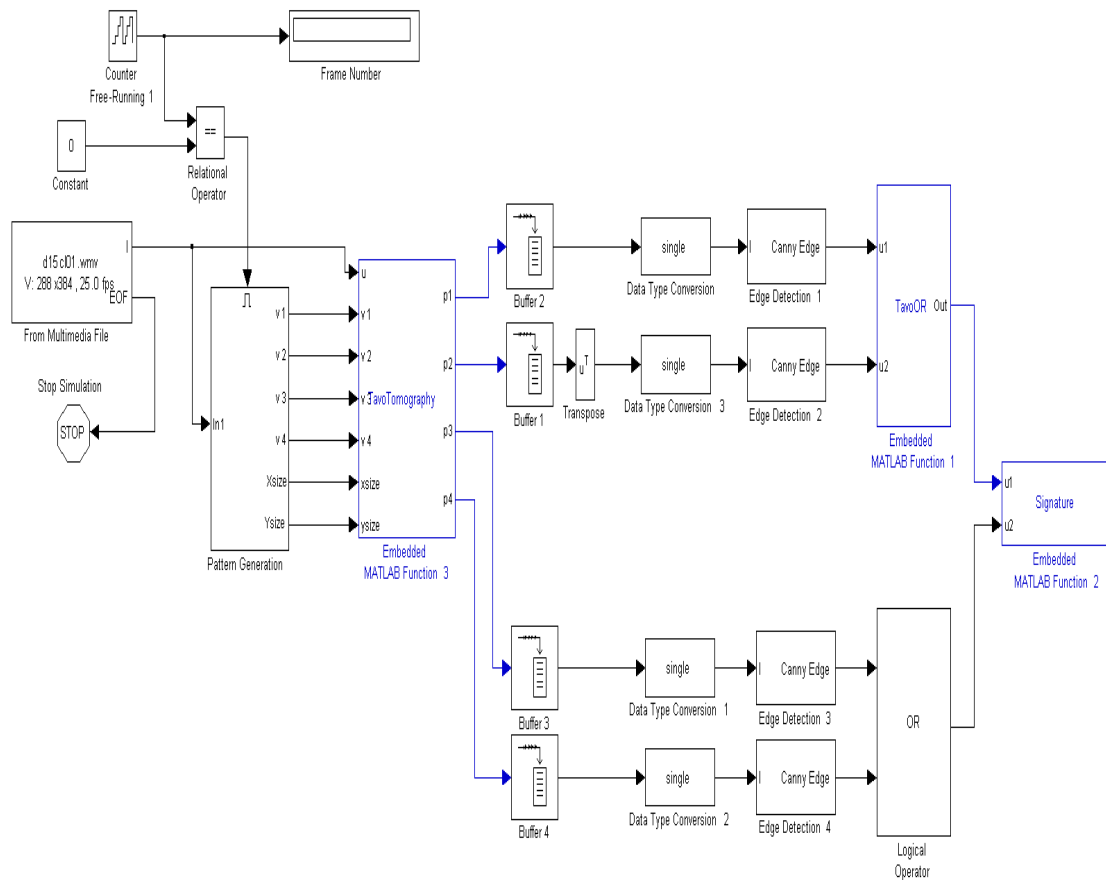


Figure 27 – SIMULINK model for signature generation (complete)

The MATLAB program for video signature generation, follows the SIMULINK model in Figure 27, the difference is the way the frames from input video are read. SIMULINK reads one frame at a time, MATLAB uses “mmreader” function which creates an object that contains the multimedia file, S frames are read at a time, in the following code can be seen how the variable “VIDEO” contains a shot of S frames.

```
Obj = mmreader (FileIn);
VIDEO = read (obj, [FrameNum FrameNum+WindowSize-1]); %Read Frames
```

The frames read by “mmreader” are in RGB format, it is necessary to calculate the intensity using a linear equation.

```
FRAME = 0.2989 * VIDEO (1) + 0.5870 *
VIDEO (2) + 0.1140 * VIDEO (3); %Y Component
```

The MATLAB program for video signature generation is a function with this header, where “FileIn” is any video file. “Window Size” indicates S, the number of frames used to calculate the signature. “Out” indicates false when any error occurs, else indicates true.

```
Function out = Video Signature (FileIn, Window
Size)
```

C++ code reads from YUV file, works similar to MATLAB code, but in C++ the vector operations are done with FOR loops.

In C++ code there are more flexibility in terms of windows sizes and gap. These and other parameters were used to explore different approaches to the signature generation problem. But the need of use YUV files makes the simulations to run very slow. First was necessary to convert the video and then the program has to deal with huge files.

4.7 Search Queries

Once the signatures are generated and the database is created, how it is determine if a new query video match any video in the database?. The first thing is to generate the signature for the query, now the comparison is made between signatures. MATLAB is used to made the comparison because its graphics tools, and because is fast generate reports. Basically the program returns the video with the shortest distance, (also checks video sizes i.e.). More detailed information is given in experiments chapter.

Chapter 5 EXPERIMENTS AND RESULTS

5.1 Introduction

This chapter presents the experiment conducted in order to validate the video identification process proposed in this thesis. Three sets of experiments were conducted; boundary detection, content detection and content detection with transformation.

5.2 Boundary Detection

TRECVID 2007 [27] had a stream for shot boundary detection on a set of videos where the ground truth is available. The experiments have been run on their set. The results are summarized in Table 1

threshold	Results	
65%	Average Recall:	90%
	Average Precision:	91%
60%	Average Recall:	77%
	Average Precision:	93%

Table 1 - Results on boundary detection on TrecVid 2007 Database

The results show that the proposed approach can reliably detect cuts in videos. At 60% threshold the precision is 93% but the recall rate is low. Increasing the threshold to 65% reduces the precision slightly but significantly improves the recall rate resulting in a

fairly accurate boundary detection solution. A fairly accurate shot detection solution is sufficient for use in video identification systems.

5.3 Content Identification

The goal of this experiment was to determine if the system can identify correctly the origin of a shot that has been taken from one of four possible movies, and with no processing made over the shot.

5.3.1 Description

The performance of the proposed solution was evaluated using up to 50,000 frames from four video: Shrek 1, Shrek 2, Pirate of the Caribbean, and one NFL Football game. The Pirates video also included advertisements. Table 2 gives the description of each video. The implementation for this experiment works only with uncompressed videos and the large space requirements of uncompressed videos make evaluation with longer videos difficult.

Video Number and Name	Number of Frames	Resolution
1. Shrek 1	39,578	720 x 480
2. Shrek 2	24,069	720 x 480
3. Pirates	50,000	1280 x 720
4. Football	50,000	1280 x 720

Table 2 – Video Characteristics

Video signatures are generated for the first 90 frames of long shots. All the signatures are combined in a video database. The performance is measured by searching for a given shot and match is recorded if exactly one shot is identified. Shots are identified by measuring the Euclidean distance between the query shot and all the shots in the database. Since a simple distance metric is used, the complexity of searching for a match is very low. Different distance thresholds are evaluated. The experiments are repeated without shot boundary detection and using constant clips of length 90 frames.

Figure 28 shows the result of a query for a shot. The figure shows that there is only one match and the average distance of the signatures in the matching video is smaller than the other three videos. Shots distances are show for the first 73 shots for improved readability. The distance is the Euclidean distance between video signatures (vectors of dimension 32).

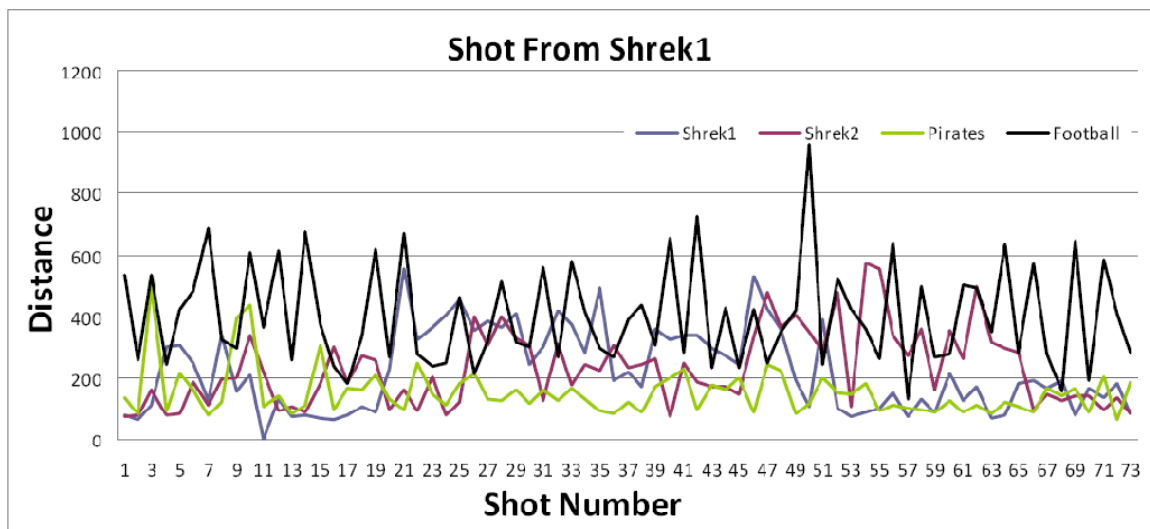


Figure 28 – shot distance against the entire database

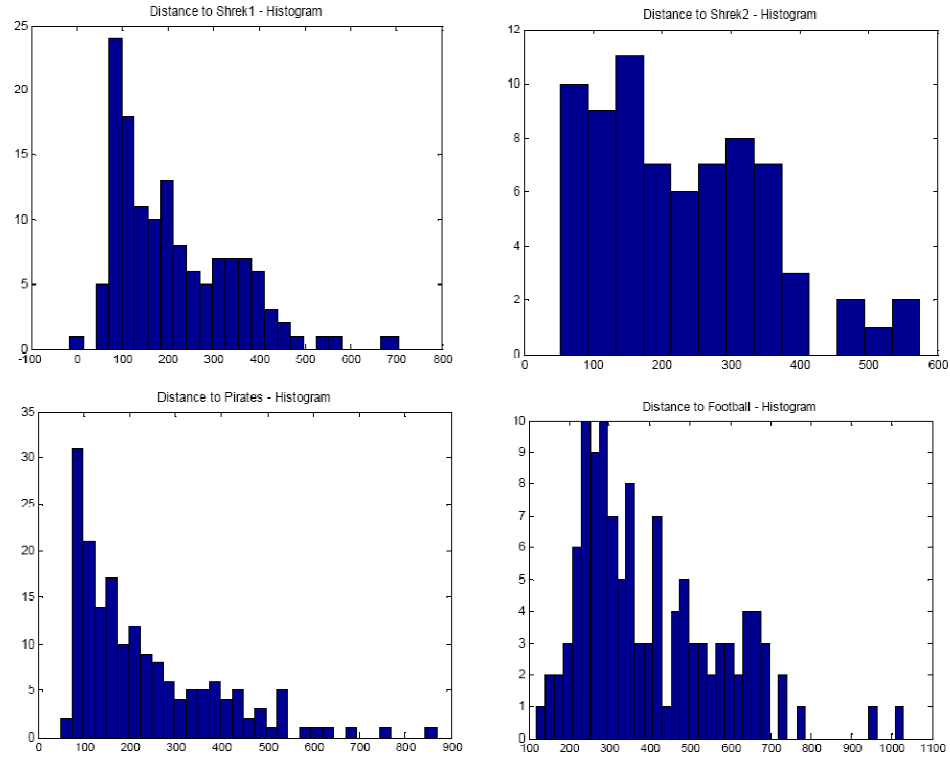


Figure 29 – shot distance histogram

Figure 29 shows the histograms of shot distances for the shot match shown in Figure 28. The histograms show that the closest matches are in Shrek 1. The results indicate that the video signature used has a good discriminating ability and serves as a good basis for video identification systems.

5.3.2 Performance with shot boundary detection

The performance of the system with shot boundary detection is summarized in. The results show that every shot was successfully identified and has a recall rate of 100%. The precision however is greater than 97% for a distance threshold of 5. This

means by setting a distance threshold to 5, most of the shots can be accurately identified with very small number of shots incorrectly identified as the given query. In these experiments, the query returns all matching shots with a distance less than the threshold.

I examined the cases where multiple minimums were returned and found that the problem occurs when there are blank frames due to shot transitions and ad switching.

5.3.3 Performance with constant shot lengths

With constant clip lengths, the shot detection module is bypassed and signatures are generated for shot lengths of 90 frames. The results of this experiment are summarized in Table 3. The results show that recall is 100% and precision is over 97% for a threshold of 5; as threshold increases the systems returns more false positives. The constant clip length signatures may be appropriate when shot detection is not efficient or for systems where the same shot detection system cannot be used at the content provider and service provider sites.

Number of Shots	Video	Signature Bytes	Average Distance 1	Average Distance 2	Average Distance 3	Average Distance 4	Std Deviation Distance 1	Std Deviation Distance 2	Std Deviation Distance 3	Std Deviation Distance 4	# Distance < 05, Recall	# Distance < 10, Recall	# Distance < 20, Recall	# Distance < 40, Recall	# Distance < 60, Recall	# Distance < 05, Precision	# Distance < 10, Precision	# Distance < 20, Precision	# Distance < 40, Precision	# Distance < 60, Precision
With shot boundary detected																				
137	1	32	184.747	186.394	200.099	275.559	95.412	91.880	115.101	147.857	1.000	1.000	1.000	1.000	1.000	0.972	0.972	0.926	0.381	0.067
73	2	32	186.394	183.314	200.292	260.467	89.048	88.544	108.470	143.074	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.987	0.503	0.080
176	3	32	200.099	200.292	210.430	281.431	95.649	93.284	119.515	145.396	1.000	1.000	1.000	1.000	1.000	0.989	0.989	0.946	0.251	0.059
118	4	32	275.559	260.467	281.431	228.885	95.033	87.800	107.274	124.080	1.000	1.000	1.000	1.000	1.000	0.983	0.983	0.983	0.874	0.359
With constant shot length of 90																				
430	1	32	207.878	206.022	217.413	302.686	91.836	92.742	110.660	154.122	1.000	1.000	1.000	1.000	1.000	0.982	0.964	0.890	0.317	0.046
261	2	32	206.022	202.666	214.257	301.349	89.028	92.308	109.670	154.441	1.000	1.000	1.000	1.000	1.000	0.970	0.970	0.903	0.274	0.039
549	3	32	217.413	214.257	221.597	311.886	93.940	96.189	116.817	154.279	1.000	1.000	1.000	1.000	1.000	0.993	0.917	0.658	0.123	0.028
549	4	32	302.686	301.349	311.886	251.948	96.580	91.532	107.868	125.996	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.998	0.781	0.195

Table 3 – Content Detection Results

5.4 Content Identification with Transformation

The goal of this experiment was to determine if the system can identify correctly the origin of a shot, when the queries are transformed videos. These are more likely the scenarios in the real world. The performance was evaluated with (CIVR2007) database.

5.4.1 (CIVR2007) Competition

The ACM International Conference on Image and Video Retrieval (CIVR) series of conferences was originally set up to illuminate the state of the art in image and video retrieval between researchers and practitioners throughout the world. This conference aims to provide an international forum for the discussion of challenges in the fields of image and video retrieval.

The video identification solution proposed in this thesis is going to participate in the next CIVR, so it would be a very good way to evaluate this system, comparing the result from (CIVR 2007). There was a competition organized during the ACM International Conference on Image and Video Retrieval (CIVR2007) and was supported by the network of excellence MUSCLE.

This evaluation Showcase for video Copy Detection was one of the three live evaluation events which took place at the ACM CIVR 2007

The competition covered the following scenarios.

- Transformed full-length movies with no post production and a possible decrease of quality (camcording i.e.);
- Short segments on TV streams with possibly extensive large post-production transformations.
- Short videos on the Internet with various transformations (may be extracted from a TV stream);
- Therefore, the video queries can be singles videos (videos available on the internet for examples) or video stream (Web TV and TV).

5.4.2 (CIVR2007) Competition Participants

The teams which have taken part in the competition are:

- IBM T.J. Watson Research Center , USA
- ADVESTIGO , France
- City University of Hong Kong , Hong Kong
- Institute of Computing Technology, Chinese Academy of Sciences , China
- Bilkent University RETINA Group , Turkey

Others groups were interested in this competition and are working with this. For various reasons, they have not participated in the competition.

- LTU , France
- Columbia University, USA
- University of Queensland, Australia
- Thomson, France
- NII, Japan
- INRIA Lear, France

5.4.3 (CIVR2007) Competition Database

Main database:

- About 100 hours of video materials coming from different sources: web video clips, TV archives, movies.
- The videos cover very large kind of programs: documentaries, movies, sports events, TV shows, cartoons etc.
- The videos have different bitrates, different resolutions and different video format.
- These videos have been provided in their original format and also in an MPEG1 format by a re-encoding.

5.4.4 (CIVR2007) Competition Task

Copy of whole long videos, the videos has length from 5 minutes to 1 hour. The data can be re-encoded, noised, or slightly retouched. The most difficult queries could be movies re-acquired by a camcorder.

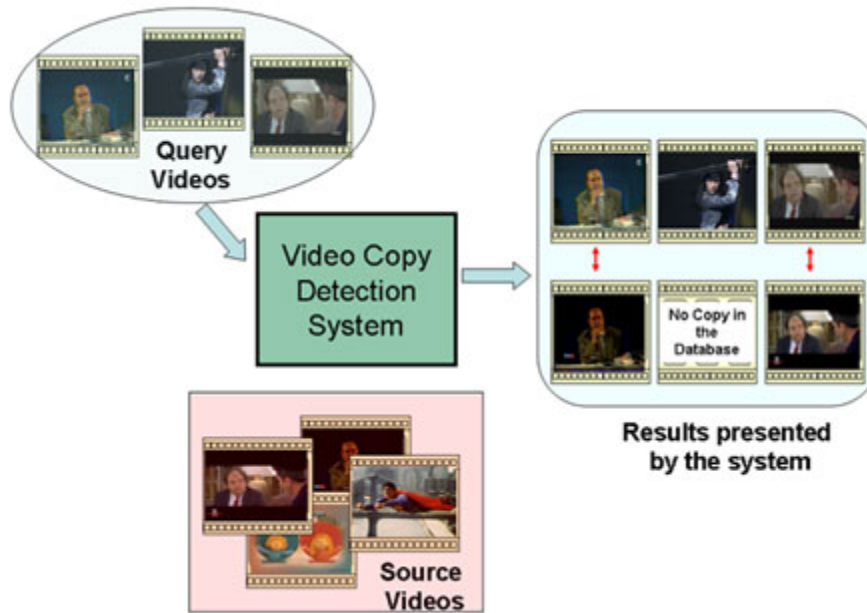


Figure 30 – (CIVR 2007) Video Stream Query

A set of video is used as queries and each query returned an answer: the file is a copy of a video (or of a part of a video) in the database or the file was not a copy. Quality is measured as:

$$Quality = N_{correct} / N_{queries}$$

Queries for task:

- 15 videos with transformations
- total length of queries: 2 hours 30 minutes

The queries describe the kind of situations that a video identification system has to face to. Table 4 shows the description for the 15 query videos, Figure 31, Figure 32 and Figure 33 show some videos and its transformations screenshots.

Query number	Origin	Transformation
ST1Query1	movie27	color adjustment + Blur
ST1Query2	not_in_db	
ST1Query3	movie8	reencoding + color adjustment + cropping
ST1Query4	not_in_db	
ST1Query5	movie44	reencoding with strong compression
ST1Query6	movie76	frontal camcording + subtitles
ST1Query7	not_in_db	
ST1Query8	not_in_db	
ST1Query9	movie9	colors phase modification + color adjustement
ST1Query10	movie21	non frontal camcording
ST1Query11	movie37	frontal camcording
ST1Query12	not_in_db	
ST1Query13	movie11	flip
ST1Query14	movie17	resizing + subtitles
ST1Query15	movie68	resizing (longest video)

Table 4 – (CIVR 2007) query videos description



Figure 31 – (CIVR 2007) query 1



Figure 32 – (CIVR 2007) query 10



Figure 33 – (CIVR 2007) query 14

5.4.5 Performance of the Proposed Solution

This section explains how the video identification system was tested with the (CIVR2007) content. The Video Signature Process was applied to the complete database before any experiment. The shot length was 64 frames. The video signature process was applied to the query video, now signatures are ready to be compared. The comparison is based on distance, Euclidean distance.

To explain the process let's assume that a query video has 3 minutes length and is 25 frames per second, the number of frames in the video is 13500 frames, dividing by 64 frames per shot, we have 210 shots, in other words, we have 210 signatures for this query.

The distance between these 210 signatures and the first video in the database is calculated. A vector of 210 distances is created, the mean value of the distance vector is the distance between the query and the first video, now we have a number which represents the distance between two complete videos. We can call this number the "Video Distance".

The process is repeated for all the videos in the database, the minimum "Video distance" indicates which the closest video is to the query. "Video distance" has to be a valid distance, this is video and query must have about the same number of frames

If the minimum "Video Distance" is greater than a threshold, then the query video does not match any video in the database, the query does not exist in the database.

Table 5 shows the result for the 15 queries videos, the first column indicates query number, second column where the query comes from, the third column correspond to the transformation used. Minimum valid distance indicates the numeric value of the minimum true all the database, origin column is where the minimum value comes from, and the final column says is there is or not a match.

Query number	Origin	Transformation	minimun valid distance	Origin	Match
ST1Query1	movie27	color adjustment + Blur	449.26	27	1
ST1Query2	not_in_db		empty	empty	1
ST1Query3	movie8	reencoding + color adjustment + cropping	224.83	8	1
ST1Query4	not_in_db		empty	empty	1
ST1Query5	movie44	reencoding with strong compression	1614	44	1
ST1Query6	movie76	frontal camcording + subtitles	1372	76	1
ST1Query7	not_in_db		empty	empty	1
ST1Query8	not_in_db		798	31	0
ST1Query9	movie9	colors phase modification + color ajustement	139	9	1
ST1Query10	movie21	non frontal camcording	1717	21	1
ST1Query11	movie37	frontal camcording	1795	4	1
ST1Query12	not_in_db		empty	empty	1
ST1Query13	movie11	flip	226	11	1
ST1Query14	movie17	resizing + subtitles	1810	17	1
ST1Query15	movie68	resizing (longest video)	1003	68	1

Table 5 – Results on CIVR content

For the 15 queries only one video was misclassified because the minimum value was located under the threshold, the query was read like a match to video 31. Table 6 exposes the performance again the other competitors, this system has a precision of $14/15 = 0.93$, time spent to calculate the signatures and make the comparisons is shown in Table 6.

Team - run	ST1 score	ST1 search time
Advestigo	0,86	64 min
Bilkent	n/a	n/a
Chinese academy of sciences - 1	0.46	41 min
Chinese academy of sciences - 2	0.53	14 min
City university of Hong Kong	0.66	45 min
IBM - 1	0.86	44 min
IBM - 2	0.73	68 min
IBM - 3	0.8	99 min
Content Identification Using Video Tomography	0.93	38 min

Table 6 – Performance Evaluation compared to other CIVR competitors

The video identification system based on video tomography has impressive results, the precision is better than anyone in this competition. Since I do not know the identification processes used by other competitors, I am not able to make comparison in that point, but the system proposed in this work is by far a smart solution to the video identification problem.

5.5 Signature Generation Complexity

Generating the signatures for a video clip has relatively low complexity. The complexity is dominated by the complexity of edge detection in tomography images. On a 2.4 GHz Intel Core 2 PC it takes about 100 milliseconds to generate a video signature for a 180 frame video clip with 720x480 resolution. At 30 frames per second, the complexity of signature generation is negligible and can be implemented in standard video player without sacrificing playback performance. Using MATLAB video processing toolbox, in the same PC above, more than 100 frames per second can be processed.

Chapter 6 CONCLUSIONS AND FUTURE WORK

6.1 Introduction

In this chapter the conclusion and the future work are drawn

6.2 Conclusions

This thesis presents a novel, low complexity method for video identification. The proposed video identification system is capable of:

- Creating digital Signature based on video tomography for every shot of video.
- Compare, using a simple metrics, two videos and decide if they have the same content.
- Identify two videos with the same content even when non trivial transformations are present.
- Reducing the size of a shot (64 frames) to 32 bytes, for identification purposes.

On the other hand, the system is subject to the following restrictions:

- A standalone application is not feasible with current implementation in MATLAB.
- The system has not been tested on frame rate changes scenarios.

Through study and experimentation, this work has reached the following conclusions:

1. A shot boundary detection scheme based on video tomography can be embedded into the signature generation engine to create video signatures for video shots, or can be an independent application.
2. The results show that the proposed system has a recall of 100% and a precision over 93%, even 97% if the video has no transformation.
3. The experiments conducted give a good confidence on the performance of the system. Since the signatures are evaluated exhaustively – each signature in the database is compared against all other signatures – the high recall and precision show that the signatures designed are able to uniquely identify video clips.
4. The proposed system has low complexity for both signature generation and matching it only needs 64 bytes to represent a shot. The system can process over 100 frames per second.
5. Since the video signatures used are derived from spatio-temporal characteristics that are robust to compression artifacts the proposed solution can survive recompression and transcoding.
6. The proposed video identification system proved to be independent to video compression algorithms, video resize, color domain changes, video cropping, video flip and even the addition of subtitles.

7. MATLAB proved to be an important tool when developing prototypes due to its built-in video processing and mathematical tools. For standalone implementation the use of a player in C++ is required.

6.3 Future Work

Possible avenues for future work related to this thesis include:

1. Integration with a video player is necessary in order to create a standalone application
2. Combine the system with another video features could raise overall identification performance.
3. To avoid the possible frame rate dependence, a resample process can be implemented.

BIBLIOGRAPHY

- [1] G. Doerr and J.L. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Communication*, Volume 18, Issue 4, April 2003, Pages 263-282.
- [2] T.T. Ng, S.F. Chang, C.Y. Lin, and Q. Sun, "Passive-blind Image Forensics," in *Multimedia Security Technologies for Digital Rights*, Elsevier (2006).
- [3] W. Luo, Z. Qu, F. Pan, J. Huang, "A survey of passive technology for digital image forensics," *Frontiers of Computer Science in China*, Volume 1, Issue 2, May 2007, pp. 166 – 179.
- [4] T. Gloe, M.Kirchner, A.Winkler, and R. Böhme, "Can we trust digital image forensics?," *Proceedings of the 15th international Conference on Multimedia*, Multimedia '07, pp. 78-86.
- [5] X. Fang, Q. Sun, and Q. Tian, "Content-based video identification: a survey," *Proceedings of the Information Technology: Research and Education*, 2003. ITRE2003. pp. 50-54.
- [6] 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 *Proceedings of the 6th ACM international Conference on Image and Video Retrieval*, CIVR '07, pp. 371-378.
- [7] T. Can. and P. Duygulu. "Searching for repeated video sequences," *Proceedings of the international Workshop on Multimedia information Retrieval*, MIR '07, pp. 207-216.
- [8] Y. Yan, B.C.Ooi, and A. Zhou, "Continuous Content-Based Copy Detection over Streaming Videos," *24th IEEE International Conference on Data Engineering (ICDE)* 2008
- [9] C.Y. Chiu, C.C. Yang. and C.S. Chen. "Efficient and Effective Video Copy Detection Based on Spatiotemporal Analysis," *Ninth IEEE International Symposium on Multimedia*, 2007, pp.202-209.
- [10] N. Guil, J.M. Gonzalez-Linares, J.R. Cozar, and E.L. Zapata, "A Clustering Technique for Video Copy Detection," *Pattern Recognition and Image Analysis*, LNCS, Vol. 4477/2007, pp. 451-458.

- [11] G. Singh, M. Puri, J. Lubin, and H. Sawhney, "Content- Based Matching of Videos Using Local Spatio-temporal Fingerprints," Computer Vision – ACCV 2007, LNCS vol. 4844/2007, Nov. 2007, pp. 414-423.
- [12] A. Akutsu and Y. Tonomura, "Video tomography: An efficient method for camera work extraction and motion analysis," Proceedings of the 2nd international Conference on Multimedia, ACM Multimedia 94, 1994, pp. 349-356.
- [13] A. Yoshitaka and Y. Deguchi. "Video Summarization based on Film Grammar" Proceedings of the IEEE 7th Workshop on Multimedia Signal Processing, Oct. 2005, pp.1-4.
- [14] C. W. Ngo et. al., "Video Partitioning by Temporal Slice Coherency", IEEE Trans. CSVT, 11(8):941-953, Aug, 2001.
- [15] C. W. Ngo, Ting-Chuen Pong, HongJiang Zhang, "Motion-Based Video Representation for Scene Change Detection," International Journal of Computer Vision 50(2): 127-142 (2002)
- [16] Chong-Wah Ngo, Ting-Chuen Pong, HongJiang Zhang, "Motion Analysis and Segmentation Through Spatiotemporal Slices Processing", IEEE Transactions on Image Processing, Vol. 12, No. 3. 341-355.
- [17] J.F. Canny, "A Computational Approach to Edge Detection", IEEE Trans. Pattern Analysis and Machine Intelligence, vol. 8, pp. 679-698, 1986.
- [18] R.M. Bolle, A.W. Senior, N.K. Ratha, and S.Pankanti, "Fingerprint minutiae: A constructive definition," Lecture Notes in Computer Science, Vol. 2359/2002, pp. 58–66.
- [19] J. Mas and G. Fernandez, "Video Shot Boundary Detection Based on Color Histogram", TrecVid 2003.
- [20] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W., "Automatic Partitioning of Full-motion Video", Multimedia Systems (1993) Vol. 1, No. 1, pp. 10-28.
- [21] Shahraray, B., "Scene Change Detection and Content-Based Sampling of Video Sequences", in Digital Video Compression: Algorithms and Technologies, Arturo Rodriguez, Robert Safranek, Edward Delp, Editors, Proc. SPIE 2419, February, 1995, pp. 2-13.
- [22] Kasturi, R. and Jain R., "Dynamic Vision", in Computer Vision: Principles, Kasturi R., Jain R., Editors, IEEE Computer Society Press, Washington, 1991.
- [23] Arman, F., Hsu, A., and Chiu, M-Y., "Image Processing on Encoded Video Sequences", Multimedia Systems (1994) Vol. 1, No. 5, pp. 211-219.

- [24] Little, T.D.C, Ahanger, G., Folz, R.J., Gibbon, J.F., Reeve, F.W., Schelleng, D.H., and Venkatesh, D., "A Digital On-Demand Video Service Supporting Content-Based Queries", Proc. ACM Multimedia 93, Anaheim, CA, August, 1993, pp. 427-436.
- [25] Zabih, R., Miller, J., and Mai, K., "A Feature-Based Algorithm for Detecting and Classifying Scene Breaks", Proc. ACM Multimedia 95, San Francisco, CA, November, 1993, pp. 189-200.
- [26] John S. Boreczky and Lawrence A. Rowe, "Comparison of Video Shot Boundary Detection Techniques", Storage and Retrieval for Image and Video Databases ({SPIE})", 1996, pp. 170-179.
- [27] TRECVID 2007, <http://www-nlpir.nist.gov/projects/trecvid/Results>,
<http://www-nlpir.nist.gov/projects/tv2007/active/results/shot.boundaries/runTable.full>
- [28] <http://msdn.microsoft.com/en-us/vstudio/default.aspx>
- [29] <http://www.mathworks.com/>
- [30] <http://www.intel.com/cd/software/products/asmo-na/eng/compilers/284132.htm>
- [31] <http://www.intel.com/cd/software/products/asmo-na/eng/302910.htm>