



THESIS APPROVAL

GRADUATE SCHOOL, KASETSART UNIVERSITY

Doctor of Engineering (Computer Engineering)

DEGREE

Computer Engineering

FIELD

Computer Engineering

DEPARTMENT

TITLE: Image/Video Cropping and Indexing by Cinematography Knowledge

NAME: Mr. Paween Khoenkaw

THIS THESIS HAS BEEN ACCEPTED BY

THESIS ADVISOR

(Associate Professor Punpiti Piamsa-nga, D.Sc.)

THESIS CO-ADVISOR

(Assistant Professor Chaiporn Jaikaeo, Ph.D.)

DEPARTMENT HEAD

(Associate Professor Anan Phonphoem, Ph.D.)

APPROVED BY THE GRADUATE SCHOOL ON _____

DEAN

(Associate Professor Gunjana Theeragool, D.Agr.)

THESIS

IMAGE/VIDEO CROPPING AND INDEXING BY
CINEMATOGRAPHY KNOWLEDGE

PAWEEN KHOENKAW

A Thesis Submitted in Partial Fulfillment of
the Requirements for the Degree of
Doctor of Engineering (Computer Engineering)
Graduate School, Kasetsart University
2014

Paween Khoenkaw 2014: Image/Video Cropping and Indexing by Cinematography Knowledge. Doctor of Engineering (Computer Engineering), Major Field: Computer Engineering, Department of Computer Engineering. Thesis Advisor: Associate Professor Punpiti Piamsa-nga, D.Sc. 135 pages.

Because of high-bandwidth availability of networks, such as 3G and Wi-Fi, is essential for smart phones and computer tablets, applications of video broadcasting onto those devices become more important. Usually, sizes and aspect ratios of both contents and display screens are drastically unmatched. Scaling the contents to fit onto the display screen is not an appropriate solution because scaled objects are too small to see and it also produces inefficient “black bar” areas on display devices. Cropping only center area of the frame could lose some important information if it is located at far left or far right of the frame. Manual cropping is also very exhaustive. This paper presents a fully automatic algorithm for selecting the most appropriate areas of image and video show on heterogeneous displays.

We proposed to analyze cinematic features in video and use it for identifying important parts of each video frame or image. This information was used to crop the areas to fit its display screen. The well-known motion pictures and image dataset are used for performance evaluation. The experimental results clearly showed that audiences preferred results processed by our algorithm.

Cinematic features and methods for image and video cropping also used as a foundation for video indexing algorithm. The N-gram model is proposed to extract shot transition and used as a signature. This indexing algorithm yields high precision and fast detection time but low on recall. However, we also proposed an improved algorithm based on spectrogram. This algorithm gets high precision rate over all recall but requires more computation time.

Student's signature

Thesis Advisor's signature

____ / ____ / ____

ACKNOWLEDGEMENT

I am most grateful to my advisor, Associate Professor Dr. Punpiti Piamsa-nga, for providing guidance and encouragement that has enabled this work to emerge and complete. He introduced me to the field of multimedia content analysis, advised me knowledge in research methodology, and worked hard in reviewing on my journal article. Also, I would like to present deeply thank to my research committees, Assistant Professor Dr. Chaiporn Jaikaeo and late Assistant Professor Dr. Thitiwan Srinark for their constructive and valuable advices.

I would like to present deeply thanks to my friends of the MAD Laboratory, Phas Pramokchon, Jitdumrong Preechasuk, Ouychai Intharasombut, Suttasinee Chimlek and Suthikarn Bojkrapan, for their high spirit and friendship.

Finally, most of all, my appreciations devote to my family for their love, understanding, warmth and encouragement to accomplish this research.

Paween Khoenkaw

July 2014

TABLE OF CONTENTS

	Page
TABLE OF CONTENTS	i
LIST OF TABLES	ii
LIST OF FIGURES	iii
INTRODUCTION	1
OBJECTIVES	9
LITERATURE REVIEW	10
MATERIALS AND METHODS	23
Materials	23
Methods	24
RESULTS AND DISCUSSION	60
Results	81
Discussion	103
CONCLUSION AND RECOMMENDATION	105
Conclusion	105
Recommendation	108
LITERATURE CITED	109
APPENDIX	118
CIRRICULUM VITAE	135

LIST OF TABLES

Table		Page
1	Comparison of Aspect Ratio Adjustment Method And Display Size	7
2	Filters Used In Various Auto Focusing Techniques	15
3	Comparison of Video Player Properties	17
4	Cinematic Features And Interpretation	25
5	Result In Tilt Mode Using Difference Zoom Parameter (Z)	
	Comparison Between Crop and Uncrop	50
6	Result In Pan Mode Using Difference Zoom Parameter (Z)	
	Compared With Un-Crop Version	51
7	List Of Displays Used For Algorithm Evaluation	66
8	Detail Of Video Coding Used In This Experiment	70
9	Queries To Build Video Collections	75
10	Image Cropping Evaluation Result	81
11	User Satisfaction Score Classified By Background In Photography.	84
12	Descriptive Statistic Of Different Ages	85
13	Anova Result Of Age Factor	86
14	Statistical Details Of Satisfaction Scores Of Each Video	
	Cropping Method	88
15	Cropping Precisions Of Our Algorithm On Selected Motion Pictures	90
16	Cropping Accept Rate For Undesirable Cases	93
17	Video Indexing Based On N-Gram Average Results	96
18	Average And Standard Deviation Of Precision of Indexing Result.	98
19	Cinematic Used In Each Part Of This Thesis	107

LIST OF FIGURES

Figure		Page
1	Aspect Ratios Commonly Used In Motion Pictures From Nearly Square To Anamorphic Widescreen.	2
2	“Shoot And Protect” Framing Method, Important Objects Must BeCluster In The Center Of Image	3
3	Pan And Scan Process	4
4	Tilt And Scan Process	5
5	Examples Of Aspect Ratio Adaption Results	6
6	Examples Of Aspect Ratio Adaption Results	6
7	Focusing Mechanism In Camera	13
8	General Model Of Camera Auto Focusing System	13
9	Mobile Video Broadcaset Framework	17
10	An Example of Rack-focus shot	20
11	An Example Of Pan Shot	21
12	An Example Of Tracking Shot	22
13	Model of our proposed image and video cropper system	24
14	Examples Of Three Different Cinematic Types	26
15	Observation On Effect Of Motion Blur In The Same Lens Configuration	27
16	Decision Chart For Important Object Identifier Based On Characteristics Of Motion	29
17	Proposed Automatic Image Cropping Model	30
18	Importance Maps Of Photograph At Different Focal Point.	32
19	Example Of Cropping Result From Different Maps	33
20	Result From This Algorithm In Pan-And-Scan Mode	33
21	An Example Result Of Automatic Zoom Algorithm	36
22	Block Diagram Of Importance Map Generator In Server-Side Module.	36
23	Importance Map Generated By Different Focus Position	39

LIST OF FIGURES (Continued)

Figure		Page
24	Class 2 Shot Type Where $k \leq \frac{bin}{2}$	41
25	Class 2 Shot Type Where $k > \frac{bin}{2}$	42
26	Class 3 Shot Type (Everything Is Equally Important)	42
27	Block Diagram Of Video Cropper At Client-Side Module.	43
28	Example Result Of Each Iteration Of An Automatic Zoom Algorithm	45
29	An Example Of Cropper Result	46
30	Example Of Video Cropper Output	47
31	Example Of Video Cropper Output	48
32	Compare Algorithm Result Of Each Aspect Ratio	49
33	Proposed Model Of Video Copy Detection Based On N-Gram.	52
34	Proposed Model For Video Indexing Based On Spectrogram	54
35	The Ordinal Feature Extraction Process	55
36	A Dimension Of Spectrogram Signature Example	56
37	The Minimum Cost Path Of Different Type Of Video Comparison.	59
38	Best Fitness And Mean Fitness Score (Low Is Good) Of Each Generation While Solving One Zoom Iteration.	61
39	The Interface Of Google Picasa. 3 Cropped Versions (A, B, And C) And Original Image (D)	62
40	User Interface Of Questionnaire Website For Qualitative Evaluations.	63
41	Example Of Video Cropping Result Of Scenario Source Aspect Ratio > Target Aspect Ratio	64
42	Landscape Mode: (A) Letterbox Method; (B) Cropped Method, Portrait Mode: (C) Letterbox Method; (D) Cropped Method.	65
43	The Illustrated Of Display Aspect Ratio Used In This Experiments	66
44	User Interface Of The Algorithm Evaluation	68
45	Ground Truth Generation From Vcd	69

LIST OF FIGURES (Continued)

Figure		Page
46	Pan Offset Used To Generated Vcd Version From Dvd	70
47	Ordinal Feature Extraction Process For Signature Generation	71
48	Video Frame Before De-Interlace (A) And After De-Interlace (B)	71
49	Different Frame Sequencing Shot Found In Dvd And Dvd	72
50	Pan-And-Scan Offset Signal Before Filter (A) And After Filtered (B).	72
51	Example Of The Exact Same Frame Of 2.35:1 And 4:3 Release Of Fake-Widescreen Movie	73
52	Film Areas Used By Face-Widescreen Format	74
53	Example Of The Exact Same Frame Of 2.35:1 And 4:3 Release Of True Widescreen Movie	74
54	Accuracy, Precision, Recall And F With Different Number Of Quantization Levels (L)	76
55	Accuracy, precision, recall and F at different number of n-gram in query (k)	77
56	Query Time At Different Number Of N-Gram Over All Queries.	77
57	The Accuracy At Different Grid Size	78
58	Precision At Different Stft Window Size	79
59	Overlap Window	80
60	Data Loss Distribution Of Four Cropping Algorithms.	82
61	Relationship Between User Satisfaction Scores And Percent Data Loss Of Four Cropping Algorithms.	83
62	Results Group By Video Clip	86
63	The Mean Result Of Each Image Loss Produced By Aspect Ratio Different	86
64	The Mean Result Of All Method Over All Image Loss	87
65	Results By User Group	87
66	Example Of Scenario That There Is More Than One Correct Position	89
67	Comparison Results In Different Precisions	91

LIST OF FIGURES (Continued)

Figure		Page
68	Example Of Webpage For Evaluation Between Result From Expert And Result From Our Algorithm	92
69	Example Of Undesirable Cases	94
70	Precision, Recall And F-Measure Compared With LSH-E And VK.	96
71	Ranking Performance Of Sliding Window Algorithm	99
72	Ranking Performance Of Std Algorithm	99
73	Ranking Performance Of Dtw Algorithm	100
74	Ranking Performance Of Dtw+ Algorithm	100
75	Ranking Performance Of Proposed Algorithm	101
76	Radar Plot Of All Algorithms When Recall=1	101
77	Precision Compare Between Video Groups At Recall =1	102

IMAGE/VIDEO CROPPING AND INDEXING BY CINEMATOGRAPHY KNOWLEDGE

INTRODUCTION

When viewing high-resolution images on a small display device, the image must be scaled down to match the device's aspect ratio and resolution; as a result, important contents are sometimes too small to see. Image cropping is an alternative for selecting important visual information to show on an arbitrary aspect-ratio display. Photo cropping is widely used in photo album software, such as Google Picasa 3 (Google, 2013b) and Cropp.me (Imagga, 2013); it can suggest choices of cropped results by different algorithms to the users in order to ask them for their favors on a specific display or printer. However, there are many devices whose display areas have varieties of sizes, resolutions, and aspect ratios. A single version of cropped photographs cannot optimally be appropriate for every type of output device. Multiple versions of manual cropping on a single photograph is also a very tedious task since the number of photographs owned by a person is too high to process each for many types of displays. Rather than just being a recommender for cropping, photograph cropping algorithm is required to be automatic.

Displays of mobile devices usually have different shapes and different sizes. Its shape varies from square to widescreen and its size varies from wristwatch to tablet. However, not only display device that comes with variety of shapes but video source also comes with many shapes varying from nearly square PAL/NTSC shape to widescreen Blu-Ray format. In order to fit the source video content onto target display, backgrounds of video and display, video aspect ratio adaptation methods are introduced as follows.

The nearly-square picture frame was introduced in the early stage of motion picture (Wheeler, 1969). Then, the picture frame dimension becomes wider and

many formats were introduced (Katz, 1991). A parameter to represent shape of picture frame is called *aspect ratio*, which is the ratio between width and height of video frame or display screen. There are many aspect ratios used in different video formats, such as 2.35:1 (Anamorphic), 1.85:1 (Widescreen), 16:9 (Letterbox), 4:3 (Conventional television) (Wright, 2006). The shapes of picture frames are illustrated in Figure 1. Generally, it is easy to cut a screen in a theater to match any aspect ratio of movie. However, aspect ratio of television is fixed. Therefore, movies with heterogeneous aspect ratios had to be modified to be fit onto 4:3 television.



Figure 1. Aspect ratios commonly used in motion pictures from nearly square to anamorphic widescreen.

Video aspect ratio adaption is a method for transforming video from one aspect ratio to another. There are three major methods to deliver the most appropriate contents on to targeted screen: letterboxing, squeezing and cropping or zooming. Letterboxing is one of the most common methods in 4:3 television (Watkinson, 2001). This method produces the movie that preserves all visible contents. It is done by resizing each video frame linearly to fit the screen width. However, there will be unused “black bar” areas in both top and bottom of screen (Figure5a) if aspect ratio of the movie is wider than of the targeted screen, i.e. a 16:9 movie is mapped onto 4:3 TV screen. On the other hand, if the aspect ratio of movie is narrower, the black bar shows on the left and right of the screen.

Letterboxing is simple. There is no distortion in image frames. However, screen is not efficiently utilized. It is not appropriate for small screen such as mobile phone since the display might show mainly the background of the frame and the important part of the frame is scaled until it is unable to see. The more aspect-ratio difference between display and movie causes the less efficiency of display screen.

Squeezing is another way of aspect ratio adaption; it preserves all visible areas by scaling original video frames to corresponding targeted screen. Therefore, the screen area is fully utilized. However, result of this method has a problem of image distortion (Figure5b). The more different aspect ratio is used, the more distorted the movie is.

Generally, when a cameraman shoots a film, he/she concerns only on how objects are presented on the aspect ratio of camera in his/her hand. No one can concentrate concurrently on multiple aspect-ratio screens. Therefore, a method to help cameraman to shoot more than one aspect ratio at a time is introduced. It is called “fake widescreen composition” (Prince, 2004). An example of fake composition is “Shoot and Protect” method (Dalton, 1996). During the shooting, there are many aspect-ratio guidelines to help cameraman such as in Figure 2. Usually, guideline of 4:3 aspect ratio is located in the middle and that area is called “safe area”. Shooting in the safe area ensures that important objects will not be eliminated by center cropping.

Using fake composition, film can be cropped into any aspect ratio by fixing a cropping window at the center. However, this method does not comply with the photographic composition rules (Byers *et al.*, 2003). This composition can ruin the aesthetics of its picture.



Figure 2 “Shoot and protect” framing method, important objects must be cluster in the center of image

When video content is played onto a device with tiny screens, some unimportant areas are usually discarded and most of the important content could be cropped to display on the available screen area (Evans-Pughe, 2005; Ming-Sui *et al.*, 2007). Cropping method is done by using aspect ratio of display screen as a window template to crop onto original video frames; and then scaling the cropped image frames to fit the target display. Usually, cropping is done at the fixed central area of video frame (Figure 5c).

“Pan and Scan” is a special cropping method that “cropping window” is manually moved along the original video frame to capture the most important action (Figure 3). A window associated with a new aspect ratio is created, and then used it to pan across source frames horizontally in order to select the most appropriate contents. If the target aspect ratio is greater than source aspect ratio, the window is scan vertically in process call tilt and scan (Figure 4). The method fully utilizes screen area. Picture obtained from the method is not distorted. However, some unimportant parts of frame will be lost. The comparison of every aspect ratio adjustment method is shown in Table 1.

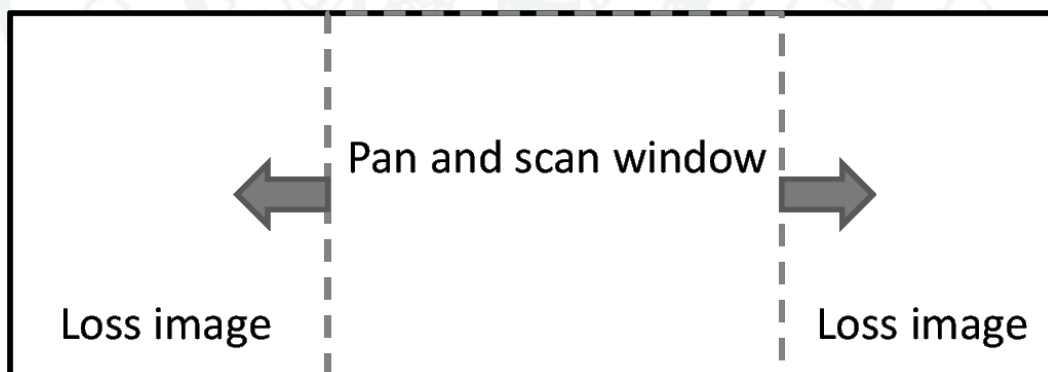


Figure 3 Pan and scan process

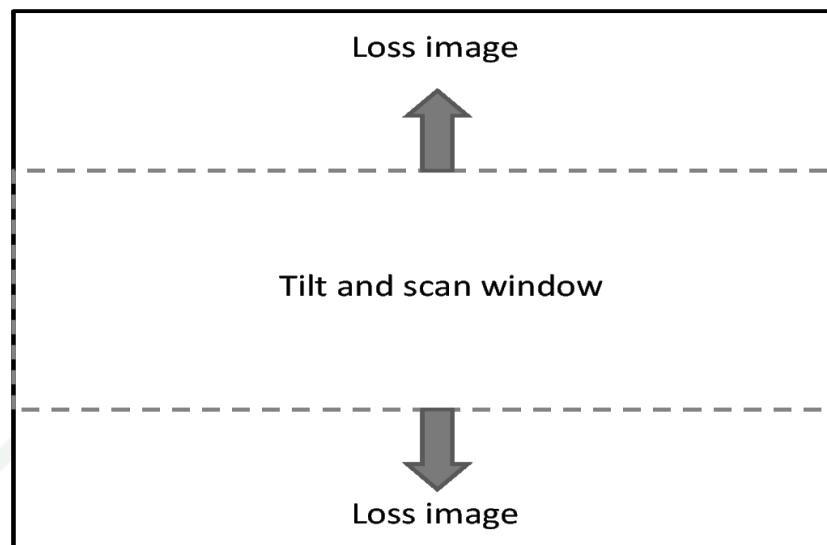


Figure 4 Tilt and scan process

Pan-and-Scan process has advantages over letterboxing and squeezing since it delivers the most appropriate content to the audiences with negligible loss and no geometry distortion (Figure 5d). This method of aspect ratio adaption is best for mobile video viewing experience that has a small display area. Figure 6 shows the example result of aspect ratio adaption for viewing widescreen video (A) using letterbox method (B) and pan-and-scan method (C), it clearly shows result from letter box is too small and result from pan-and-scan is more comfortable to watch.

However, pan-and-scan is an exhaustive process; an operator must manually identify important parts of frame then mark the scan offset of every frame for a specific aspect ratio. Unfortunately, pan-and-scan can produce only one target version at a time. If there are many targeted aspect ratios, the operator has to repeat his/her work for each aspect ratio.

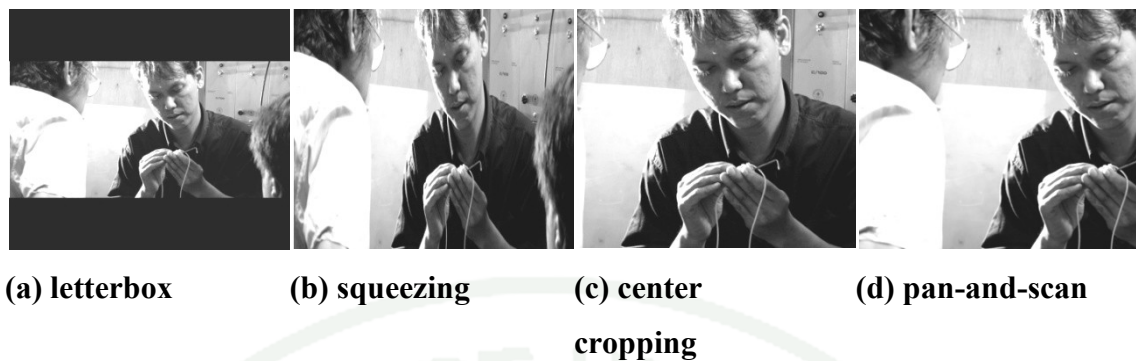


Figure 5 Examples of aspect ratio adaption results



Figure 6 Examples of aspect ratio adaption results

Although we can manually prepare video content in all different formats, receivers have to choose the most appropriate version that matches to their display screens. However, this method can be done only in unicast transmission because each version requires separated channel to transport them. This method brings about transmission of redundant contents and many preprocesses. Even if we can tolerate that, bandwidth utilization is also inefficient (Jordan and Schatz, 2006). This method cannot be used in live programs because video is impossible to prepare.

The proposed solution to this problem is to broadcast original program in single format on single stream and then the clients perform automatic adaption. To broadcast video content on heterogeneous display format, the traditional pan-and-scan method must be tackled with novel automatic cropping methods.

Table 1 Compare aspect ratio adjustment method and display size

	<i>Distortion</i>	<i>Content loss</i>	<i>Automatic</i>	<i>Screen utilized</i>	<i>Applied for</i>
Letterbox	No	No	Yes	No	Large screen
Pan-and- scan	No	Yes	No	Yes	Small screen
Squeeze	Yes	No	Yes	Yes	Small screen

Contributions

The contributions of our research consist of:

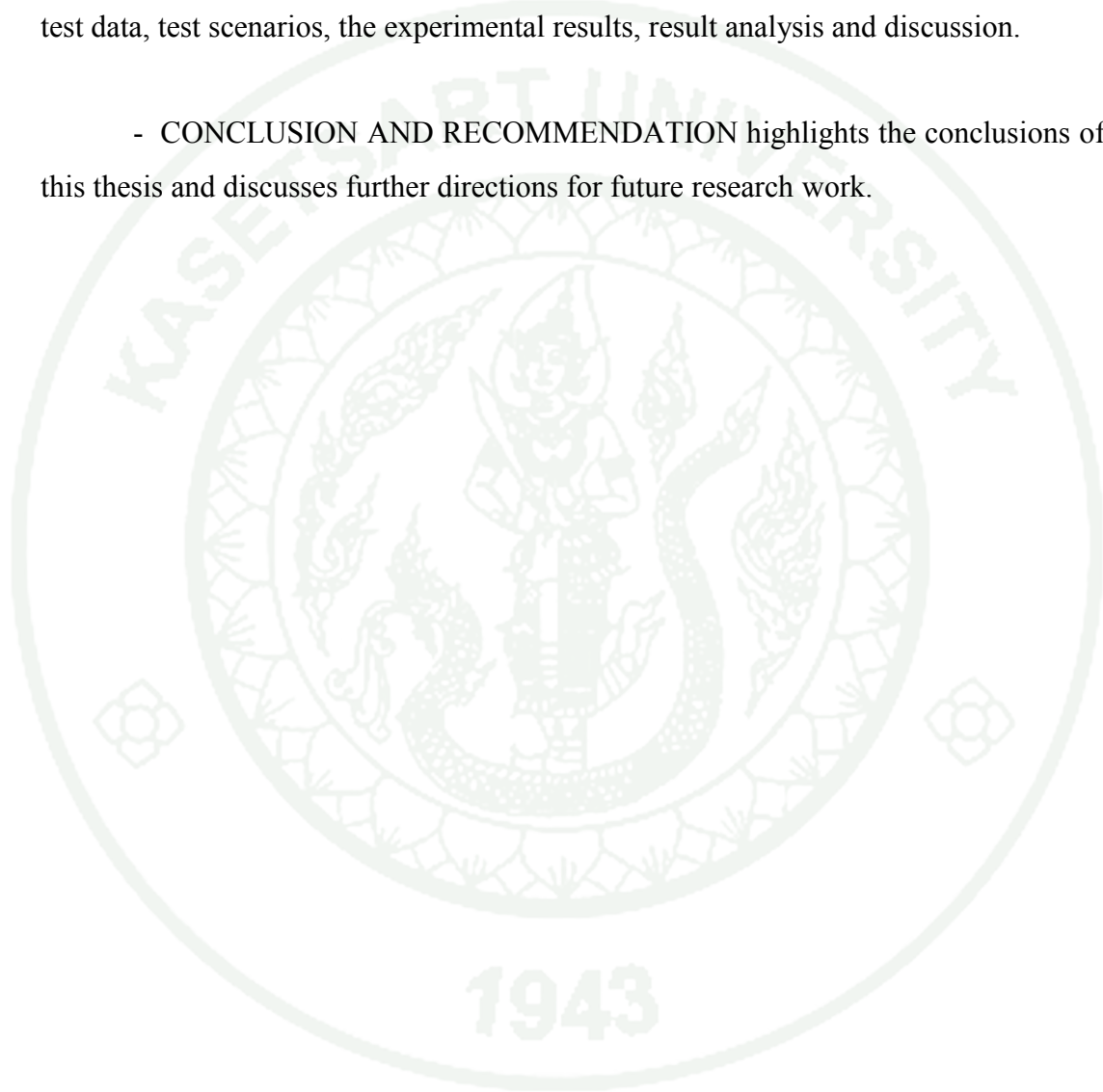
1. An image and video important part of frame identification method using cinematic
2. Map of important objects representation for videos
3. Automatic video cropping algorithm based on cinematic
4. Automatic video indexing algorithm based on cinematic

Thesis Organization

The rest of this thesis is organized as follows:

- OBJECTIVES describe the objectives of the research.
- LITERATURE REVIEW presents and discusses. A summary of previous studies on important part of frame in image and video, image cropping, video cropping and indexing.

- MATERIALS AND METHODS explains about cinematic features and the computational model of cinematic. The image cropping algorithm, video cropping algorithm and video indexing are proposed
- RESULTS AND DISCUSSION describe performance evaluation methods, test data, test scenarios, the experimental results, result analysis and discussion.
- CONCLUSION AND RECOMMENDATION highlights the conclusions of this thesis and discusses further directions for future research work.



OBJECTIVES

1. To find image and video important part of frame identification algorithm based on cinematic
2. To find map of important objects representation method for videos.
3. To find automatic video cropping algorithm using cinematic
4. To find automatic video indexing algorithm using cinematic

LITERATURE REVIEW

This section details about the important part of frame identification method and cropping method for image and video data. The survey of auto focusing algorithm that we were used in part of our proposed method also present in this section.

Automatic image and video cropping method

Letterboxing and Squeezing methods are not appropriate for displays of small devices; however, Pan-and-Scan method has to be done manually for each specific aspect ratio. In order to exhibit video content perfectly on heterogeneously small displays, automatic pan-and-scan cropping algorithm is needed to be investigated.

The automatic cropping model is generally composed of two main parts: “importance part of frame” finder and cropper (Chen *et al.*, 2003; Fan *et al.*, 2003; Grinias and Tziritas, 2002; Jun *et al.*, 2004; Kopf *et al.*, 2006; Liu and Gleicher, 2005, 2006; Liu *et al.*, 2003; Mingju *et al.*, 2005; Santella *et al.*, 2006; Setlur *et al.*, 2005; Suh *et al.*, 2003). The importance-part-of-frame finder plays important role for analyzing content of video and then ranking the important areas in each frame, the information that represents the important area is called *importance map*. The cropper, which is located at the remote client, is to decide what to be kept or ignored by using importance-map to generate a new suitable video version for targeted aspect ratio.

There are three main approaches to generate importance map: psychology model of human vision, feature detection, and combination between both methods. The psychology model is usually based on saliency model, which is about to find the most attractive part of frame by mimicking a psychology principle on how human eyes interact onto photographs (Itti and Koch, 2001). This model commonly uses sum of weighed features to decide which part of frame is important (Frintrop *et al.*, 2010). For feature detection, features, such as Intensity, color and orientation, face

and text, are commonly used to identify important parts of an image frame (Chen *et al.*, 2003; Frintrop *et al.*, 2010; Liu *et al.*, 2003; Mingju *et al.*, 2005).

Some recent research projects about image and video cropping are listed as follows. *Hang et al.* (Mingju *et al.*, 2005) combined saliency map with face detection to create automatic photograph cropper. *Bongwon et al.* (Suh *et al.*, 2003) also used the same principles to create automatic photograph cropper for thumbnail generator. *Liu et al.* (Liu *et al.*, 2003) used saliency map and text detection to create virtual panning across large picture to display it onto small cell phone screen. Moreover, *Fan et al.* (Fan *et al.*, 2003) introduced a model for displaying video on heterogeneous screens using the same saliency model with motion analysis to create automatic video cropping. Chen *et al.* (Chen and Luo, 2011) also introduced techniques to locate saliency in video based on motion. Xue *et al.* (Xue *et al.*, 2011) proposed a model that identifies important area based on face detection and background subtraction method. *Wang et al.* (Jun *et al.*, 2004) used motion analysis in automatic zooming algorithm for displaying video surveillance on tiny mobile screens. *Liu et al.* (Liu and Gleicher, 2006), used combination of image saliency, motion saliency and face detection to generate importance-map. *Kopf et al.* (Kopf *et al.*, 2006) created auto video cropper by generating the importance map using semantics extracted from face, text and motion instead of visual attention model. Kopf *et al.* (Kopf *et al.*, 2011) combined visual attention model, face detection and object classification to detect the most occurred in shot. Cheng *et al.* (Cheng *et al.*, 2007) created importance map by including camera motion as additional features. The low complexity model was also investigated, Hyeong-Min *et al.* (Hyeong-Min *et al.*, 2010) proposed the importance map generator based on discrete cosine transform coefficients and motion vectors in MPEG video.

However, these models rely on saliency model and/or high-level object detection algorithms. The saliency model is based on bottom-up computation visual attention model (Frintrop *et al.*, 2010), which requires high computation to obtain real-time responses (Ma and Zhang, 2003). Face detection is not reliable; errors in facial detection occur by multi-facial poses, presence or absence of components such

as eyeglasses, facial expression, occlusion, image orientation and image condition (Ming-Hsuan *et al.*, 2002).

Automatic focusing algorithm

The camera made of two important elements, lens and optical sensor. When light pass thou lens it will bend to a certain total degree to form a real image at the optical sensor, CCD or chemical based film. Photographer moves the lens to change focal point in order to select what object to be sharp and what else to be blur. Digital auto focus algorithm use the fact that when object are in-focus it will sharpest and it begin to blur when it out-of-focus. The sharp images contain high frequency component than the blur one, Auto focus mechanism use this knowledge to control lens and iris to make high frequency component in a desire area part of frame as high as it possible.

Focusing system is the mechanics that adjusting a lens appropriate to the distance of the subject to have a sharp picture. By changing the lens position the sharpest object can be changed, the general idea of focusing was shown in Figure 7. There are two types of autofocus system on camera: direct and indirect measurement. Direct measurement is using physical sensor like infrared or ultrasonic, the direct measurement measure distance between sensor units and object location as parameters to adjust the position of lens. The indirect focal measurement constructs autofocus without finding the distance between the object and sensor unit. This modern method uses image processing technique to estimate the best lens position to produce sharpest image of target object(Feng and Hong, 2005).

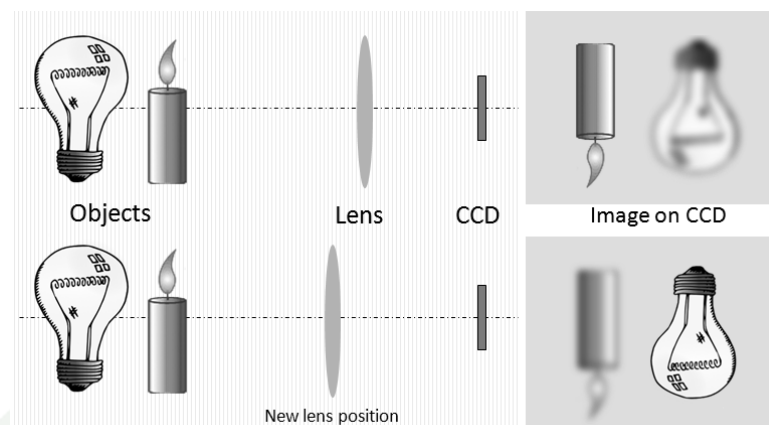


Figure 7 Focusing mechanism in eyes type camera

An autofocus system is composed of four components: lens position control, image capture device, digital filter/transformer, and focus quality measurement. Light passing through the lens forms an image on the capture device, typically a charge-coupled device (CCD). Then, features related to focus quality of the digitized image are extracted. Finally, the focus quality measurement uses these extracted features to assess the focus-quality of the target objects and adjust the lens position. The process is repeated until the highest focus quality is obtained. The auto focus system is depicted in Figure 8

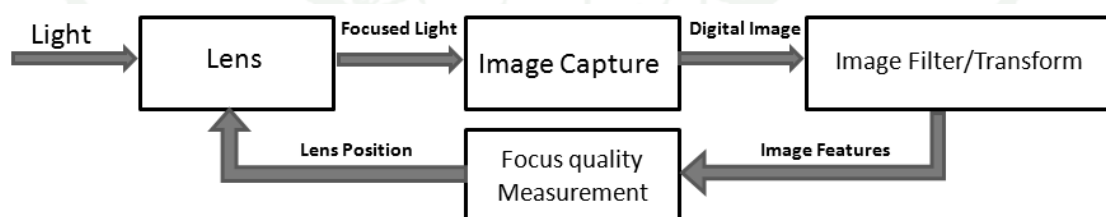


Figure 8 General model of camera auto focusing system

From Table 1, there are some digital auto focus algorithms have been proposed that using difference type of filter or transformation. By closer look, every filter is just the difference way of image frequency component analysis. It is designed on the principle of sharpen image having a high frequency component energy (Cheol-Hee *et al.*, 2000; Chun-Hung and Chen, 2006; Sung-Hyun *et al.*, 2005) than the blurry image. Therefore, it tries to control lens and iris movement and iris in order to maintain the selected part in-focus

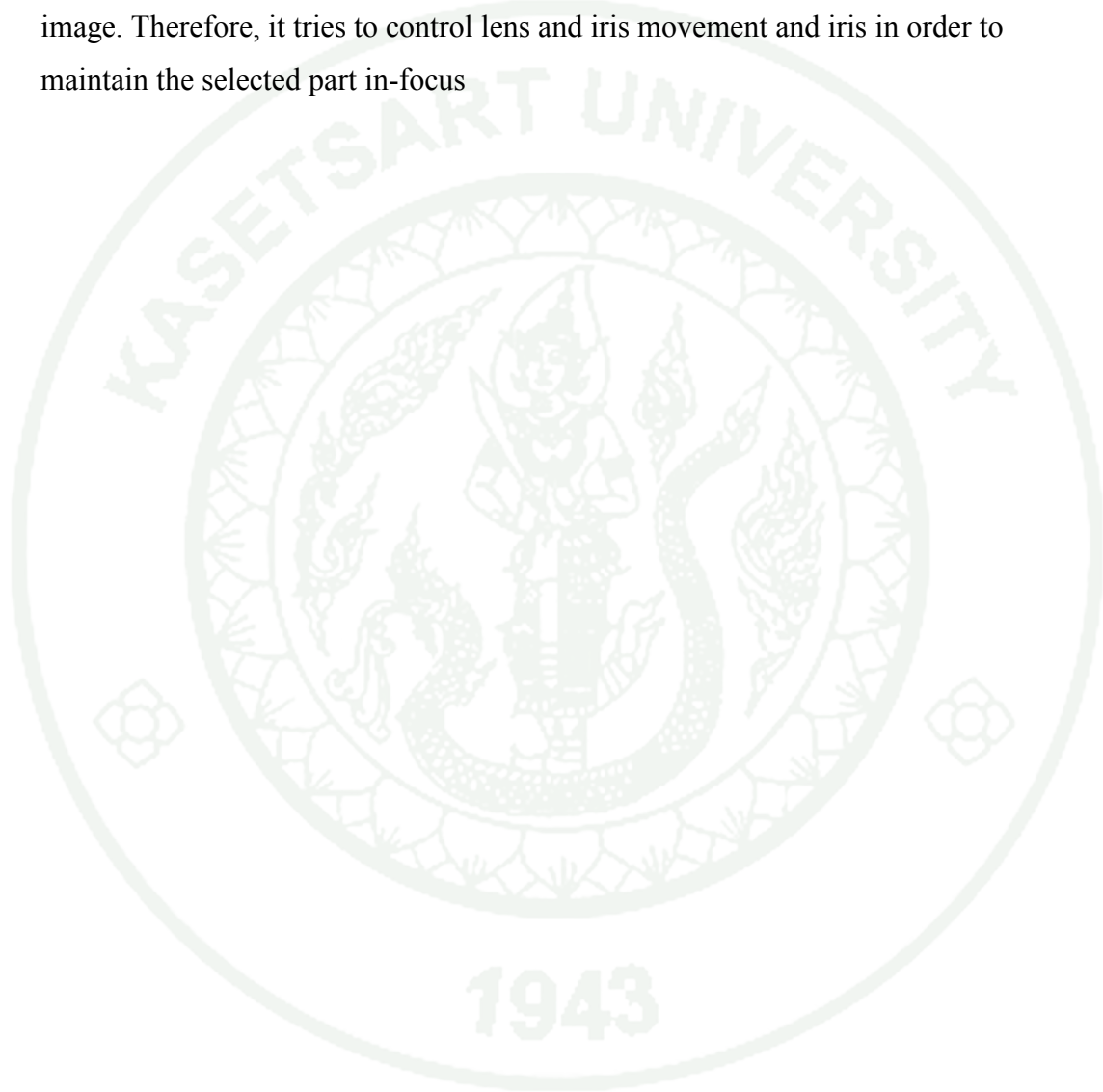


Table 2 Filters used in various auto focusing techniques

<i>Researcher (year)</i>	<i>Filter</i>
M. Subbarao (1993) (Subbarao <i>et al.</i> , 1993)	Laplacian mask
J. Baina (1993) (Baina and Dublet, 1995)	Discrete Cosine Transform (DCT)
Sang ku kim (1998) (Sang Ku <i>et al.</i> , 1998)	Discrete Fourier Transform (DFT)
Chung Nam Cho (1999)(Chung Nam <i>et al.</i> , 1999)	Edge detection
I. kharitononko (2000) (Kharitonenko and Zhang, 2000)	Squared gradient
Cheol-Hee Park (2000) (Cheol-Hee <i>et al.</i> , 2000)	High pass filter (Gaussian shape optical transfer function)
Chee Hwa Chin (2003) (Chee Hwa <i>et al.</i> , 2003)	Wavelet
Jie He (2003) (2003) (Jie <i>et al.</i> , 2003)	Gradient (Laplacian)
KANG ZM (2003)(KANG Z M, 2003)	Entropy
Tae-Kyu Kim (2006) (Tae-Kyu <i>et al.</i> , 2006)	Edge detection
Chun-Hung Shen (2006) (Chun-Hung and Chen, 2006)	Discrete Cosine Transform (DCT)
S.Y.Lee (2006) (Lee <i>et al.</i> , 2006)	Discrete Cosine Transform (DCT)
Weibin Tang(2008)(Weibin <i>et al.</i> , 2008)	Laplace filter

Mobile video broadcast

Mobile video broadcast is a service on 3G network that has great demand (Drude and Klecha, 2006; Herrero and Vuorimaa, 2004). The survey by (Jordan and Schatz, 2006) shows user spend 12-50 minute to watch news, sport and weather forecast in real-time in the quality closed to television broadcast. The contents are streamed directly using internet from file server to smart phone (Figure 9) or terrestrial broadcast

There are two standards in terrestrial mobile broadcast: DVB-H (Digital broadcasting –Hand held) and DMB (Digital multimedia broadcast) (Jordan and Schatz, 2006). Both systems required special equipment in both client and network station, which must have three important properties (Faria *et al.*, 2006; Kampe and Olsson, 2005).

First, the system must support energy saving mode, receiver must stop receiving data in some moment using Time Slicing technique to prolong battery life (Kornfeld, 2004).

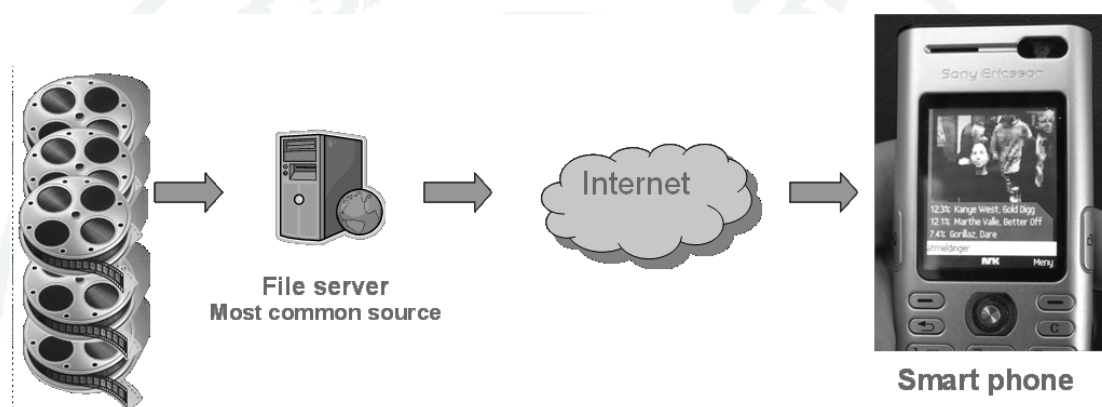
Second, the receiver is always moving, thus the system must support roaming from cell to cell.

Finally, data transmission must be robust to multi-path channel where signal is highly disrupted by noise.

However, live feed mobile broadcast requires content modified. Cell phones have the screen size limitation because the screens are small. Video must be cropped (Evans-Pughe, 2005) to prevent letterbox or black bars effect. Nowadays, most television program is produced in screen format that do not match cell phone display. The properties of devices in details are shown in Table 3.

Table 3 Receiver comparison

	<i>Devices</i>	<i>Screen resolution</i>	<i>Energy usage</i>
Portable receiver	Car receiver	800x600	Do not concern
	Laptop	1024x768	Do not concern
Hand held receiver	PDA	320x240	Concern
	Cell phone	192x176	Concern

**Figure 9** Mobile video broadcast framework

Video indexing

Exponential growth of video contents on the internet makes cyber life uncomfortable for users to select the most appropriate contents; however, redundant uploads/posts of the same contents make internet much less enjoyable. Redundant post on the internet is usually related to violation to copyright laws and website policies as well. To prevent redundant uploads, video similarity detection algorithms have been proposed (Google, 2013a). There are two approaches to tackle this problem: frame-by-frame comparison method and signature based method. Although frame-by-frame comparison achieves high accuracy, it requires exhaustive computation; therefore, it is impractical for long video. The signature based method is about to represent video contents by a smaller vector and then that vector is used to

be compared instead of the original content; therefore, it is faster and more practical, even though it may lose accuracy.

1. Video signature

Early signature-based technique was about to digest video contents using strong hash algorithm, such as SHA (Bolosky *et al.*, 2000; Mogul *et al.*, 2004). This method is efficient for comparing videos that contained exactly the same contents; however, it is not practical for real application since there are many existing formats of video and the video files usually contains other additional information (A *et al.*, 2000). Many other techniques have been proposed as follows literatures.

There are many video features that can be used as video signature. Parts of video stream such as DCT coefficients extract from MPEG streams is used to for internet video similarity detection (Wu and Polychronopoulos, 2012) however this method is limited to the MPEG encoded videos. Weighted color component (BaoFeng *et al.*, 2010), color histogram (Sánchez *et al.*, 1999) and color with applying coherence vector (Lienhart *et al.*, 1997) are used as signature to detect the television commercials. Color deterioration in motion pictures usually degrade signature quality. Therefore, there are many proposed techniques to improve robustness of color. For example, Xian-Sheng *et al.* proposed to downsample video frame to create the temporal shape of relative intensity (Cao and Zhu, 2009; Xian-Sheng *et al.*, 2004) and Lifeng *et al.* proposed to use local average of gray-level values for signature creation (Shang *et al.*, 2010). However, in many situations in real applications, color deterioration is a main trouble to video signature detection. Other proposed techniques are based on high-level feature extraction, such as motion (A *et al.*, 2000), shot durations, event length (Mohan, 1998), camera behavior (Ardizzone *et al.*, 1996; Flickner *et al.*, 1995; Shivakumar, 1999; Xie *et al.*, 2012), and object recognition (Dong *et al.*, 2008; Wan-Lei *et al.*, 2010). These types of signatures can represent its source videos; however, it suffers many problems. Color histogram is simple to extract but it is not robust to color deterioration. Motion is robust to color deteriorated but it required complex calculation to extract motion signatures, such as motion estimation and template matching. Combined features to

create signature can gain more accuracy than the previous proposed ones but it required more computational power to prepare.

2. Signature comparison

Sliding window method was introduced to solve this problem in order to measure distances between many small video chunks rather than a full-length video. This method is feasible for comparing different length signatures and detecting a small video section that is a part of a long video, such as TV commercials (BaoFeng *et al.*, 2010). However, sliding window does not allow frame deletion, frame insertion, and comparison between video clips that use different frame rate. To solve this problem, sequence alignment technique is introduced. The majority of alignment technique is based on Dynamic programming, such as Needleman-Wunsch algorithm (Xian-Sheng *et al.*, 2004), longest common subsequence (Shang *et al.*, 2010) and Edit distance (Mohan, 1998). These algorithms are still effective only for short videos. However, processing long video by dynamic programming still has pitfalls that it still takes long time if both video clips are quite long. The method proposed in (Xian-Sheng *et al.*, 2004) selects small chunks of video to compare as it does in sliding window and then add more chunks in dynamic programming style in order to compute similarity. Moreover, this method allows video insertion and deletion. Generally, features of video are represented by vectors of real numbers, which is still computationally intensive. Therefore, the method in (Shivakumar, 1999) proposed to use a sequence of discrete symbols extracted/transformed from video content and then compare them by string edit distance. This method improves accuracy, but it still requires exhaustive ranking (Xie *et al.*, 2012). To avoid exhaustive ranking, research in (Ardizzone *et al.*, 1996) proposed to use Laplace of Gaussian filter to extract a key-frame and use only key-frame as signature. Video key frame itself represents information in video shot; therefore, it eliminates redundancy. However, this method relied heavily on unreliable key-frame extraction process, bad tuning can result a signature that does not contain vital information.

Cinematic Structure

Cinematic structure is divided into three parts: camera, performance and sound. By using composition, director of photography can control the audience's attention to the main content. Focus and camera movement have the important role in drawing attention to the main character (Katz, 1991; Mamer, 2003).

Focusing helps draw attention of the audience to main character, focus can divide into 2 types: deep focus and shallow focus. Deep focus shot is created using complex optic module; the objective is to create the image that all parts of frame has the equal sharpness. Shallow focus created a limited depth shot, this type of focus is used to draw attention to the in-focus object while blurs the others. Rack-focus is also used to move the attention to the other part of frame (Figure 10) and also used for scene transition.

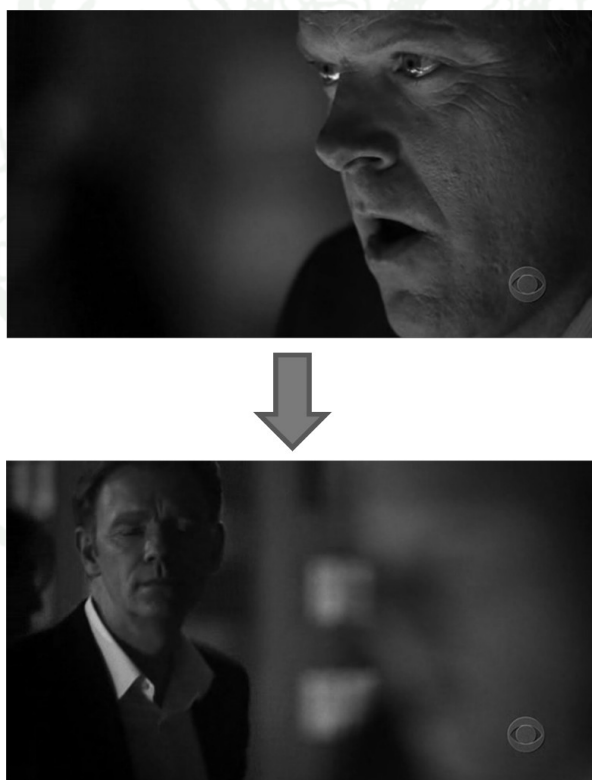


Figure 10 Example of Rack-focus shot in the same shot

Camera movement is used to replace the process that required some editing to keep important object in frame. The camera movement can divide into two types: Pan-tilt and tracking. In panning mode the camera itself is not moving, the only camera angle is changed. There are three objectives for camera panning: adjust image frame to track object, to cover the whole scene and to draw attention to the new scene (Figure 11). Tracking is created using crane or dolly. The objective is to eliminate distraction and draw the attention to the tracked object Figure 12.

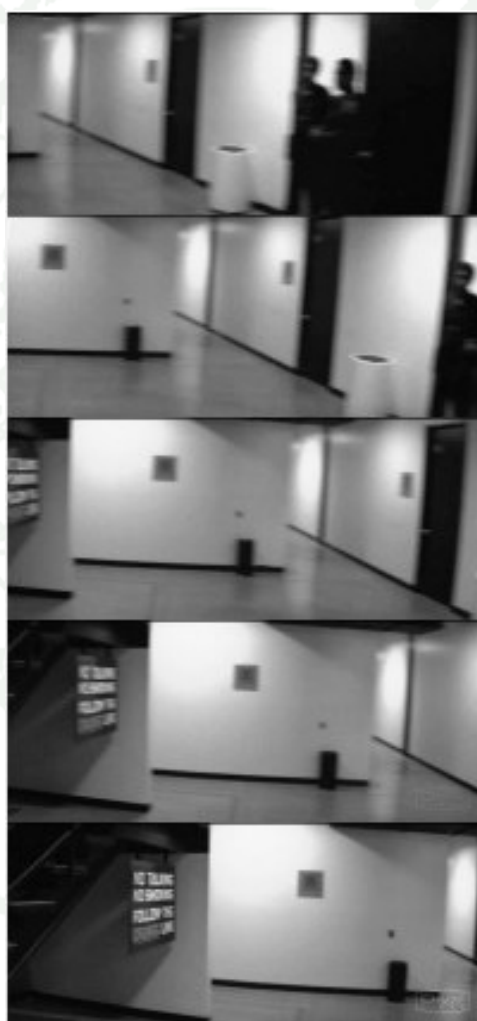


Figure 11 Example of pan shot



Figure 12 Example of tracking shot

MATERIALS AND METHODS

Materials

1. Computer System

1.1 Hardware

- Computer Server Intel(R) Xeon(R) CPU E5620@ 2.4 GHz
- RAM DDR3 98376 MB
- Hard disk 4 Terabytes
- Monitor 17 Inches
- Keyboard
- Mouse

1.2 Software

- MATLAB Version 7.9.0.529 (R2009b)
- VirtualDub 1.9.10
- GNU PSPP

2. Data

- Transformers: Revenge of the Fallen (2009)
- Raiders of the Lost Ark (1981)
- Indiana Jones and the Temple of Doom (1984)
- Indiana Jones and the Last Crusade (1989)
- Mercury Rising (1998)
- Star Trek (2009)
- National Geographic photo of the day dataset
- CC_WEB_VIDEO dataset

Methods

Our proposed automatic image and video cropping based on cinematic features.

Cinematic analysis is proposed as a replacement for computational visual attention model, in order to generate an importance map and reduce computational complexity.

Our framework is composed of three main parts shown in Figure 13: cinematic feature extractor, importance-map generator, and image-video cropper. The cinematic analyzer and importance-map generator are located on the server side, while the image-video cropper is located on the remote client side. The importance-map generator is used for tagging important parts of frames by analysing video or image content based on cinematic features. The video cropper part will use this importance map to generate cropping coordinate and optimize viewing area to match the display.

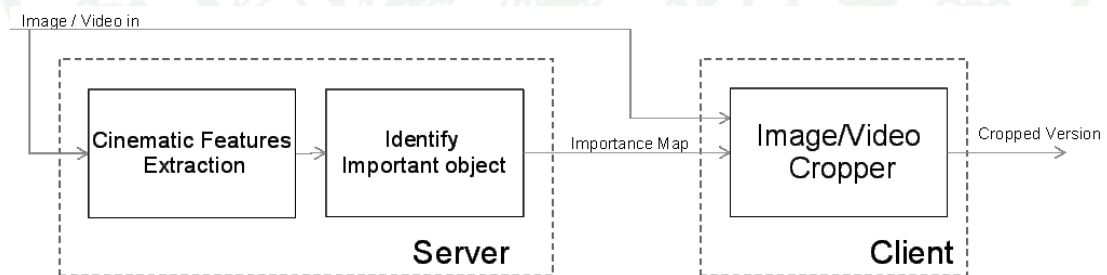


Figure 13 Model of our proposed image and video cropper system

1. Cinematic features

Cinema is different from other types of movies, such as home or surveillance video, because cinema is carefully produced under cinematic rules such as camera configuration, sounds, lighting, and sequence of shots to influence the audience interpretation of its content (He *et al.*, 1996;Katz, 1991;Prince, 2004). Filmmakers

uses cinematic rules to make a shot in order to persuade the audience to focus on important action and ignore unimportant objects or background (Prince, 2004). We can reverse this by extracting cinematic features and using them to the location of important objects in video. Interpretation to identify the important objects from cinematic features is classified as shown in Table 4.

Table 4 Cinematic features and interpretation

<i>Type</i>	<i>Camera Behavior</i>	<i>Object Behavior</i>	<i>Interpretation</i>
1	Fixed	Stationary	Object in focus is an important object
2	Fixed	Moving	Object that move is an important object
3	Pan/ Move	Moving	Tracked object is an important object
4	Pan/Move	Stationary	Every object is equally important. No exactly important object.

In this research, we use cinematic features to implicitly detect the camera's behaviour and object's behaviour to identify focused objects and tracked objects. However, in reality, such information is not explicitly available and using image processing to acquire that information is also not a straightforward process. Each type is illustrated in Figure 14.

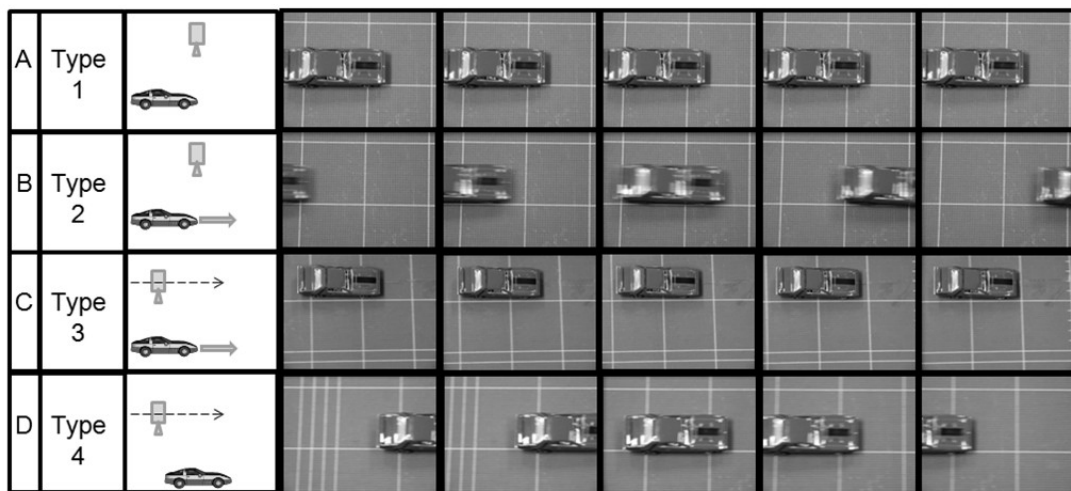


Figure 14 Examples of three different cinematic types

Type 1: Camera is fixed; every object is still; and every frame in this shot is identical (A). The shallow focusing technique is used to emphasize attention or significant objects while blurring the unimportant ones (Prince, 2004). It is easy to determine which object is in-focus by looking for the sharpest object.

Type 2: Camera is fixed and objects are in motion. The important object should be in focus when taking a shot. However, if we take a look on each frame, there is no sharp image because the important object is in motion in every single frame. The faster it moves, the more blurred it is. By our observation on every single frame in one continuous shot, we found that the most important object is not likely to be in focus. Figure 15 illustrates the observation of focal point of stationary moving images. Images in both A and B are taken from the same lens configuration. However, the front coin is stationary in A but moving in B. The front coin in B looks almost invisible because of the effect of motion blur. The effect of motion blur on moving objects is also shown in B; the moving car is blurred in every frame even if the intention is to keep it in focus. We can see that effect of motion blur occurring at the focal point, which is the object of interest. This type of scenario is difficult for auto-focus algorithms (Ooi *et al.*, 1990).

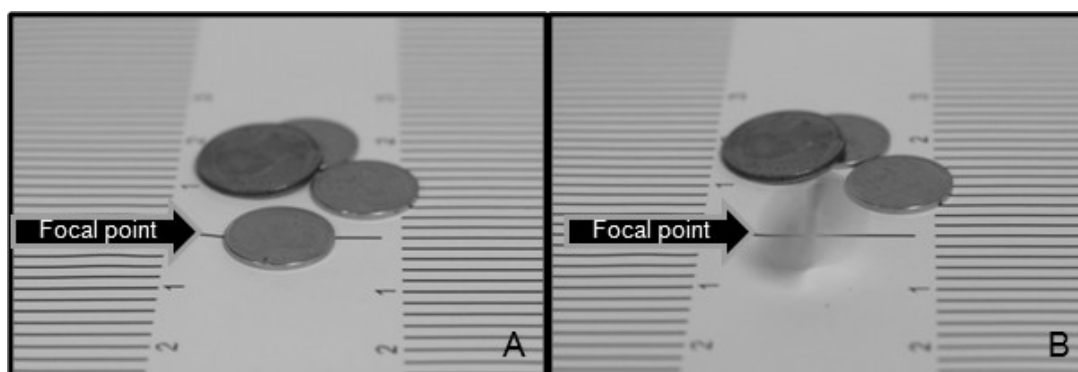


Figure 15 Observation on effect of motion blur in the same lens configuration

Type 3: Camera is panned or moved to track an important object. This shot type creates some motion in video. An example is shown in C. The camera tracks moving object in order to keep it in the frame. This creates heavy motion of the background and little motion of the tracked object.

Type 4: Camera is panned or moved but there is no intention to track anything. The intention of this shot often is to demonstrate the actual dimension of the scene; for example, camera is panned across the field without any particular interesting objects. This shot always creates the heavy motion on every part of frame because information at every pixel is changed. An example of this shot type is shown in D.

Generally, we can design a video-cropping algorithm by identifying types of cinematic features used in each shot and then creating four independent algorithms for each cinematic type. Algorithm for Type 1 is to detect focal points. Types 2 and 3, have similar behaviour that there are different levels of motion between areas on the object of interest and the background. Type 2 produces high motion at the object and little motion in the background; on the other hand, Type 3 produces high motion at the background and little motion at the object. In both cases, the area with minor motion is an area of object of interest. Type 4 produces uniform motion over the whole frame area. Therefore, we re-classify our cinematic types into 3 classes: In-

focus object is important (Type 1), Minor-motion object is important (Types 2 and 3) and everything is equally important (Type 4).

Class 1: In-focus object is important. There is no motion in the video shot. This shot class occurs when a fixed camera captures motionless objects. Therefore, the in-focus object is considered to be important.

Class 2: Minor-motion object is important. There are two possible situations for this shot class: moving object is captured by fixed camera and moving object is tracked by camera on dolly. Firstly, since the camera is fixed, there is very little background movement and most motion is likely from targeted objects. Secondly, moving objects are tracked by panning or camera being on a dolly. Usually the characteristic of this shot is likely a large amount of moving background and small amount of motion produced by tracked objects.

In this class, we generate importance maps based on small objects (or parts of frame) that have different motion behaviour from dominant ones. If there is little motion occurring then the heaviest motion area is the important area but if the motion is great then the lowest motion area is the important area. For example, when a camera captures a car running towards camera, the car is considered to be important because it generates little motion while a large area of background has no motion. On the other hand, if a car is tracked by camera panning, the car is considered to be important because the moving background generates huge amount of motion while there is little motion generated by the tracked object. In both examples, it can be noticed that the most important area occurs in “minor motion” area.

Class 3: Everything is equally important. There are two possible situations for this class. The first one is when a fixed camera is capturing a crowd of moving objects. The second one is when the camera is panned, tilted, or zoomed without any intentions to track any particular object. It is used to reveal dramatic information by enlarging the viewer’s field of view (Prince, 2004), in this case we can interpret that everything in the frame is “equally important”. For example, the camera is panned at

constant speed to review the full structure of a building. This class is very difficult to crop, however, in our observation we found that this class often occurred only at the beginning or the ending part of a film.

These cinematic rules can be expressed as decision chart shown in Figure 16. By following these cinematic rules, we can design an algorithm to locate important objects.

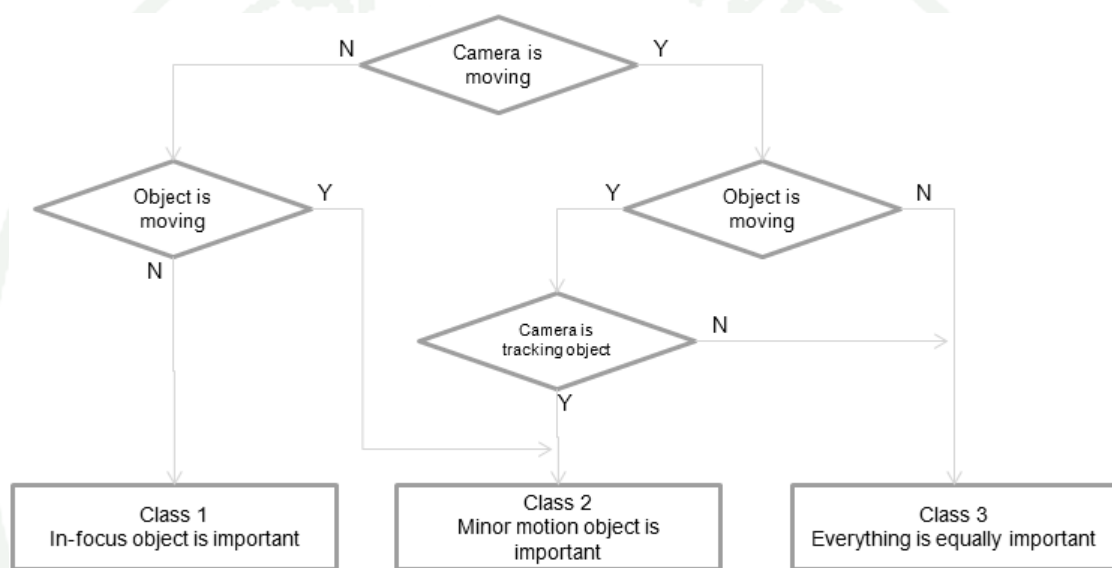


Figure 16 Decision chart for Important object identifier based on characteristics of motion in shot.

2. Image Cropping Algorithm

Because most digital cameras have autofocus as a standard feature and human face detection is an option for many cameras, focal point and human face become evidence about where the attentive locations defined by photographers are. For focal point detection, we investigated research on camera's autofocus system and used the best filter to detect the location of focal points in a picture. Results of the filter are called "focus map". For human face detection, any algorithms such as (Colmenarez

and Huang, 1997; Viola and Jones, 2001; Yang and Huang, 1994), can be used to generate a “face map”.

The concept of this algorithm is to 1) find the focus area and an available face area to generate a focus map and a face map, respectively; 2) combine both feature maps to create an importance map; and 3) determine an optimal cropped area based on a given aspect-ratio using a genetic algorithm. The process is shown in Figure 17. The importance map generator and optimizer are described next.

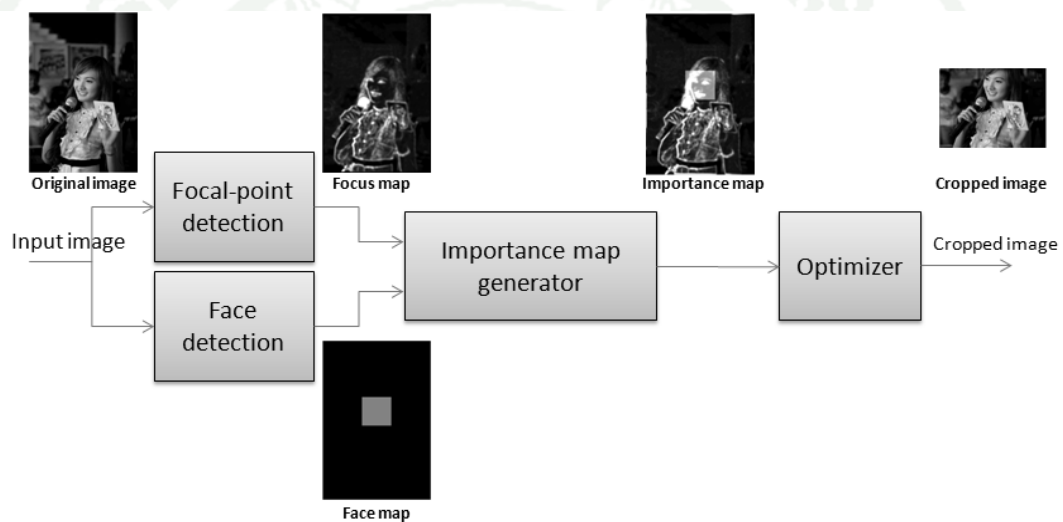


Figure 17 Proposed automatic image cropping model

2.1. Importance map generator for image cropping

The importance map generator is composed of two feature maps: focus map and face map. The face map is generated by detector of Viola and Jones (Viola and Jones, 2004) that is implemented in OpenCV (Bradski, 2000) to detect human faces. In this research, human face is an additional feature used to improve accuracy in images where human faces are detected. However, the main contribution of this work in importance map generator is the focus map generator since it is the clue given by photographer.

Focus map is generated based on the same principle of autofocus system by applying the same filter used in a camera's autofocus system as the filter for finding focus location in an image. Laplace filter, one of the most accurate filters for autofocus camera system (Diansheng *et al.*, 2010), is adopted into this research. Let I be an input image and $w(s, t)$ be a Laplace kernel. In this research, we used

$$w(s, t) = \begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}.$$

A focus map I_h is defined by equation (1).

$$I_h(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b w(s, t) I(x + s, y + t) \quad (1)$$

Figure 18 depicts focus maps of the images that are taken from the same location but with different focal points. By observation, highlights appeared on importance maps are focused area in original images.

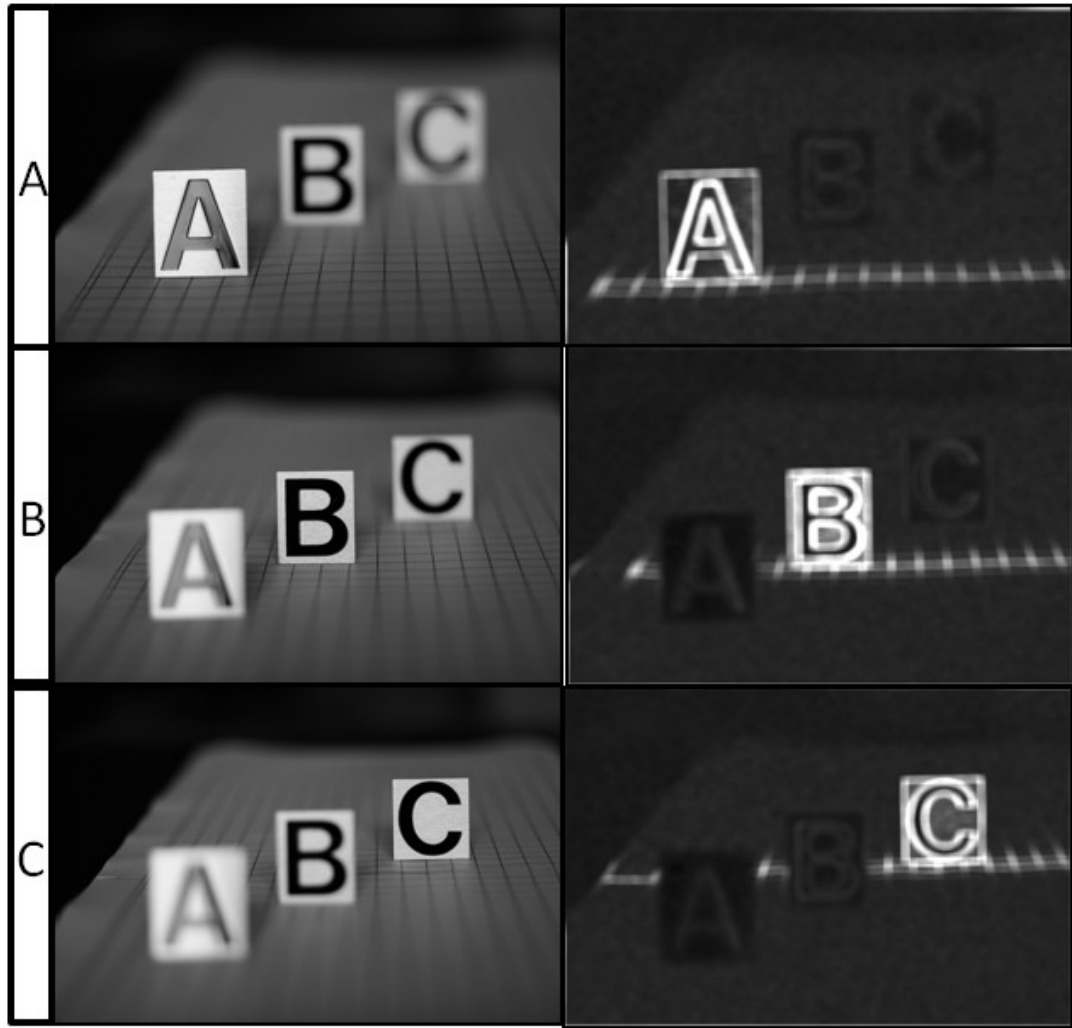


Figure 18 Importance maps of photograph at different focal point.

If all objects are located at the hyperfocal distance (focus at infinity) then every part of image is in-focus, hence the focal map will not have any clue to indicate a potential cropping area. For example, in images of stars, every star is in-focus no matter how far a star is from the others. Therefore, we require an alternative feature for this case. Face is also an important feature for cell-phone photographers, since a cell-phone camera cannot usually create depth of field. Therefore, we created an importance map I_g as a linear combination of the focus and face maps, as should in equation (2) where I_f is the face map; and w_1 and w_2 are weighted.

$$I_g = w_1 I_h + w_2 I_f \quad (2)$$

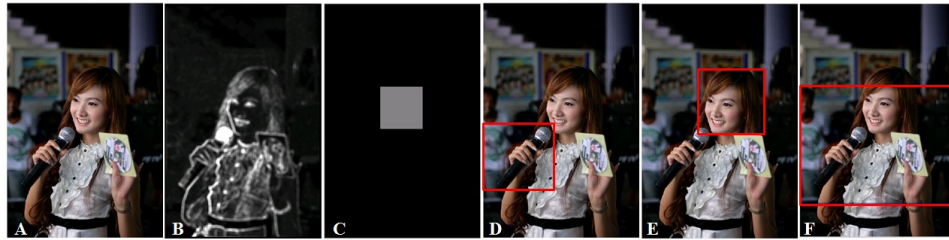


Figure 19 Example of cropping result from different maps



Figure 20 Result of this algorithm in pan-and-scan mode

In many cases, results from the two maps suggest totally different cropping areas; therefore, it is hard to make a decision. For example in Figure 19, the focus map I_h (Figure 19B) shows that a focus area of Figure 19A is mainly at the microphone and the face map (Figure 19C) indicates that the important area is on the girl's face. Cropped versions using only focus map and only face map are shown in Figure 19D and Figure 19E, respectively. However, the suggested results should be Figure 19F, where the surrounding area outside the feature maps must be investigated.

Selecting a cropped area is about finding the area that has maximal information per area. There are two approaches to image cropping: “pan-and-scan” and “automatic zoom”. The pan-and-scan method crops an image to fit the aspect ratio of the display or printing devices, such as printing a widescreen image on A4 paper. Automatic zoom selects any area in the image in order to display that particular area very clearly. For example, to match aspect ratio of the cell phone display not only must an image be cropped, it also must be zoomed around the important object to see it easily on a small display.

2.2. Cropping window optimizer for image cropping

Pan-and-scan selects a cropping window of the required size from the source image that has maximal information. Usually, either the widths or heights of source image and cropping window will be equal. The best cropping location should produce the maximum sum of I_g , in equation (3), where (x', y') are coordinates of one corner of a cropping window of size w' by h' . This equation is solved using a genetic algorithm. An example of pan-and-scan result without face map is shown in Figure 20. An example of automatic zoom with face map is shown in Figure 19F.

$$[x', y'] = \text{ArgMax}_{x', y'} \sum_{x=x'}^{x'+w'} \sum_{y=y'}^{y'+h'} I_g(x, y) \quad (3)$$

In principle, we can just add a zoom factor as a new parameter along with aspect ratio to the pan-and-scan algorithm; however, it is not efficient since this problem is NP-complete and requires a brute force method to find the exact solution. In this research, we get around this by proposing an automatic zoom algorithm by creating a wrapper module on top of the pan-and-scan algorithm. The wrapper takes a zoom rate (d ; $0 < d < 1$), a given aspect ratio (θ), a zoom limit ϕ , and threshold t on the amount of information allowed to be lost. The concept of the wrapper is to reduce the size of cropping window until we find the smallest area that can keep information more than the given threshold. Algorithm 1 shows pseudo code of the proposed automatic zoom algorithm. The algorithm starts by creating candidate target window which is d times smaller than the original size of image (line 5). Then the algorithm uses this window to crop the source image using proposed “pan-and-scan” method (line 12) to create the candidate result. After that, it will try to reduce size of cropping window further to find a new candidate. If the loss of a newer candidate is over the threshold, the current candidate becomes the final result and the process is ended.

An example of a result from Algorithm 1 is depicted in Figure 21. A green window in Figure 21A indicates result of automatic zoom algorithm. Figure 21B is importance map of Figure 21A. In Figure 21B, all candidates of cropping window

are shown as red rectangles; white rectangle indicates the first candidate that loss information more than allowance threshold; and the green rectangle is the final decision of cropping widow.

The genetic algorithm in the optimizer module (line 12) is the stochastic process that can create slightly different results in every time it performs. In our experiment every configuration was running 20 times and use average result.

Algorithm 1 Automatic zoom

Input: importance-map (I_g), zoom rate (d), information-loss threshold (t), zoom limit (ϕ) and target aspect ratio (θ)

Output: Rectangular coordinate $[x, y, w, h]$

Step 1: $z = \frac{\log(\phi^{-\phi})}{\log(d)}$

Step 2: $x'_0 = 1, y'_0 = 1$

Step 3: $h'_0 = \text{height}(I_g), w'_0 = \text{width}(I_g)$

Step 4: FOR $i=1$ to z

Step 5: IF $h'_0 \geq w'_0$ THEN

Step 6: $w'_i = w'_{i-1} \times d$

Step 7: $h'_i = w'_i \times \theta^{-1}$

Step 8: ELSE

Step 9: $h'_i = h'_{i-1} \times d$

Step 10: $w'_i = h'_i \times \theta$

Step 11: ENDIF

Step 12: $[x'_i, y'_i] = \text{ArgMax}_{x', y'} \sum_{x=x'_{i-1}}^{x'_{i-1}+w'_i} \sum_{y=y'_{i-1}}^{y'_{i-1}+h'_i} I_g(x, y)$

Step 13: $I_{g_i} = \sum_{x=x'_i}^{x'+w'_i} \sum_{y=y'_i}^{y'+h'_i} I_g(x, y)$

Step 14: IF $i > 2$ AND $\frac{I_{g_{i-1}}}{I_{g_i}} > t + 1$ THEN

Step 15: EXIT FOR

Step 16: END IF

Step 17: END FOR

Step 18: RETURN($x'_{i-1}, y'_{i-1}, w'_{i-1}, h'_{i-1}$)

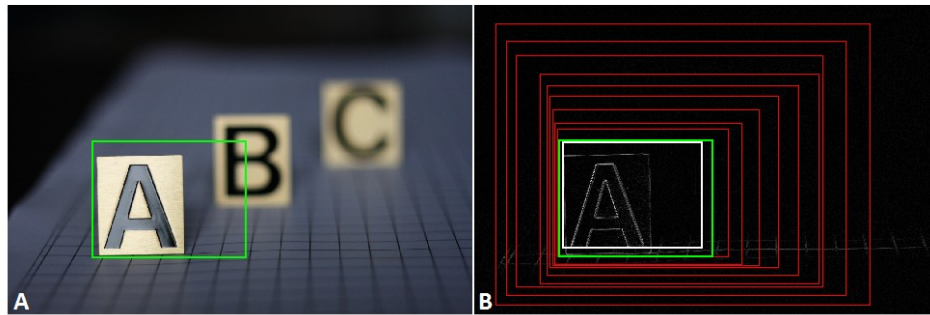


Figure 21 An example result of automatic zoom algorithm

3. Video Cropping Algorithm

The importance-map is generated by detecting cinematic features as described in the previous section. In this section, we describe an algorithm to classify video shots and generate the map. A block diagram of the importance-map generator is shown in Figure 22.

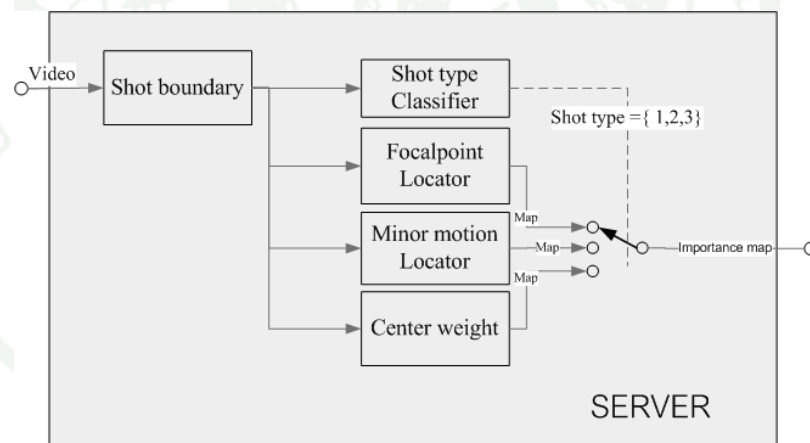


Figure 22 Block diagram of importance map generator in server-side module.

3.1. Importance map generator for video cropping

After an input video is separated into many small shots by the cut-detection algorithm, it is then fed into three importance-map generator modules simultaneously. The final result is selected by the shot-type classifier from those three results.

The algorithm uses average motion intensity of the video shot to determine the most appropriate class. Usually, there is almost no motion, moderate motion, or extreme motion in classes 1, 2, and 3, respectively. We use absolute differentiation of images to determine amount of motion between two consecutive frames and then average them through the whole shot as amount of motion, m , which is defined by:

$$m = \frac{\sum_x \sum_y \left(\sum_{i=1}^{n-1} |\mathbf{Frame}_i - \mathbf{Frame}_{i+1}| \right)}{n \times r} \quad (4)$$

$$c = \begin{cases} 3; m > \theta_2 \\ 2; \theta_1 < m \leq \theta_2 \\ 1; m \leq \theta_1 \end{cases} \quad (5)$$

Where r is an image resolution, n is number of frames, and θ are thresholds to determine class c .

Class 1: in-focus object is important, the goal of this class is to determine important parts of a video frame when there is little motion in the frame. In this case, we use the focal point of the camera as a significant clue to identify a region of interest. The focus map is generated in the same process as described in image cropping section. In order to transmit the importance map to the client efficiently, we represent them as horizontal and vertical projections using equations (6) and (7),

respectively, where w is frame width; h is frame height; $\mathbf{P}_y(y)_j$ is horizontal

projection and $\mathbf{P}_x(x)_j$ is vertical projection of image frame j and I_h is focus map.

This method reduces the size of the importance map and the computational complexity on the client. Ideally, localization of the focal point on any single frame is accurate enough to represent every frame in the shot since if there is no motion then every frame should be identical. However, in practice, every frame in a shot is not pixel-wise identical; errors of mechanical parts in an analog camera can result in image shaking vertically and horizontally; and digital cameras can produce image frames with random noise. To overcome this problem, all frames need to be

processed in order to find the average and then use it to create an importance map. The importance map is a pair of vertical and horizontal projections (**V** and **H**) of the high-pass component. **V** and **H** are defined in equations (7) and (8).

$$\mathbf{P}_y(y, j) = \sum_{a=1}^w \mathbf{I}_h(a, y)_j \quad (6)$$

$$\mathbf{P}_x(x, j) = \sum_{a=1}^h \mathbf{I}_h(x, a)_j \quad (7)$$

$$\mathbf{H}(y) = \frac{1}{|j|} \sum_{a=1}^{|j|} \mathbf{P}_y(y, a) \quad (8)$$

$$\mathbf{V}(x) = \frac{1}{|j|} \sum_{a=1}^{|j|} \mathbf{P}_x(x, a) \quad (9)$$

In the equations, P_x and P_y are horizontal and vertical projections of the high-pass filtered image I_h , where j is the frame and $|j|$ is the frame count. The average of all frames is computed in equations (8) and (9) to create vertical and horizontal importance maps **V** and **H** as shown by the white line in Figure 23.

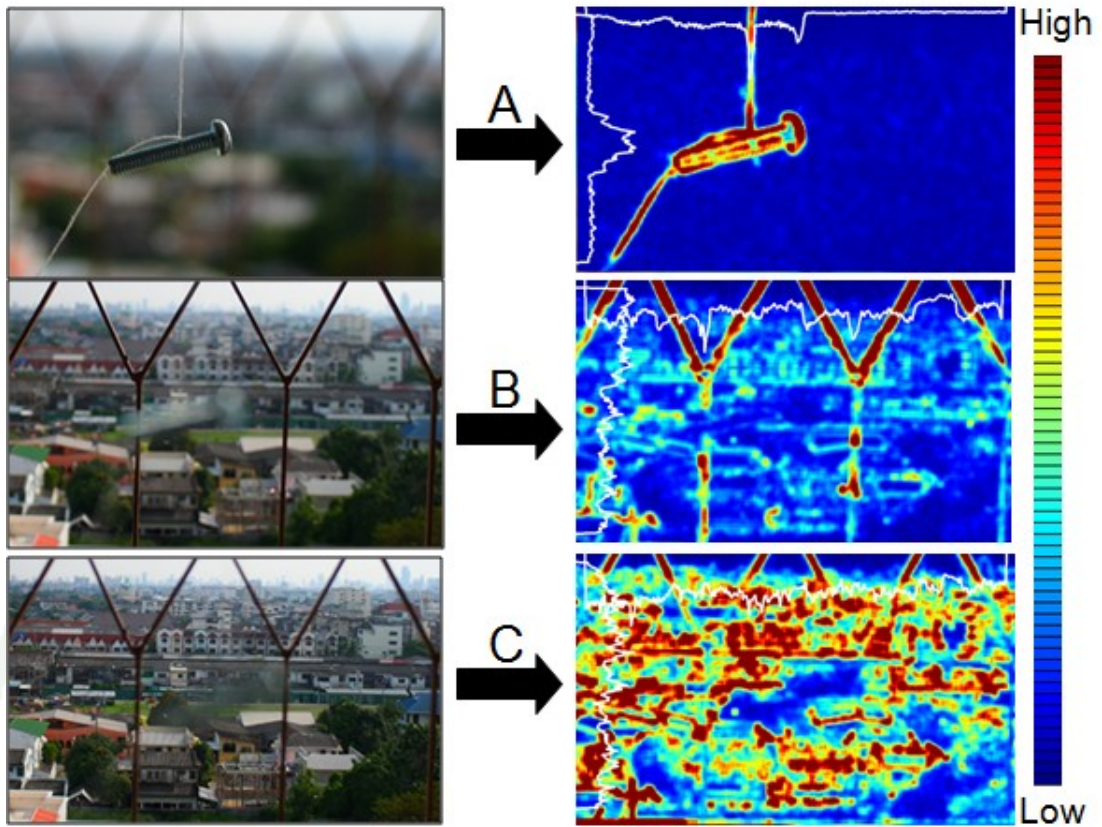


Figure 23 Importance map generated by different focus position

Class 2: Minor motion object is important, in this class we rank the important area by considering the distinguishing motion pattern of one area from the rest.

In order to do motion analysis, motion parts of the frame are extracted by motion estimation. Generally, motion estimation works well and gives the best result for pixel or blob analysis while it might produce poor results for semantics analysis (Piamsa-nga and Babaguchi, 2004). To find a distinctive or contrast object using a noisy motion vector is difficult. In order to make the algorithm more robust to noise and video compression artefacts, we proposed an alternative method to overcome this problem. First, we calculate the motion matrix \mathbf{M} by pixel-wise sum of the differences of every image sequence \mathbf{I} in the shot using equation (10).

$$\mathbf{M} = \frac{\sum_{i=1}^{n-1} (|\mathbf{I}_i - \mathbf{I}_{i+1}|)}{n} \quad (10)$$

Second, finding a majority magnitude of \mathbf{M} by using histogram $\mathbf{Hist}(f,i) =$ histogram (\mathbf{M}, bin) where f is a frequency, bin is the number of bins and i is index of a histogram bin. Let k be an index of the bin that has the maximum frequency in histogram $\mathbf{Hist}(f,i)$. There are two situations of concern here; moving object on a stationary background, and stationary object on a moving background, such as a camera dolly tracking an object.

We classify an image frame as type 2 (Figure 14B) or type 3 (Figure 14C) based on the pattern of the histogram. Since the histogram represents the distribution of motion density, if we use event number of the bin count then there are only two possible cases.

In case $k \leq \frac{bin}{2}$, it means that pixel frequency of stationary object is dominant,

by our observation it can occur by a moving object on a stationary background (type 2). The opposite motion to the dominant one is called minority motion. In this module, we are to find a part of the frame that contains minority motion or motion contrast to the background in the frame. We filter all pixels in \mathbf{M} that have a value in range of the bin that has the maximum frequency ($\mathbf{Hist}(f,k)$). This filter is shown in equation 11. Then, we generate the minority motion map \mathbf{I}_d by using equation (12).

The minority map \mathbf{I}_d is then converted to an importance map (\mathbf{V} and \mathbf{H}) using equations 13 and 14. An example of shot types is shown in Figure 24. Figure 24A shows the actual image sequence and Figure 24B shows its sum of the differences (\mathbf{M}). In B, the histogram in the first bin has maximum frequency; therefore, the element of $\mathbf{M}(x,y)$ that has maximum frequency will be removed in order to create its projections.

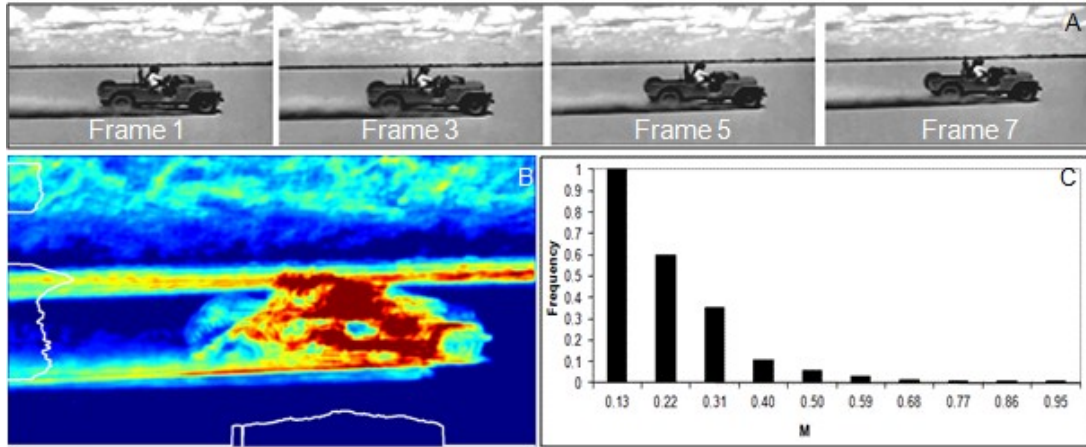


Figure 24 Class 2 shot type where $k \leq \frac{bin}{2}$

In case $k > \frac{bin}{2}$, the motion of the moving object is dominant. An example is shown in Figure 25; motion of the tracked object is less than that of the background.

Therefore, importance map is located where it has less motion, so $I_d(x, y)$ is equal to $-M(x, y)$ as describe in equation (11). Figure 25A shows frame examples of an image sequence and Figure 25B shows its sum of difference (M). This kind of sequence produces maximum frequency in the right-half of the histogram as shown in Figure 25C. The projection of its importance map (V and H) is shown in Figure 25B

$$M(x, y) = \begin{cases} M(x, y); & M(x, y) \neq Hist(f, k) \\ 0 & ; \text{others} \end{cases} \quad (11)$$

$$I_d(x, y) = \begin{cases} M(x, y) & ; k \leq \frac{bin}{2} \\ -M(x, y) & ; k > \frac{bin}{2} \end{cases} \quad (12)$$

$$H(y) = \sum_{a=1}^w I_d(a, y) \quad (13)$$

$$V(x) = \sum_{a=1}^h I_d(x, a) \quad (14)$$

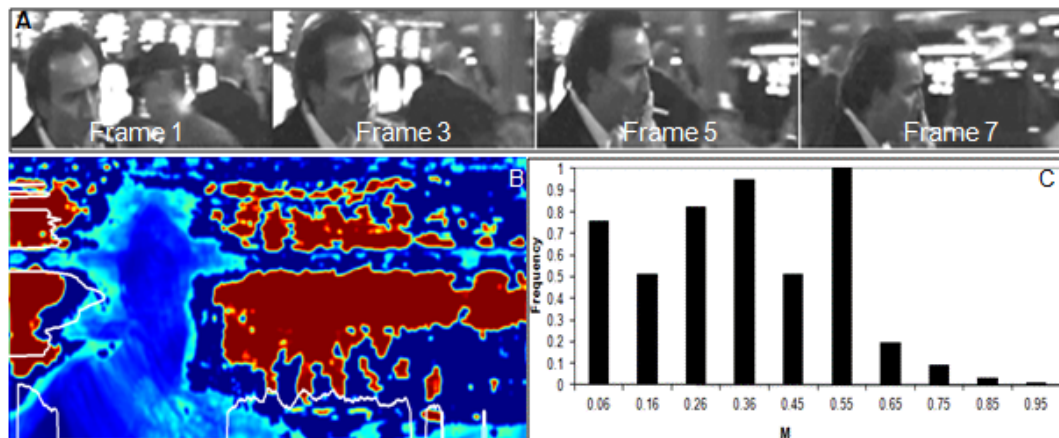


Figure 25 Class 2 shot type where $k > \frac{bin}{2}$

Class 3 everything is equally important, shot of Class 3 is a video shot having enormous motion. This type often occurs when camera is panned through the scene. There are no distinct motion and no focal point in any part of frame. Therefore, everything in the frame is equally important; and on the other words, nothing is more important than others. Because any kind of cropping algorithms could not produce different amount, we use center cropping. We forced decoder part to center crop by assigning Gaussian curve value as $\mu = 0$ to \mathbf{V} and \mathbf{H} . The video cropper at the client will generate cropping window that is growing from center of the video frame. The example of importance map for this class is shown in Figure 26

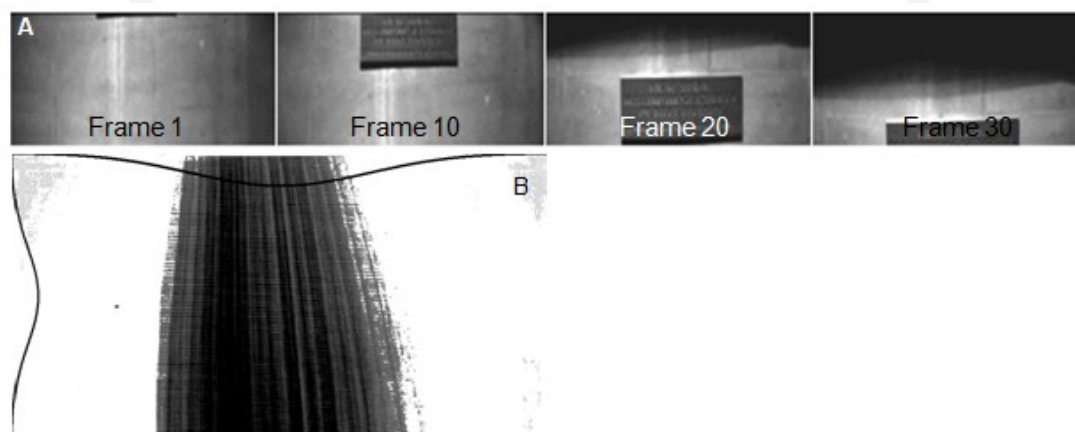


Figure 26 Class 3 shot type (Everything is equally important)

3.2. Cropping window optimizer for video cropping

The main function of the cropper unit is to crop video to fit a particular display size and aspect ratio such that most of the important information is preserved. Using importance-map information (V and H) provided by the encoder unit, the cropper can automatically crop video frames to fit a variety of screen sizes and formats with little computation time.

The cropper is composed of two main parts: pan-and-scan processor and video cropper. The function of the pan-and-scan processor is to determine the cropped window coordinates to match the target aspect ratio. A block diagram of the video cropper module is shown in Figure 27.

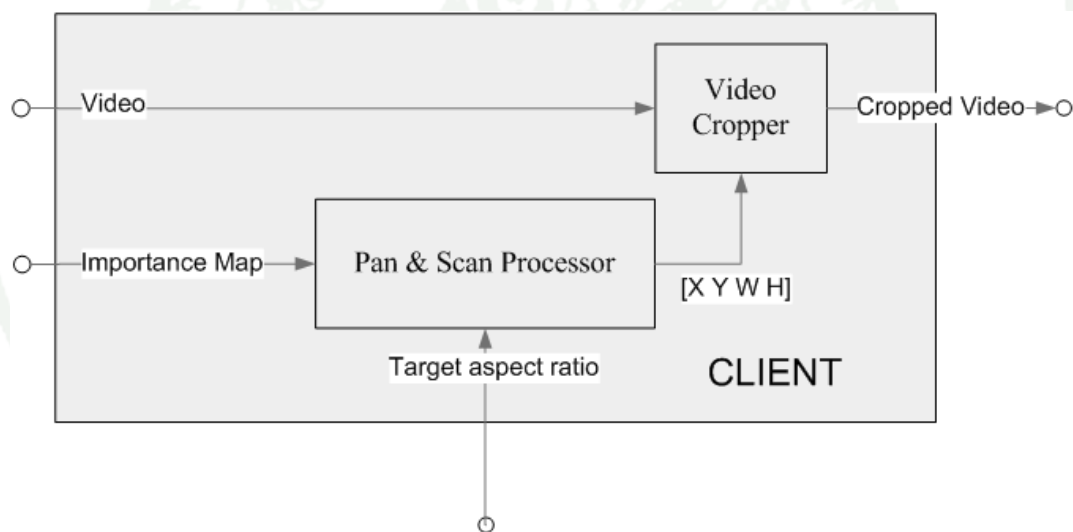


Figure 27 Block diagram of video cropper at client-side module.

3.3. Iterative zoom algorithm for video cropping

The main purpose of the process is to carefully trim video frame from outside-in to prevent error causing from local maxima and prepare the video for pan-and-scan step. The process of iterative zoom algorithm is controlled by two parameters: zoom

ratio (d) and zoom factor (z). The zoom ratio is about to determine the zoom step size for each iteration and zoom factor determine how much the frame area will be dropped. The output of this algorithm is window offset (Top and $Left$) and its size (w and h). Zoom factor can be estimated using equation (15) where ϕ is the magnification ratio, for example 2 is assigned to ϕ if 2 times magnification ratio is needed. Iterative zoom algorithm is described in Algorithm 3 and example algorithm result with z varies from 1 to 19 was demonstrated in Figure 28.

$$z = \frac{\log(\phi^{-1})}{\log(d)} \quad (15)$$

Algorithm 3 Iterative zoom algorithm

Input: Importance-map (V, H) and zoom factor (z)

Output: Rectangular coordinate [$Top, Left, w, h$]

Step 1: $FromX \leftarrow 1, ToX \leftarrow size(V), FromY \leftarrow 1, ToY \leftarrow size(H)$

Step 2: $w \leftarrow (ToX - FromX) \times d, h \leftarrow (ToY - FromY) \times d$

Step 3: $Left \leftarrow ArgMax_j \sum_{i=j+FromX}^{j+FromX+w} V(i)$

Step 4: $Top \leftarrow ArgMax_j \sum_{i=j+FromY}^{j+FromY+h} H(i)$

Step 5: $FromX \leftarrow Left, ToX \leftarrow w + Left, FromY \leftarrow Top, ToY \leftarrow w + Top$

Step 6: **Goto** step 2 for z times

Step 7: **Return** [$Top, Left, w, h$]



Figure 28 Example result of each iteration of an automatic zoom algorithm

3.4. Automatic pan-and-scan algorithm for video cropping

Our automatic pan-and-scan algorithm is performed automatically by sliding window on the video frame using a pan-and-scan algorithm shown in Algorithm 4. Given the target aspect ratio, this algorithm will produce final cropping coordinates expressed as $[X \ Y \ W \ H]$. The result is illustrated in Figure 29, the figure shows the actual video frame superimposed on the importance map V and H , the target window width w and height h are moving from the left position to the final position (x, y) calculated by.

Algorithm 4 Automatic Pan-and-scan

Input: Importance-map (V, H) boundary (w, h) and target window (w', h')

Output: Rectangular coordinate $[X, Y, W, H]$

Step 1: $\phi = \frac{w}{h}$ // original aspect ratio

Step 2: $\phi' = \frac{w'}{h'}$ // w' and h' are target screen width and height, respectively.

Step 3: *IF* $\phi > \phi'$ *THEN*

$$X = \underset{j}{\text{ArgMax}} \sum_{i=j}^{j+w'} \mathbf{V}(i) \text{ // Pan and scan}$$
 Step 4:
 Step 5: $[X, Y, W, H] = [X, 1, w', h']$
 Step 6: *ELSE*

$$Y = \underset{j}{\text{ArgMax}} \sum_{i=j}^{j+h'} \mathbf{H}(i) \text{ // Tilt and scan}$$
 Step 7:
 Step 8: $[X, Y, W, H] = [1, Y, w', h']$
 Step 9: *END IF*
 Step 10: Return $[X, Y, W, H]$



Figure 29 An example of cropper result of the automatic pan-and-scan algorithm

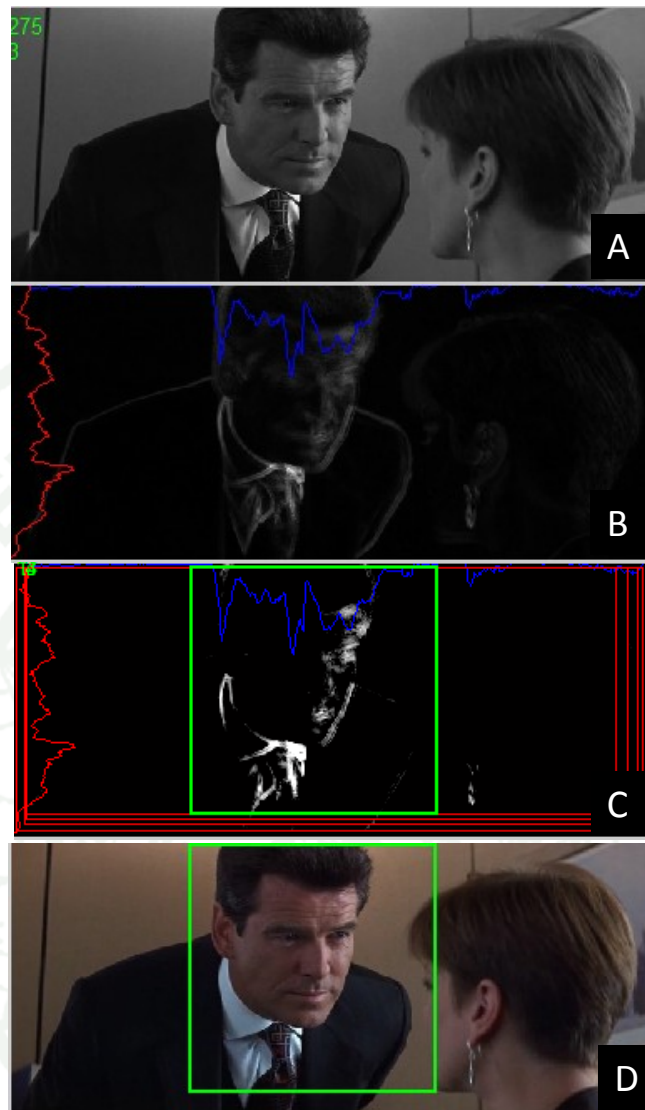


Figure 30 Example of video cropper output

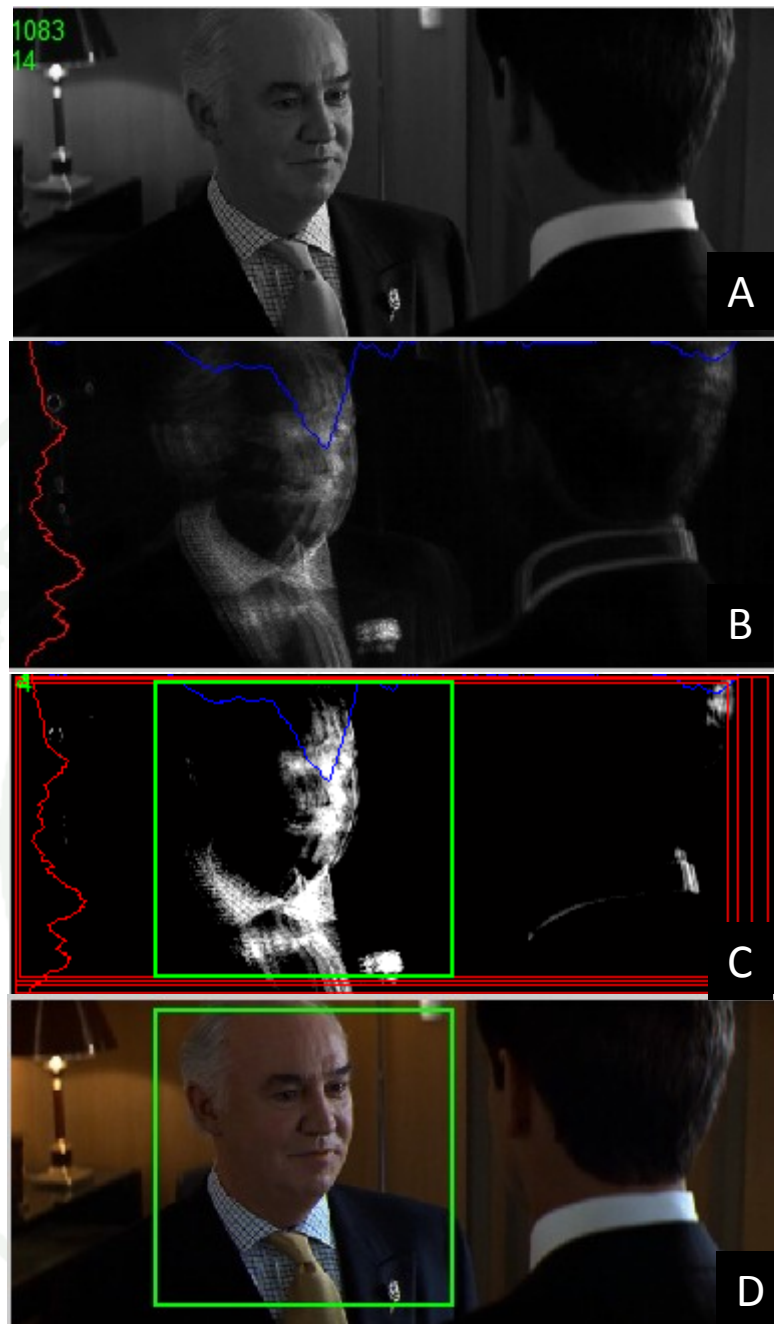


Figure 31 Example of video cropper output

Example result of all process are shown in Figure 30 and Figure 31, both images show the process of adapting aspect ratio from 2.35:1 to 1:1. Image (A) show original video frame, (B) show motion map as bright value and important map as projection, (C) show automatic zoom in each iteration as red box and automatic pan-and-scan is show in green box, finally cropped result is shown in green box in (D).

The comparison of this algorithm with center crop shown the more different aspect ratio is the more center crop is likely to fail. Results of 4 several aspect ratio differences are shown in Figure 32.

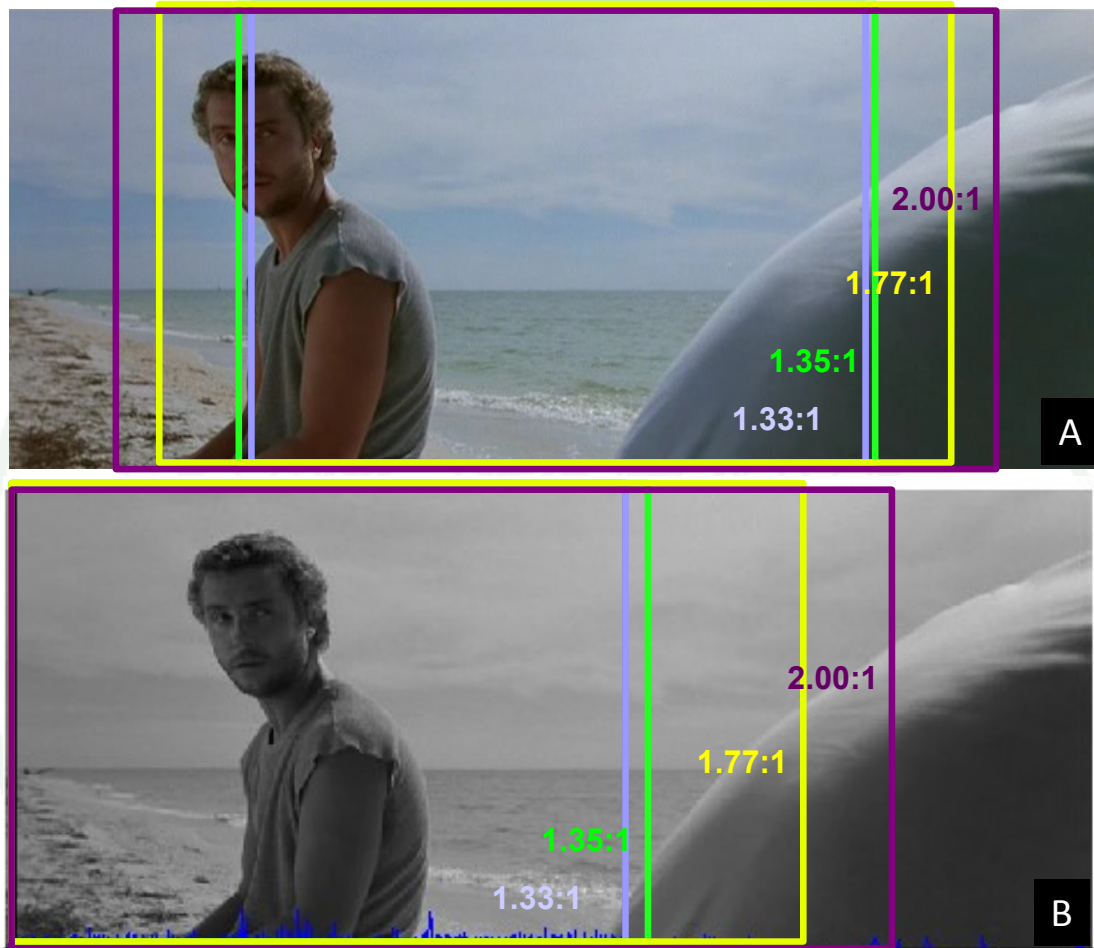


Figure 32 Compare algorithm result of each aspect ratio

The example of cropping result using this algorithm in tilt mode in three zoom parameter: 105%, 120% and 135% compared with tinder box mode (un-crop version) are shown in Table 5. The example result in pan-and-scan mode with zoom parameter: 100%, 120%, 160% are shown in Table 6..

Table 5 Result in tilt mode using difference zoom parameter (z) compared with un-crop version
































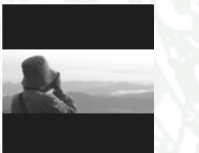




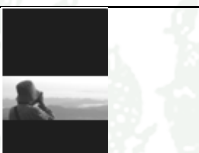


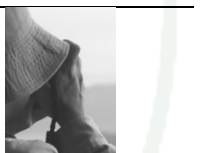
<i>Original Video</i>	<i>Tinderbox Version</i>	<i>105%</i>	<i>120%</i>	<i>135%</i>
				
				
				
				

Table 6 Result in pan mode using difference zoom parameter (z) compared with un-crop version

<i>Original Video</i>	<i>Letterbox version</i>	<i>100%</i>	<i>120%</i>	160%
				
				
				
				

4. Video indexing based on N-gram

Our system is depicted in Figure 33. First, features of interest are extracted from video stream. Second, those features are “textualized”, which is to transform features in multidimensional vector to string representation. Third, those strings are extracted as “grams” and then stored in a database as indices. The lower part of Figure 33 is about video querying. The query video has to be textualized and then used it to generate n-grams. The query n-grams are then used as video representation to match in the video archive database.

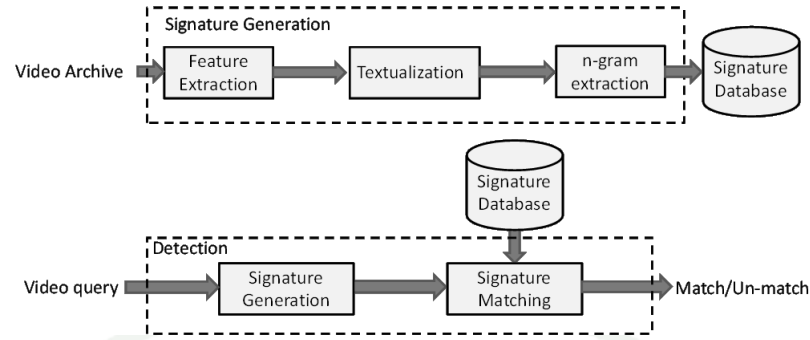


Figure 33 Proposed model of video copy detection based on N-gram.

4.1. N-gram signature generation

In order to generate a good signature, features must be carefully selected. Many previous research projects used temporal differentials of feature vectors to create signatures. This kind of features can amplify “landmark event” in video such as motion of object or scene cut detection in video sequence. However, landmark event cannot be a signature but a composition of small details around landmark can. In order to avoid problems of color deterioration problem and sampling-rate heterogeneity, we propose to use “luminance velocity” as feature. The signature generation is explained as follows. First, extract average luminance intensity of each video frame by converting image frame to gray scale and then compute average intensity of each frame to create a sequence of average luminance (I). Second, determine the “luminance velocity” by calculating the differences between adjacent elements of I using equation (16) where I_v is defined as “luminance velocity” and then its values are normalized to be in from $[0..1]$ by equation (17) where D is normalized luminance velocity vector.

$$I_v = \Delta I \quad (16)$$

$$D = \left(\frac{i_v - I_{v_{min}}}{I_{v_{max}} - I_{v_{min}}} \right) \quad (17)$$

Third, quantize the vector D into finite l levels, where $0 < l < 256$ (ASCII range), to create symbolic sequences that suitable for text processing algorithm. Practically, ASCII characters below 32 are the control code that can affect the text

processing library in order to avoid that we suggest to shift the symbol to the range of the normal alphabet, where the first character is arbitrary c as described in equation (18) where sequence Q is a quantized vector.

$$Q = \lfloor D \times l \rfloor + c \quad (18)$$

Finally, a signature string Q represents a video clip. Then Q is chopped and each piece is called candidate gram. In our experiments only gram that has frequency greater than 2 can be stored in database as signature (S).

4.2. N-gram signature matching

To match querying signature, we assume that if some parts of query are matched with some parts of video in database then the query is not an original video. We use signature generation method described in the previous section to create a signature of query. A query signature is still in the same format as of video archive, which is a set of n-grams associated with its frequencies. Actually, if an n-gram of the query is matched to the one in database, this query could assume to be video duplication; however, this brings low-recall results. We propose to use a selected set of k highest-frequent n-grams to detect the similarity. Our algorithm is described as follows and written in pseudo code shown in Algorithm 5.

Algorithm 5 Video query

Input: k , query signature(S_r), signature database(S)
Output: similar video(S_d)

Step 1: S_r = Select top k **grams** from S_r order by frequency, length(**gram**)
 Step 2: FOR each *gram* of S_r
 Step 3: $S_d = S_d +$ Select videos from S where **grams** = *gram*
 Step 4: ENDFOR

First, generate n-grams of query video (S_t) by process in previous section. Second, select the k most-frequent n-gram where length of n-gram is priority if frequency is equal. This k n-gram is denoted as S_r . Third, every element in S_r is used to search in database S using binary search. If a single n-gram in S_r is found in S , the result is a set of matched videos.

5. Video indexing based on Spectrogram

We proposed a new compact signature, which is generated by transforming time-domain signature into frequency-domain. The overview of the proposed model is shown in Figure 34. The proposed model is composed of two parts: signature generation and signature similarity measurement.

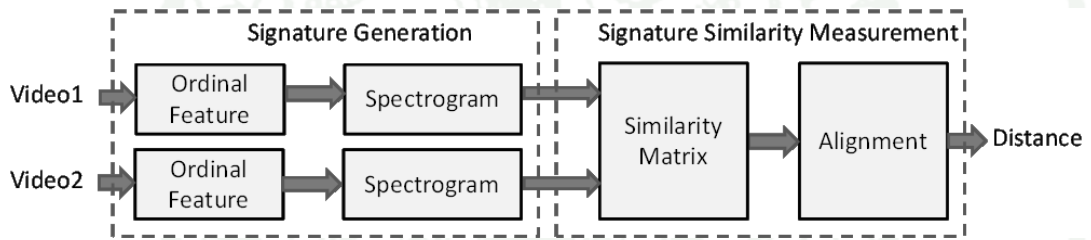


Figure 34 Proposed model for video indexing based on spectrogram

5.1. Spectrogram signature generation

In this research we focus on creating video signature that is robust to color deteriorate, intensity deteriorates and different frame rate. Therefore, ordinal feature (Lienhart *et al.*, 1997; Sánchez *et al.*, 1999; Xian-Sheng *et al.*, 2004) is used since it can solve these problems.

The ordinal feature is extracted from a single video frame by partition image into k sub-image (Figure 35A). Then, average intensity of each sub-image are calculated (Figure 35B) and assigned an ordinal rank (Figure 35C). This process is similar to (Xian-Sheng *et al.*, 2004). We re-arrange ordinal rank matrix into a vector v (Fig. 2D). This process is repeated through all video frames and ordinal feature

from every frame are concatenated to form the matrix M sized $k * l$ where l is a number of video frames.

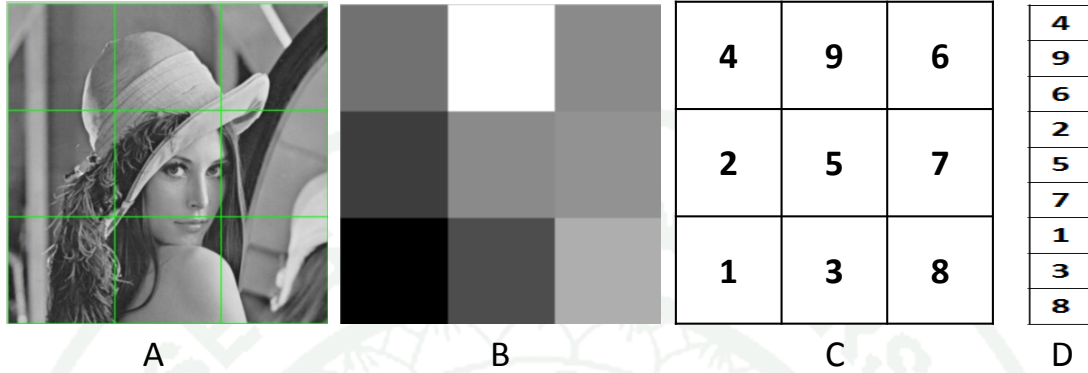


Figure 35 The ordinal feature extraction process

Considering row-wise vectors of matrix M , there are k vectors $x_1[n] \dots x_k[n]$ from each row from row 1 to row k . Each $x[n]$ is then used to generate spectrograms in order to reduce dimensions of signatures by transforming them into frequency domain. In order to preserve the time resolution a short-time Fourier transform: STFT is used. The STFT is a Fourier-related transform that is used to determine phase and frequency content of local section of a signal. The STFT is described in equation (19) where $x[n]$ is a signal to be transformed, $\omega[n]$ is window function. In this research “Hann window” is used and $X(m, \omega)$ is the Fourier transform of $x[n]\omega[n - m]$.

$$STFT\{x[n]\} \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n - m]e^{-j\omega n} \quad (19)$$

The STFT signal is used to create time-frequency distributions called spectrogram (Allen and Rabiner, 1977). It is widely used to visualize audio and speech signals. Spectrogram is estimated by computing the squared magnitude of the STFT of the signal using equation (20); the spectrogram is then used as a video signature.

By converting signal from time-domain to spectrogram, width of signal is increased from 1 into around $\frac{N}{2}$; however, length of signal is varied to window function (ω) and overlap size ($OverlapSize$). The length of this signature (l') is estimated by equation (21). In this research, we assume that $WindowSize \approx N$. Then, size of spectrogram is around $[\frac{N}{2}l']$.

$$spectrogram\{x[n]\}(m, \omega) = |X(m, \omega)|^2 \quad (20)$$

$$l' = \left\lfloor \frac{l - WindowSize}{WindowSize - OverlapSize} \right\rfloor + 1 \quad (21)$$

An example $x_i[n]$, which is a dimension original signal is shown in Figure 36(A) and its spectrogram shown in Figure 36(B). Then, we have a video signature which is composed of k spectrograms, where its length is shorter than time-domain signature.

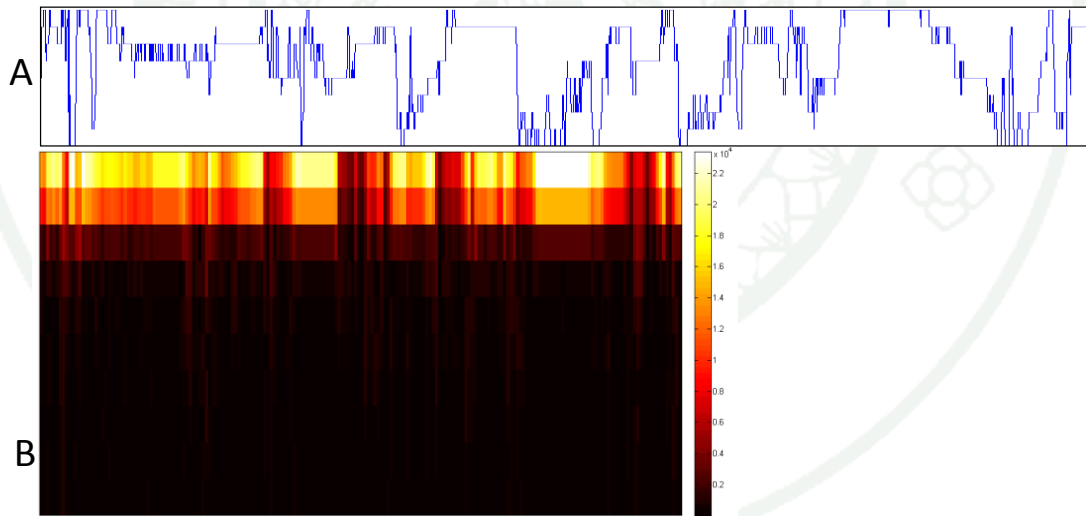


Figure 36 A dimension of spectrogram signature example

5.2. Spectrogram signature comparison

To measure similarity between two spectrogram signatures $D1$ and $D2$, we adopted a spectrogram-based method (Turetsky and Ellis, 2003) for audio analysis application, which can handle signatures with different lengths. First, similarity of pair k ($D1_k$ and $D2_k$) between two spectrograms are calculated by equation (22).

$$SM_k(i, j) = \frac{D1_k(i)^T D2_k(j)}{|D1_k(i)| |D2_k(j)|}, \text{ for } 0 \leq i < l'_1, 0 \leq j < l'_2 \quad (22)$$

Next, signatures from each dimension is aligned by Algorithm 6 in order to acquire minimum cost. Costs from all dimensions are summed by equation (23) as a distance between two videos. The design of Algorithm 6 is basically Dynamic Programming algorithm, where its alignment rule (Line 11) is from greedy penalty rules (Cao and Zhu, 2009; Turetsky and Ellis, 2003).

$$d = \sum_{i=1}^k \text{DynamicTimeWarping}(1 - SM_i, i) \quad (23)$$

Examples of the lowest cost path determined by Algorithm 6 are shown in Figure 37. The lowest cost path of exactly duplicated videos is shown a perfect diagonal line. However, the more different videos, the longer cost path is. In case of two signals $x_1[n]$ and $x_2[n]$, whose lengths are l_1 and l_2 , respectively, the complexity of dynamic time warping algorithm is $O(l_1 l_2)$ (Keogh and Pazzani, 2000). Algorithm 6 performs faster by reducing size of input vectors. First, it transforms those two input signals into spectrograms $|X_1(m, \omega)|^2$ and $|X_2(m, \omega)|^2$, where their sizes are only $[\frac{N}{2} l'_1]$ and $[\frac{N}{2} l'_2]$, respectively. Then, size of similarity matrix SM is $[l'_1 l'_2]$. Therefore, Dynamic time warping complexity of this algorithm is $O(l'_1 l'_2)$. Thus, the speedup is $O(l_1 l_2 / l'_1 l'_2)$.

Algorithm 6 Dynamic Time Warping**Input:** similarity matrix(SM, k)**Output:** minimum cost path (d)Step 1: $[r \ c] = \text{size}(SM)$ // matrix dimensionsStep 2: $D(1, :) = \infty, D(:, 1) = \infty$ Step 3: $D(1,1)=0$ Step 4: **FOR** $i=1$ **to** r Step 5: **FOR** $j=1$ **to** c Step 6: $D(i+1,j+1)=SM(i,j)$ Step 7: **ENDFOR**Step 8: **ENDFOR**Step 9: **FOR** $i=1$ **to** r Step 10: **FOR** $j=1$ **to** c Step 11: $D(i+1,j+1)=D(i+1,j+1)$
 $+\min(D(i,j), D(i,j+1), D(i+1,j))$ Step 12: **END FOR**Step 13: **END FOR**Step 14: **RETURN** $D(i+1,j+1)$

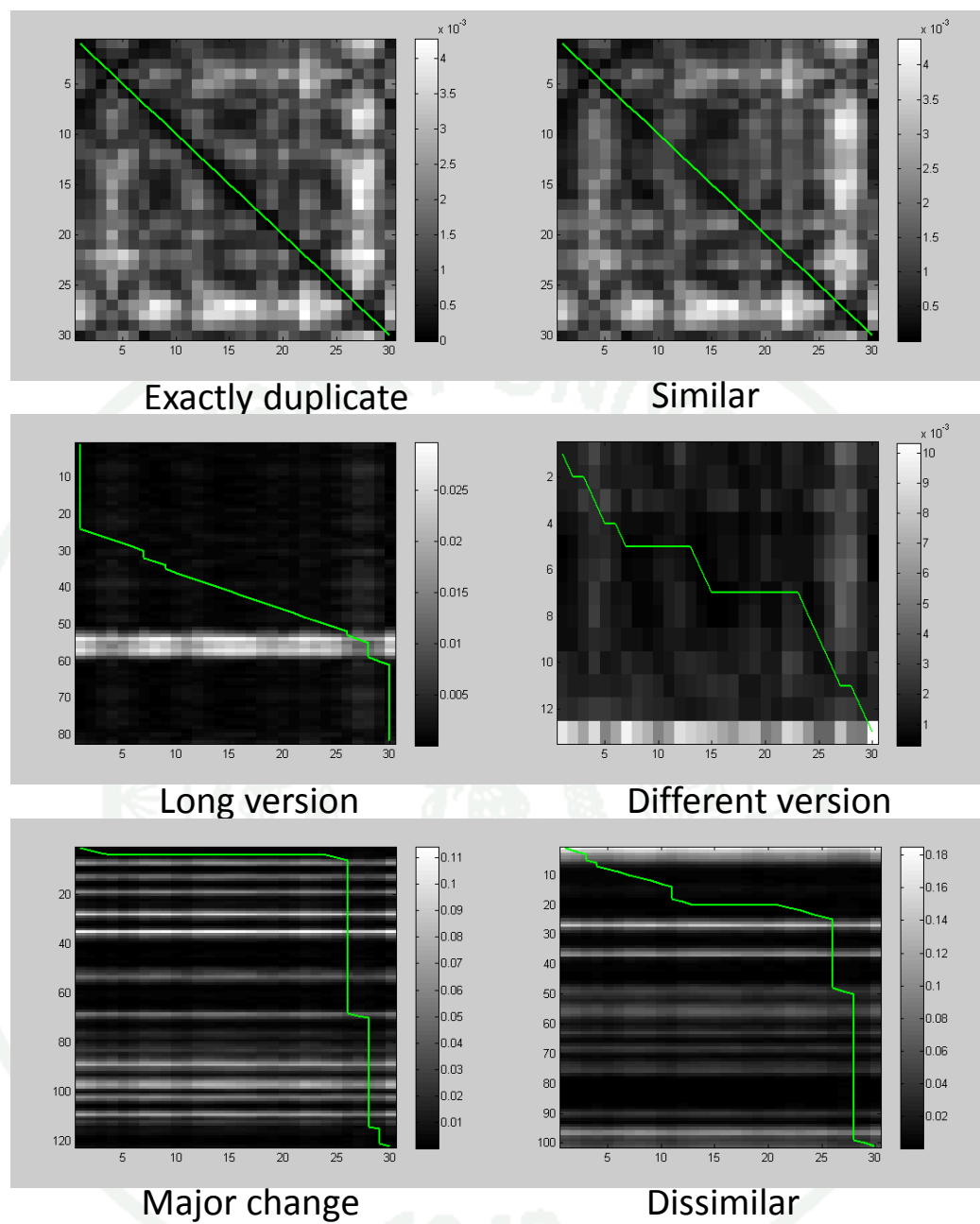


Figure 37 The minimum cost path of different type of video comparison.

RESULTS AND DISCUSSION

Performance Evaluation Methods

In this section, the evaluation of image cropping algorithm, video cropping algorithm and video indexing algorithm are explained. Dataset used in these experiments, test scenario and parameter tuning also described in this section.

1. Image cropping algorithm evaluation

Our proposed algorithm is evaluated by comparing satisfaction score. Cropped results of our proposed algorithm are compared with results from all three cropping algorithms of Picasa 3.

Dataset in the experiment is composed of 100 images randomly downloaded from the National Geographic's "Photo of the day" website (Geographic, 2013) whose reputation is linked to high quality photography, selects only very high quality images for its Photo of the Day website. By observation, there are varieties of image shooting such as macro shot, close-up shot, medium shot, and long shot. Images in dataset have various sizes ranged from 372x745 to 553x732 pixels (portrait mode) and 470x325 to 1024x750pixels (landscape mode).

There are two important parameters required by this algorithm: zoom rate (d) and information-loss threshold (t). By preliminary experiment, we recommend $d > 0.9$ and $t=0.04$; otherwise, recall rate of the system will drop rapidly. In the experiments, d is set as 0.959 and t is set as 0.04 ($t=0.04$). If application does not require large magnification ratio, or it needs only changing aspect ratio we recommend using a small value of t .

The face detection was implemented using OpenCV version 0.9.7.1. Importance map generation and automatic-zoom algorithm was implemented using

Matlab. Weights of face map (w_1) and focus map (w_2) are set as 0.5. To maximize possible loss and prevent the effect of rotation in some algorithms, the value of target aspect ratio is set as 1:1 ($\theta=1$). The equation 3 was solved by genetic algorithm module in Matlab's optimization toolbox (optimtool). Both variables x' and y' are encoded as floating-point chromosomes and processed by uniform mutation function with mutation rate at 0.02 and scattered crossover function with crossover rate is set at 0.8. Population size was limited to 20. The algorithm stops when fitness change is less than 10^{-0} . The initial populations are located around the top-left of the image ($x' \approx 0$ and $y' \approx 0$).

In Figure 38, the experimental results show that this algorithm can solve the optimization within 50 generations. That also shows the best fitness value and average fitness values of every population are rapidly converging on the stable state. By calculation, it is 70 times as fast as a brute-force method, when window size is determined by using iterative pan-and-scan method at different zoom factor.

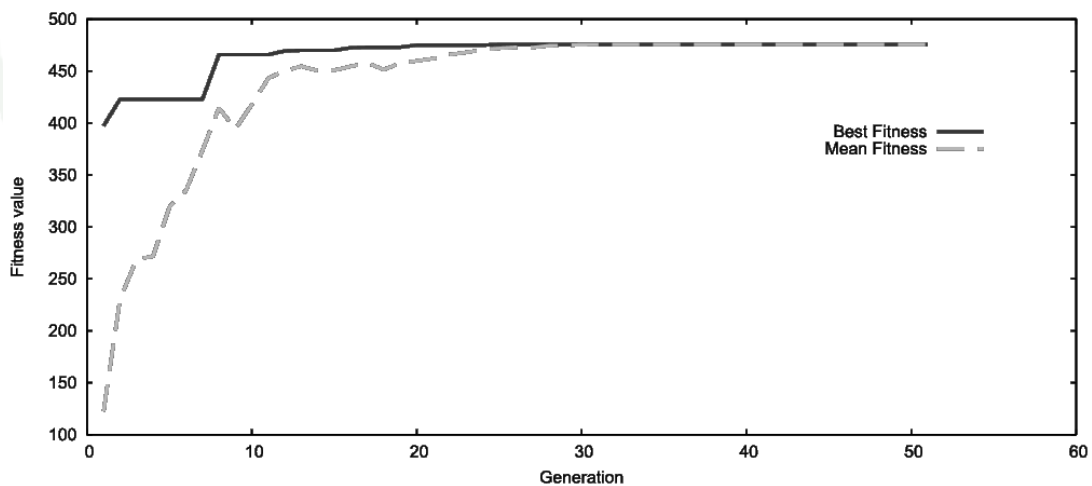


Figure 38 Best fitness and mean fitness score of each generation while solving one zoom iteration.

The reference results are prepared using Picasa 3(version 3.9.0) (Google, 2013b), a Google's photograph management application. It has a function for semi-automatic image cropping. It has been done by proposing three cropped versions from three classified algorithms as shown in Figure 39. User must select one of three algorithms or crop the picture manually. We use all three cropped images generated by Picasa to compare with results from this algorithm.

However, from preliminary experiment, we found that if the data loss in cropping process is less than 20%, differences between results from any methods are hard to be noticed. In order to make results easy to compare, we forced every algorithm to drop large amounts of data by using 1:1 as target aspect ratio. We also can ignore a situation that any algorithm may avoid data loss by rotating aspect ratio.

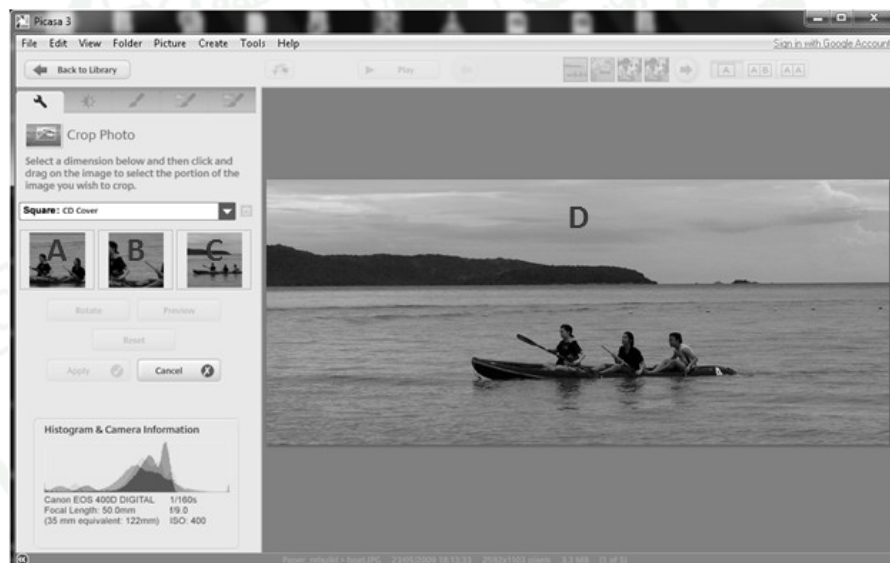


Figure 39 The interface of Google Picasa. 3 cropped versions (A, B, and C) and original image (D)

To evaluate performance of each algorithm, we develop a questionnaire website to collect opinions. The questionnaire was designed using a four-level Likert scale (Reject, Poor, Good, and Excellent). All images in the dataset were used to evaluate qualitative results of our proposed algorithm by comparing with result of three Picasa algorithms. An example of questionnaire is shown in Figure 40.

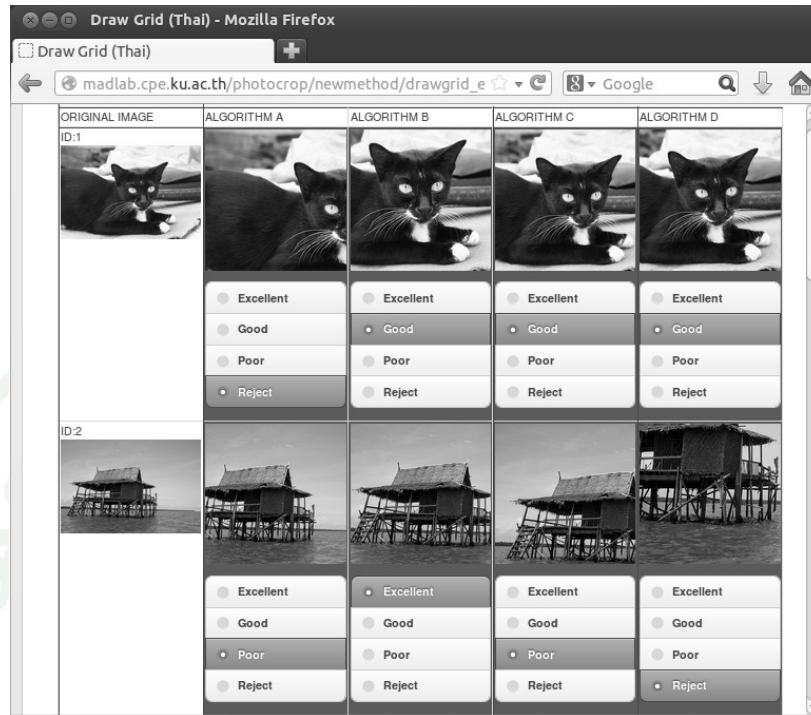


Figure 40 User interface of questionnaire website for qualitative evaluations.

2. Video cropping evaluation

In this section, we demonstrate the results of our algorithm of automatic pan-and-scan. Then, we evaluate our proposed video cropping model by comparing user satisfaction of the outputs with other two methods. First, compare this algorithm with center cropping that represents the cropping method can be found in most traditional television set and multimedia player software and manual method. Second, compare it with the manual cropped by expert.

There are two parameters in the shot-type classifier algorithm that we described earlier in the section 3.2 for which we use $\theta_1 = 0.2$ and $\theta_2 = 0.7$. Due to the film production process, the image may be shaken 0.1% vertically and 0.15% horizontally by the camera mechanism and film-to-video transfer process (SMPTE, 1994, 2003). These values are fixed through all our experiments.

There are two major cropping scenarios: target aspect ratio greater than the source's, and vice versa, which are shown in Figure 41. Figure 41A shows the original video frame displayed in letterbox mode for a target aspect ratio less than source aspect ratio; Figure 41B shows the same source image cropped to match the target aspect ratio by a pan-and-scan mode done by our algorithm. In the other scenario, when the source video frame has smaller aspect ratio lesser than target's, Figure 41C shows source video display in tinderbox mode and a cropped version in tilt scan mode (pan in vertical direction mode) done by our algorithm is shown in Figure 41D. These are examples of cropped results. Our results indicate that the image are cropped to fit the target aspect ratio while the important content is preserved, in contrast with letterbox and tinderbox versions that produce a small image surrounded by black bars.



Figure 41 Example of video cropping result of scenario source aspect ratio $>$ target aspect ratio (A,B) and source aspect ratio $<$ target aspect ratio (C,D)

2.1. Compare Video cropping algorithm with traditional method

Comparing with traditional method, we generate different cropped versions of many movies at aspect ratios of popular cell phones by different algorithms. We developed questionnaire software for user evaluation. Volunteers are asked to compare the original widescreen video with multiple cropped versions at the same target aspect ratio, simultaneously. We had 160 volunteers to evaluate the results.

Aspect ratio of interest is selected from the availability of screen size of popular mobile phones in current market. We are also interested in portrait mode of each resolution because it is always an alternative to the user on the same device. Table 7 shows aspect ratio of interest and the amount of data loss during aspect-ratio adaptation and illustrated of display aspect ratio is shown in Figure 43. Figure 42 demonstrate results cropped from landscape and portrait aspect ratios, which we can see that in landscape mode important content was easy to watch in cropped version Figure 42B, by contrast with untouched version in Figure 42A the image was shrink to fit display and result was difficult to watch. Portrait mode is also more difficult to watch for the untouched version Figure 42C but the importance object was emphasize in cropped version Figure 42D



Figure 42 Landscape mode: (A) letterbox method; (B) cropped method, portrait mode: (C) letterbox method; (D) cropped method.

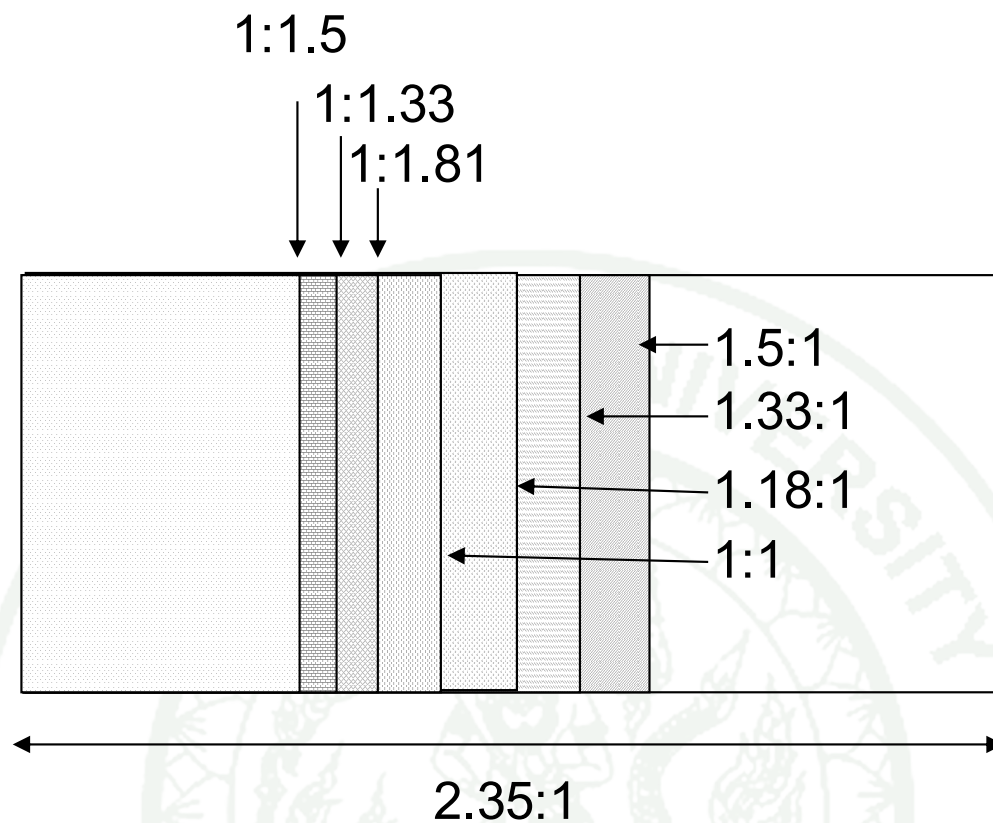


Figure 43 The illustrated of display aspect ratio used in this experiments

Table 7 List of displays used for algorithm evaluation

<i>Resolution</i>	<i>Aspect ratio</i>	<i>% Data loss</i>
480x320(Apple.com, 2012)	1.5 (1.5:1)	36.17
1024x768(lg.com, 2012)	1.33 (1.33:1)	43.40
416x352(Gsmarena.com, 2012a)	1.18 (1.18:1)	49.79
320x320(samsung.com, 2013b)	1.0 (1:1)	57.45
416x352 (Portrait mode)(Gsmarena.com, 2012a)	0.84 (1:1.18)	64.26
1024x768 (Portrait mode)(lg.com, 2012)	0.85 (1:1.33)	68.04
480x320 (Portrait mode)(Apple.com, 2012)	0.66 (1:1.5)	71.66

We asked 160 evaluators to participate in the experiments, and assigned them randomly to the experiments. 66% of the evaluators were male and 34% female. 25% of evaluators have background knowledge in photography and photo composition. The age of the evaluators ranged between 20 and 50; however, majority of them were between 21-25 years old.

The source data of our experiments are 2.35:1 widescreen video. Data are from De Lorentis's Manhunter (IMDb, 2010). We selected 30 non-overlapped clips from random part of video. Each clip is 40 seconds long which is contained 4-6 shots and its frame resolution is 720x304 pixels.

We developed software for qualitative evaluation, which has user interface shown in Figure 44. Sizes and aspect ratios of display of experimental platform are unchangeable. The evaluation was conducted in the controlled-environment recommended by the ITU-R BT.500-13 recommendation(Union, 2012). Each user has to fill their demographic information first and then they have to watch four movie clips concurrently on the same screen. One of those movies is the original widescreen version shown in letterbox version on the top left area and the others are results from different cropping algorithms. After the users watch the movie, they have to evaluate each cropped video as “reject”, “acceptable” or “excellent”. Each user has to do this multiple times for different movies and different target aspect ratios. The experiment is blind test; the positions of cropped video are always shuffled for each clip. A dummy cropping algorithm, which can crop videos at random location, is used to generate placebo. User also did not know which algorithm has been applied for each cropping clip.



Figure 44 User interface of the algorithm evaluation

2.2. Compare Video cropping algorithm result with ground truth

This part of the experiment is to compare the qualitative results between our proposed algorithms with video clips cropped by experts. We use video cropped by expert as ground truth. Usually, a motion picture is released in many different media formats. The DVD and Blu-ray versions are always produced in widescreen or letterbox version. VCD and VHS are mostly produced in a 1.33:1 cropped version. In this experiment, we deployed our automatic pan-and-scan algorithm onto an anamorphic widescreen version from DVD in order to create a cropped version at the same aspect ratio of VCD and then asked volunteers to compare the quality with the commercial VCD or VHS of the same motion picture.

There are two parts of experiments: accuracy and quality evaluations. For accuracy evaluation, we assume that the results from expert is the ground truth and then we measure the precision to indicate how much different between our result and the ground truth. For quality evaluation, we assume the results from VCD are just

ones of cropped versions. Then, we asked volunteers to compare its quality with our results. Note that, there is only one aspect-ratio adaptation configuration, which is from 2.35:1 to 4:3.

Ground truth is prepared by comparing a VCD frame and DVD frame to determine the pan offset (Figure 45). Both videos are carefully synchronized and scale to the same height as intermediate videos. Figure 46 shows the pan offset that produced VCD version from original DVD source. The details of the source video from DVD and VCD are shown in Table 8. DVD intermediate is generated by reshaping aspect ratio from 1.25:1 to 2.35:1 using Bicubic interpolation (Figure 48), because DVD squeeze 2.35:1 to fit on its native 720x576 resolution. The scan mode is change to progressive scan mode using Bob de-interlacing algorithm to eliminate artifacts, the example of non-de-interlace fame and de-interlace frame using bob algorithm is shown in Figure 48. The VCD intermediate also generated in similar way. The offsets are calculated from intermediate according to equation (24) where I_{vcd} and I_{dvd} are exact same frame taken from VCD and DVD image frame, w is width and h is height of frame.

The distance function used in this experiment need to be robust to video encoded in different format and resolution. In this experiment we used ordinal feature to measure the distance between two frames. The ordinal feature is extracted from a single video frame by partition image into k sub-image. Then, average intensity of each sub-image are calculated and assigned an ordinal rank. Ordinal rank is used as image signature, then both signatures are compared using sum of absolute differences algorithm (SAD)(Hamzah *et al.*, 2010). The SAD algorithm is shown in equation (24) and the process of extracting ordinal signature shows in Figure 47.

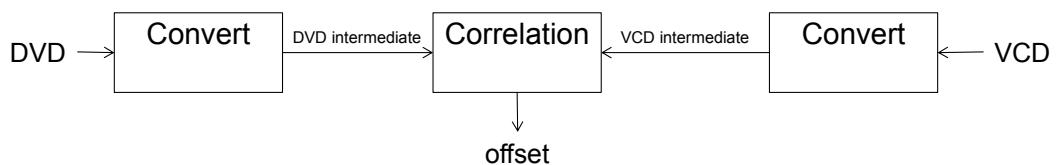
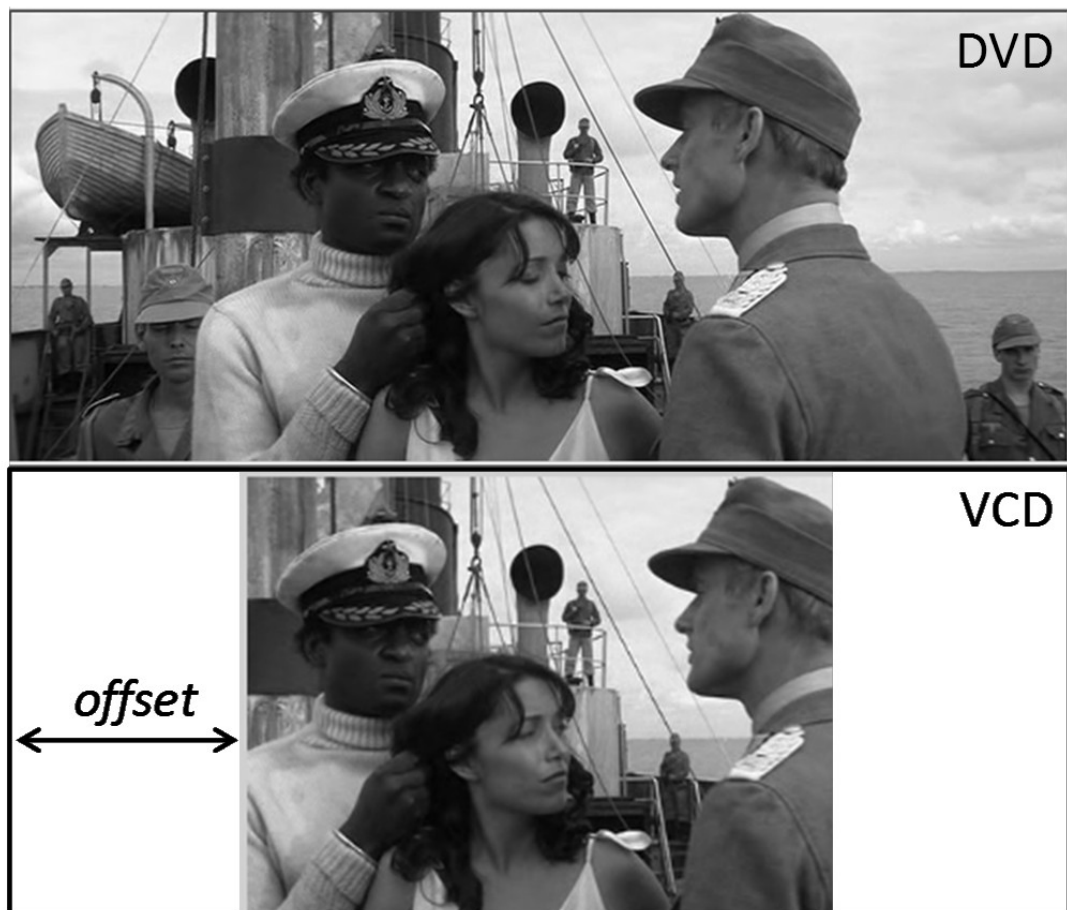


Figure 45 Ground truth generation from VCD

Table 8 Detail of video coding used in this experiment

	<i>DVD</i>	<i>VCD</i>	<i>DVD</i> <i>Intermediate</i>	<i>VCD</i> <i>Intermediate</i>
Video codec	MPEG2	MPEG1	Xvid MPEG4	Xvid MPEG4
Resolution	720x576	352x288	640x272	384x288
Frame rate	25	25	23.976	23.976
Aspect ratio	1.25:1	1.22:1	2.35:1	4:03
Scan mode	Interlace	Interlace	Progressive	Progressive

$$offset = \underset{k}{\text{ArgMin}} \sum_{i=1}^{w_{dvd}-w_{vcd}} \sum_{j=1}^h \text{distance}(I_{vcd}(i+k, j), I_{dvd}(i, j)) \quad (24)$$

**Figure 46** Pan offset used to generated VCD version from DVD

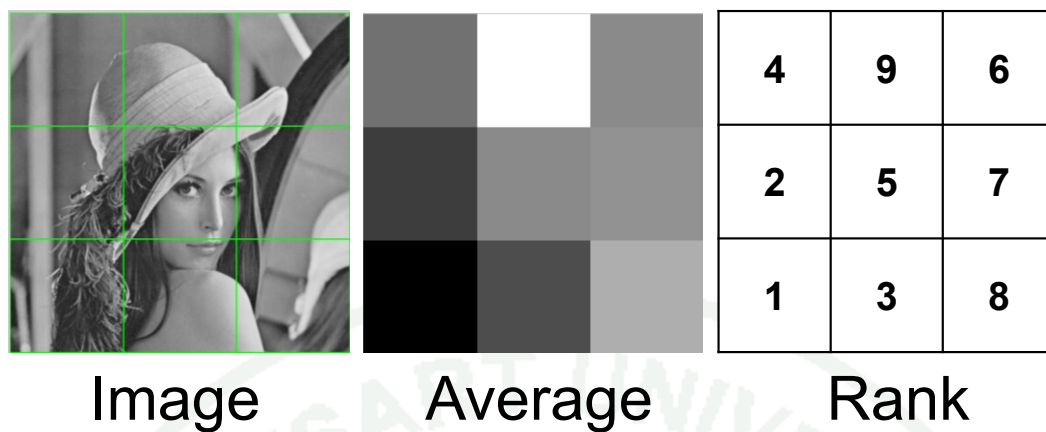


Figure 47 Ordinal feature extraction process for signature generation



Figure 48 Video frame before de-interlace (A) and after de-interlace (B)

Since DVD is encoded in MPEG2 format and VCD is encoded in MPEG1 format. This can result in error of key frame location; every first and last frame of every shot often have different key frame. The error is shown in Figure 49, to avoid this type of error the filter described in equation (25) is used. The example of pan-and-scan offset before and after applied with this filter is shown in Figure 50.

DVD	1	1	1	1	1	1	2	2	2	2	3	4	4	4	4	5	5	5	5	6	6	6	7	7	7
VCD	1	1	1	1	1	1	2	2	3	3	4	4	4	4	4	5	5	6	6	6	6	7	7	7	
ERR							X			X						X			X						

Figure 49 Different frame sequencing shot found in DVD and DVD

$$offset(i) = \begin{cases} offset(i) & ; |offset(i) - offset(i+1)| < \theta \\ drop\ i & ; others \end{cases} \quad (25)$$

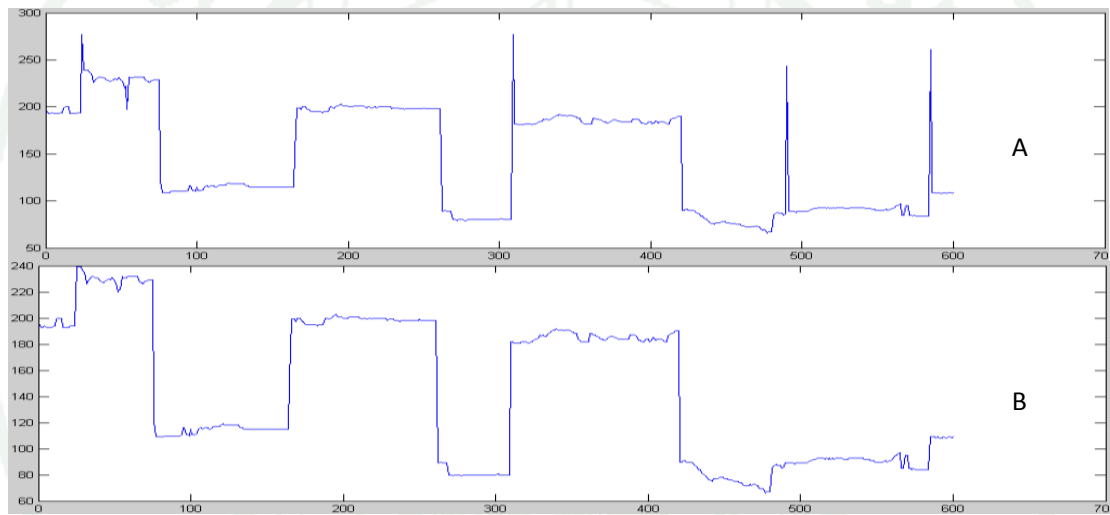


Figure 50 Pan-and-scan Offset signal before filter (A) and after filtered (B).

2.2.1. Performance metric

We evaluate performance of our algorithm by measuring precision of result from proposed algorithm based on the result cropped by experts. The definition of precision is defined by Equation (26).

$$Precision = \left(\frac{AlgorithmWindow \cap GroundtruthWindow}{GroundtruthWindow} \right) \quad (26)$$

2.2.2. Dataset

We selected many shots from the following movies in our experiment: Transformers: Revenge of the Fallen (Bay, 2009), Raiders of the Lost Ark(Spielberg, 1981), Indiana Jones and the Temple of Doom(Spielberg, 1984), Indiana Jones and the Last Crusade(Spielberg, 1989), Mercury Rising(Becker, 1998), and Star Trek(Abrams, 2009). Note that we have to avoid any fake-widescreen since it is designed mainly for center cropping. We can find the list of fake-widescreen movies at Internet Movie Database Ltd. (IMDb) website in the technical specifications section (Figure 51).Figure 52 shows example frame of fake-widescreen, this type of cinematic processed cannot be used in this experiment because movie was original shot in 4:3 format and later cropped into both 4:3 (green box in) and widescreen (red box in), it is not a pan-and-scan processed. The ground truth cannot be generated because the original shot frame as not release. Figure 53 shows exact same frame of true-widescreen movie, we can use this type of cinematography because the 4:3 released was a pan-and-scan version of 2.35:1 release.



Figure 51 Example of the exact same frame of 2.35:1 and 4:3 release of fake-widescreen movie

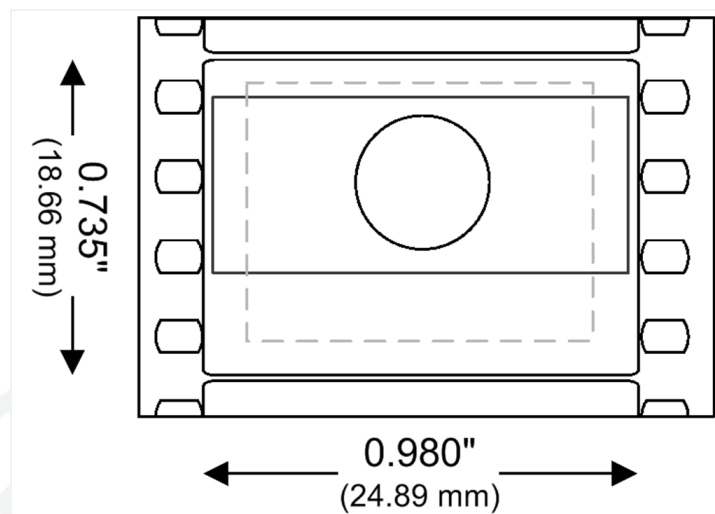


Figure 52 Film areas used by face-widescreen format

<p>2.35:1 version</p>	<p>Technical specifications for Raiders of the Lost Ark (1981)</p> <p>Camera Panavision Cameras and Lenses</p> <p>Laboratory Metrocolor, USA</p> <p>Film length (metres) 3129 m (Sweden, cut version) 3160 m (Sweden, uncut version)</p> <p>Film negative format (mm/video inches) 35 mm (Eastman 100T 5247)</p> <p>Cinematographic process Panavision (anamorphic)</p> <p>Printed film format 16 mm 35 mm 70 mm (blow-up)</p> <p>Aspect ratio 2.20 : 1 (70 mm prints) 2.35 : 1</p>
<p>4:3 version</p>	

Figure 53 Example of the exact same frame of 2.35:1 and 4:3 release of true widescreen movie

3. Video indexing based on N-gram evaluation

The experiments are divided into two parts: to determine the most appropriate number of quantization levels (l) and number of grams in video query (l); and compare our method with a baseline algorithm (Dong *et al.*, 2008; Wan-Lei *et al.*, 2010). We assume that our method should maintain accuracy where using less computation power. We used the well-known CC_WEB_VIDEO, which is a near-duplicate web video dataset. This dataset contains a collection of viral videos which can be searched by 24 popular queries from YouTube, Google Video and Yahoo!. All 12,790 videos are varied in encoding format, frame rate, bit rate, frame resolution, color / lighting change, overlay with logo and text and content modification. All 24 queries of the dataset are shown in Table 9 (Wan-Lei *et al.*, 2010; Wu *et al.*, 2007; Xiao *et al.*, 2009).

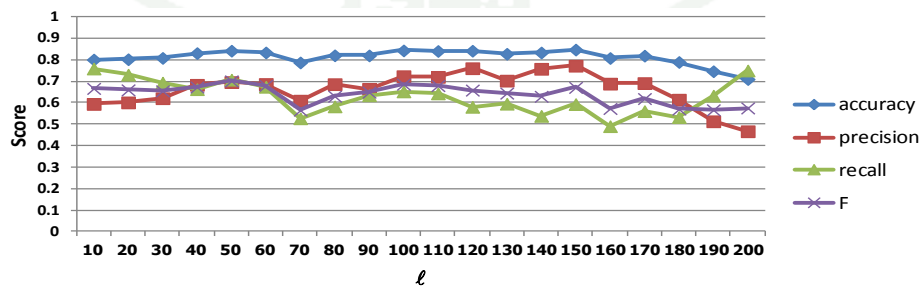
Table 9 Queries to build video collections

ID	Query	Number of Videos		Processing Time (Second)
		Total	Near-Duplicate	
1	The lion sleeps tonight	792	334	1.88
2	Evolution of dance	483	122	2.85
3	Fold shirt	436	183	0.39
4	Cat massage	344	161	0.21
5	Ok go here it goes again	396	89	1.87
6	Urban ninja	771	45	3.39
7	Real life Simpsons	365	154	1.07
8	Free hugs	539	37	2.89
9	Where the hell is Matt	235	23	0.84
10	U2 and green day	297	52	1.30
11	Little superstar	377	59	1.49
12	Napoleon dynamite dance	881	146	2.94
13	I will survive Jesus	416	384	0.38
14	Ronaldinho ping pong	107	72	0.65
15	White and Nerdy	1771	696	7.52
16	Korean karaoke	205	20	1.11
17	Panic at the disco I write sins not tragedies	647	201	2.71

Table 9 Queries to build video collections (Continued)

ID	Query	Number of Videos		Processing Time (Second)
		Total	Near-Duplicate	
18	Bus uncle (巴士阿叔)	488	80	1.70
19	Sony Bravia	566	202	1.42
20	Changes Tupac	194	72	1.15
21	Afternoon delight	449	54	1.45
22	Numa Gary	422	32	1.59
23	Shakira hips don't lie	1322	234	5.98
24	India driving	287	26	0.88
Sum		12790	3478	47.66

To determine a number of quantization level l , we measure the performance (accuracy, precision, recall, and F-measures) of different l values, ranged from 0 to 200. the details of this effect are shown in Figure 54. We found that effect of l can be divided into three regions. The first region is where $l < 80$. In this region, the result has high recall but low precision and accuracy. The second region is where between $80 \leq l \leq 160$. In this region, the accuracy is stable but the precision and recall are highly varied. Value of l in this region can be selected depending on type of an application whether it required either precision or recall. The third region is where $l > 160$. In this region, the accuracy and precision are decreased where l is increased. From the experiment, we consider the best value of parameter $l=150$ because it produced the highest accuracy and the highest precision. Then, we will use this parameter for the rest of experiments.

**Figure 54** Accuracy, precision, recall and F with different number of quantization levels (l)

To determine the most appropriate number of n-grams (k), we varied k from 2 to 70. The results are shown in Figure 55. The result can be divided into three regions. The first region is where $k < 7$. We found that accuracy and precision are high but recall is low. Therefore, value of k in this region is appropriate for applications that require high precision. The second region is where $7 \leq k \leq 17$. In this region, accuracy is stable but precision and recall are converged. Therefore, this could be an optimal zone depending on types of applications. The third region is where $k > 17$. In this region, accuracy, precision and recall are stable. Therefore, $k > 17$ should not be used because the result will not get better but computation time is however the growth of computation time is linear, details of query time at different number of n-gram show in Figure 56. From this experiment, we consider the best value of parameter $k = 8$ because it produces the highest accuracy while maintaining minimum computation and database load.

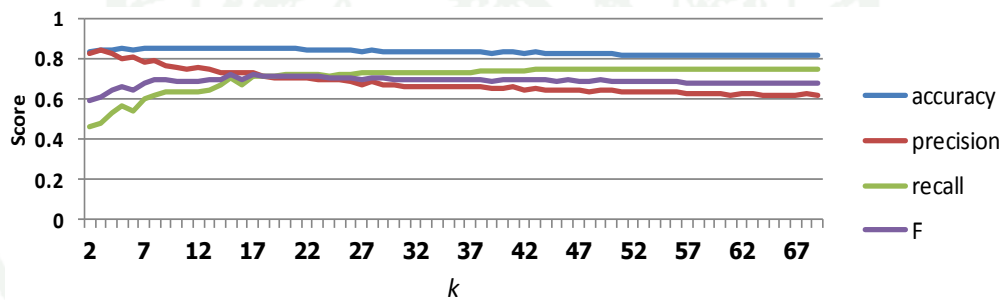


Figure 55 Accuracy, precision, recall and F at different number of n-gram in query

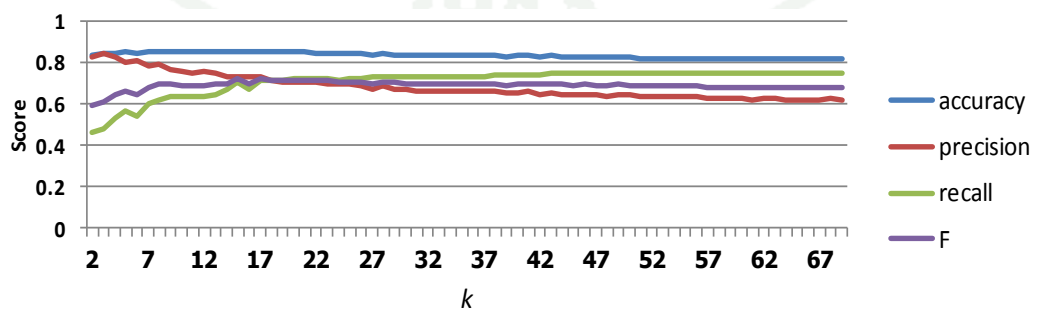


Figure 56 Query time at different number of n-gram over all queries.

4. Video indexing based on spectrogram evaluation

In this research, we use a benchmark dataset called CC_WEB_VIDEO: Near-Duplicate Web Video Dataset. This dataset contained a collection of viral video based on a sample of 24 popular queries from YouTube, Google Video and Yahoo! Video. All 12,790 videos are in various encoding formats, frame rates, bit rates, frame resolutions, lengths, frame insertion / frame deletion, color / lighting change, overlay with logos and text, and content modification (Wan-Lei *et al.*, 2010; Wu *et al.*, 2007; Xiao *et al.*, 2009). Because each topic of the CC_WEB_VIDEO dataset has different number of duplicated videos; it is possible that result may be biased on number of duplications. Therefore, we normalize result before further evaluation. We have to do an empirical experiment in order to determine 1) the best dimension k for generating ordinal features 2) the best window size $\omega(n)$ for STFT; and 3) the best overlap size of STFT.

We determine a dimension size k of ordinal feature k in the signature generation process by varying k from 2x2 to 6x6 to find top-1 to top-20 ranking results (Figure 57). We concluded that the **3x3** generated the highest precision.

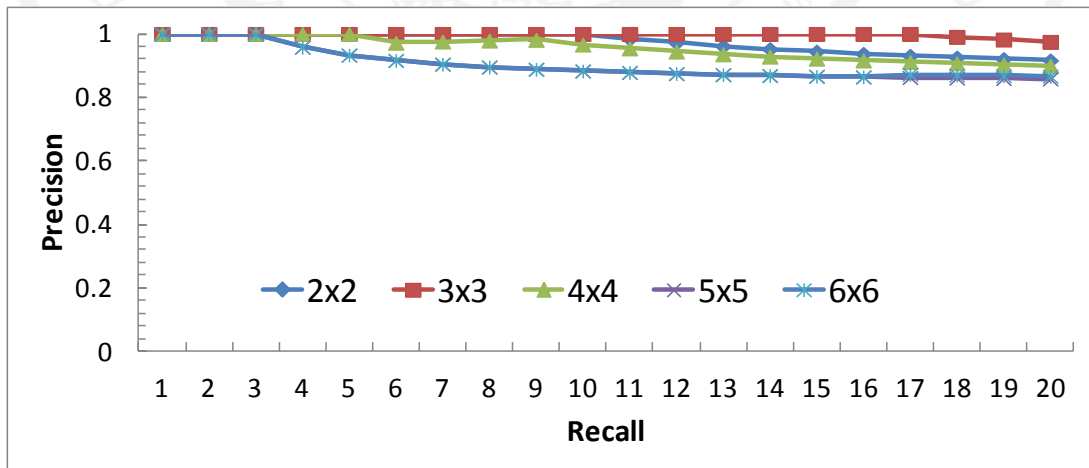


Figure 57 The accuracy at different grid size

A window size $\omega(n)$ is determined by fixing k as 3x3 and varying size from 50 to 10,000 frames. Precision of ranking (average of top-1 ranking to top-20 ranking) is shown in Figure 58. The parameter $\omega(n)$ also indicates time and frequency resolution of a spectrogram and signature length. Small window generates high resolution in time (long signature length) but low resolution in frequency and vice versa. From empirical results, we select **400** frames in further experiments.

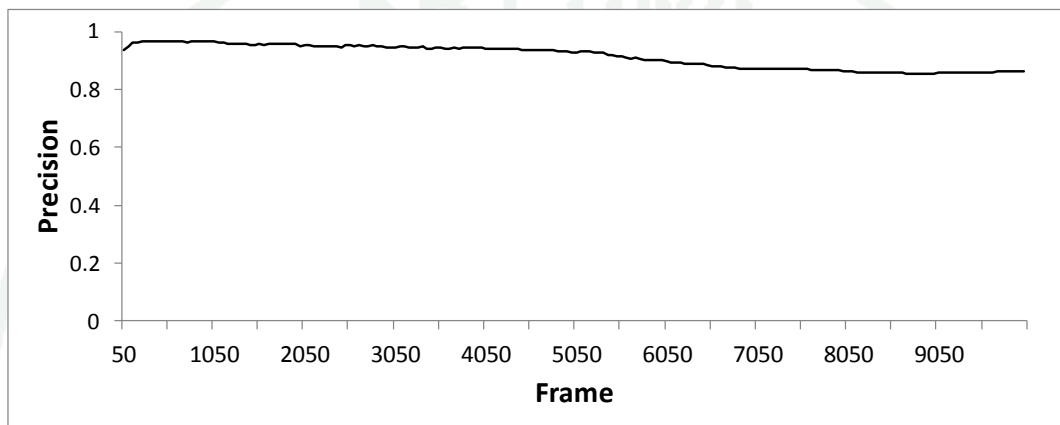


Figure 58 Precision at different STFT window size

An *OverlapSize* is determined by fixing size $\omega(n)=400$, k as 3x3 and varying this parameter from 1 to 90% of window size. Precision of ranking (average of top-1 ranking to top-20 ranking) is shown in Figure 59. The parameter *OverlapSize* indicates the overlap frame of STFT. Large overlap generates lower error rates but signature length is longer and vice versa. From empirical results, we select **81%** of window size in further experiments.

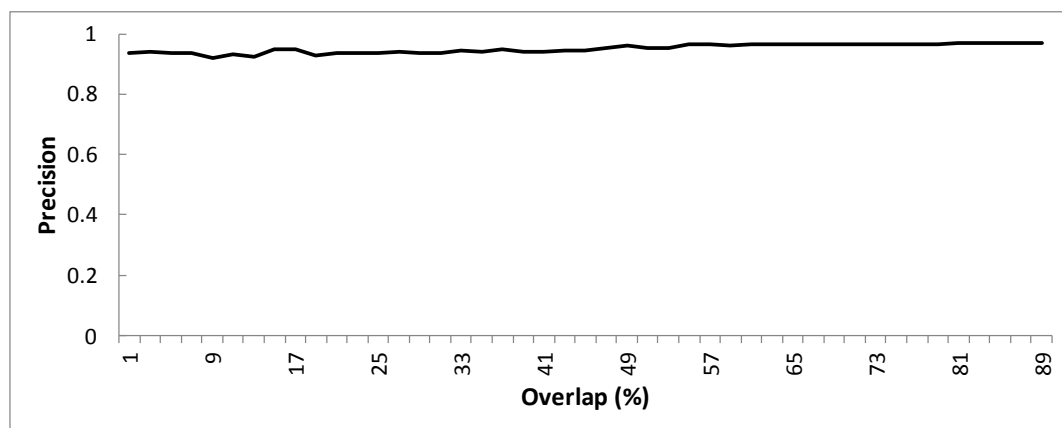


Figure 59 Overlap window

Results

1. Image cropping algorithm evaluation results

Qualitative evaluations are performed with 340 volunteers, which are a group of photographers and general Internet users. 80% of them are male, and another 20% are female. 59% have knowledge in photographic composition. Ages are between 10 and 60 years old and average age is 23.8 years old. Each volunteer is asked to evaluate cropped results from 20 original images. Totally, 3780 cases are evaluated. Results from a four-level Likert scale are converted to percentage and shown in Table 10 and they show that the user group preferred this algorithm 30% higher than Picasa 1, 23% higher than Picasa 2, and 6% higher than Picasa 3. The two-sample t-test confirmed that the mean differences of all results were significant ($p < 0.05$).

Table 10 Image cropping evaluation result

	This algorithm	Picasa 1	Picasa 2	Picasa 3
Excellent	27.38%	7.83%	11.66%	22.27%
Good	45.18%	23.43%	28.80%	42.30%
Poor	21.40%	32.67%	32.75%	24.39%
Reject	6.03%	36.05%	26.77%	11.03%
Average	64.67%	34.33%	41.33%	58.33%
Std.dev	28.39	31.75	32.63	30.73

However, different algorithms produced results with different losses of information since each algorithm can be configured to drop smaller amount of data in order to gain more user satisfaction. To make the comparison as fair as possible, we investigate the effect of image loss on each algorithm by analyzing the result of each method. The distribution of data loss of each algorithm is shown in Figure 60 Data loss distribution of four cropping algorithms.. User satisfactions of each method of all pictures and percentage of data loss are plotted on scatter chart in Figure 61. We found that Picasa 1 and Picasa 2 generated results by allowing huge loss of data; therefore, the cropped image is relatively small. Both Picasa 1 and Picasa 2 methods allow 60% loss of data on average. User satisfactions of these two algorithms are

41.6% and 34.3% (in between poor and reject). Picasa3, the best method among Picasa's cropping functions, generated results with lower loss allowance at 32% on average, where it is ranged from 24% to 57% and get the satisfaction score 58.6% (in between good and excellent). This algorithm generated results which the average loss is about 39% and ranged from 17% to almost 97%. Therefore, we can conclude that this algorithm dropped larger amount of image data and it still gained 6% more satisfaction than Picasa3 (and all other Picasa's).

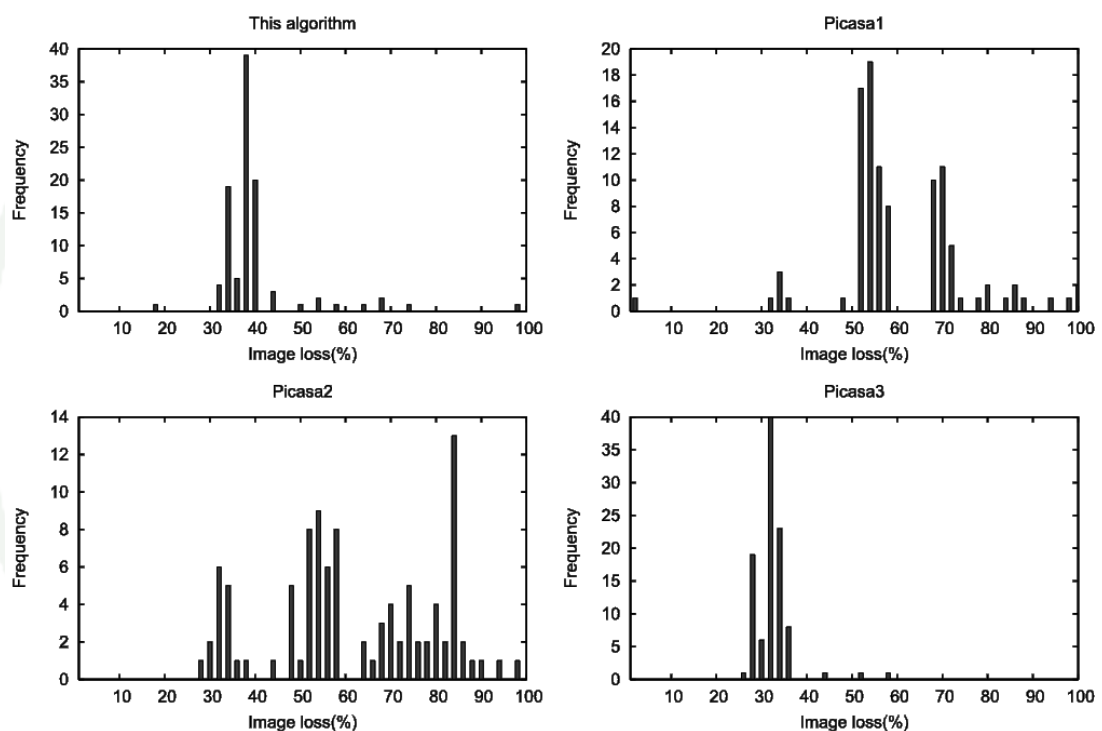


Figure 60 Data loss distribution of four cropping algorithms.

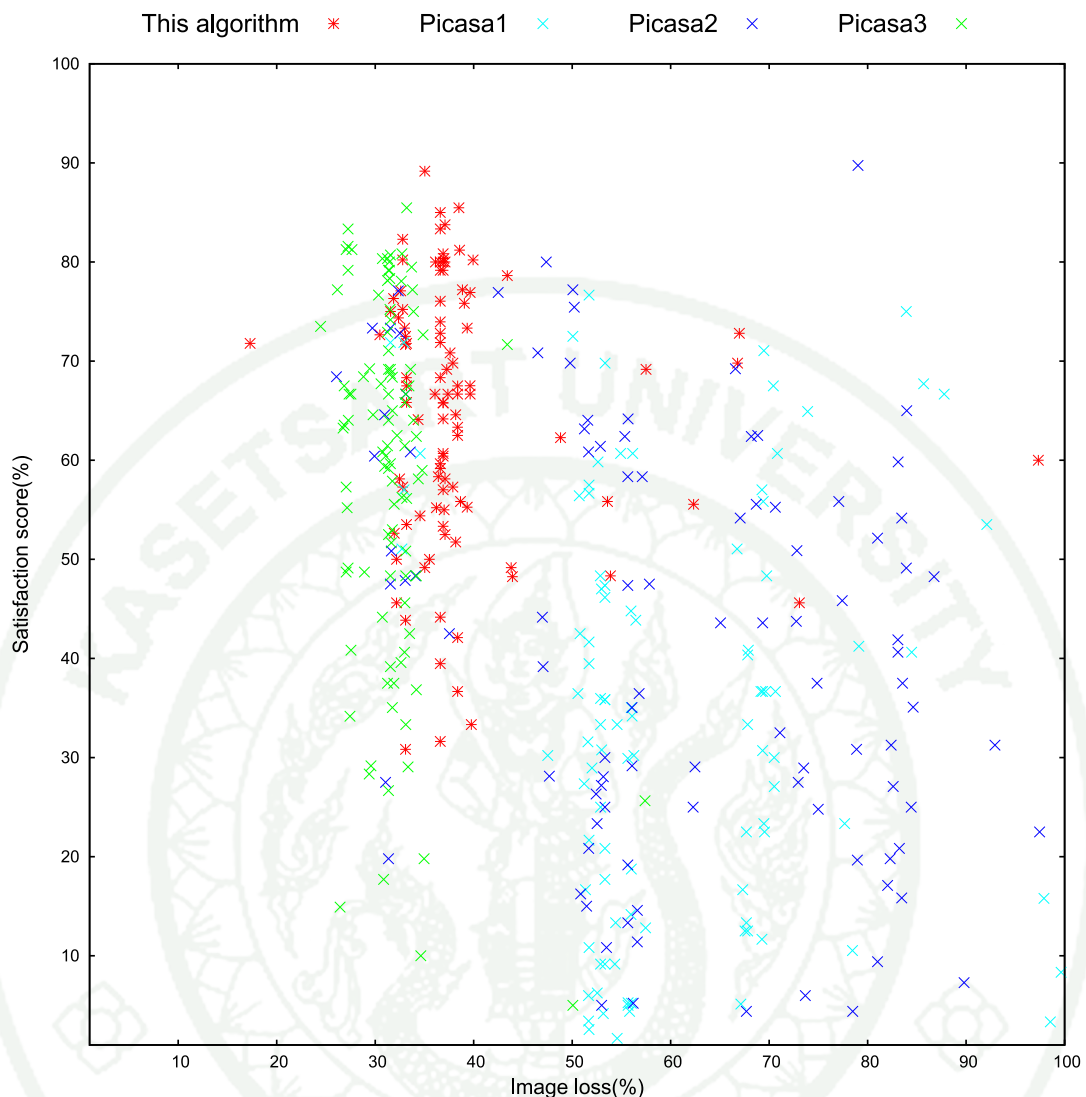


Figure 61 Relationship between user satisfaction scores (0=Reject; 1=Poor; 2=Good; 3=Excellent) and percent data loss of four cropping algorithms.

We also investigated whether the demography of the evaluators had an effect on the results. A one-way within subjects ANOVA was conducted to compare the effect of age and gender on this algorithm, Picasa 1, Picasa 2 and Picasa 3, the analysis result shown that there was not a significant effect of age and gender on every result ($p > 0.05$). However, evaluators with and without background in photographic composition had different opinions on the results. The t-test results indicate that evaluators without photographic backgrounds cannot distinguish differences between results from Picasa3 and this algorithm, the results show our is

3.3% better than Picasa 3 but t-test concluded that has no significant difference ($p>0.5$). However, evaluators who have background in photography strongly ranked this algorithm 7.3% higher than Picasa3 ($p<0.5$). Details of evaluation scores are shown in Table 11. We also conclude that, photographers prefer our proposed algorithm and general users cannot decide the best between this algorithm and Picasa 3; however, this algorithm allows higher loss of information.

Table 11 User satisfaction score classified by background in photography.

Parameters	Satisfaction (%)			
	This algorithm	Picasa3	Picasa2	Picasa1
Evaluators have background in photography	63.6	56.3	40.3	33.0
Evaluators do not have backgrounds in photography	67.3	63.0	44.0	33.0
All	64.6	58.3	41.3	34.3

2. Video cropping algorithm evaluation results

The video cropping evaluation results are divided into two parts; compare with traditional method and ground truth. The results are describe as follow.

2.1.Compare Video cropping algorithm with traditional method

Qualitative evaluation results, the evaluation of experimental results on the quality of the cropping algorithms is shown in Figure 63 . We quantify the qualitative results of “excellent”, “acceptable”, or “reject” as 2, 1, or 0, respectively. From the results, it is clear that our algorithm outperforms others especially in the situation that amount of data loss is high. In other words, the more different aspect ratio, the better our algorithm is. The statistical details of the survey are shown in Table 12. The collected data of every method and aspect ratio are consistent (standard error < 0.05). We use paired sample T-test to compare means among results of each aspect ratio and we found that our proposed method performs better

than other methods with significant ($p \leq 0.05$). The performance also depends on the aspect ratio different between source video and target display. The more aspect ratio difference is, the higher image loss is in video cropping process, the results also show that performance difference of each algorithm is not significant if image loss is low. The trend clearly shows that our method performs better when aspect ratio difference between of movies and of screens is high

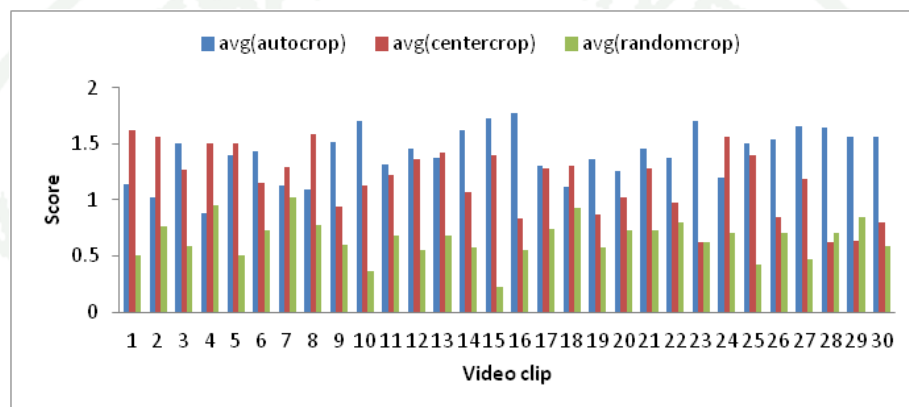
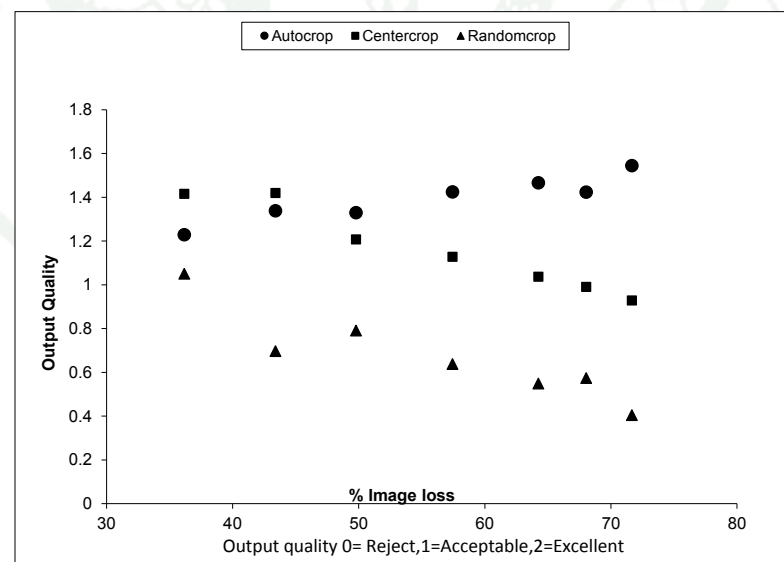
Taking demographic information into account did not reveal any statistically significant differences. We used one-way ANOVA to compare groups with different age, we get the calculated F value = 0.832 and P value=0.506; therefore, we interpret that there was no different satisfaction feedback from different group of ages, the detailed report from PSPP are shown in Table 12 and Table 13.

Table 12 Descriptive statistic of different ages

Age	N	Mean	Std. Deviation	Std. Error	95% Confidence Interval for Mean		Minimum	Maximum
					Lower Bound	Upper Bound		
16-20	30	1.3352	0.36756	0.06711	1.1979	1.4724	0.00	2.00
21-25	58	1.3255	0.32357	0.04249	1.2404	1.4106	0.00	1.83
26-30	15	1.4744	0.45096	0.11644	1.2246	1.7241	0.00	2.00
31-35	9	1.4196	0.27166	0.09055	1.2107	1.6284	1.00	1.68
36-40	2	1.5750	0.10607	0.07500	0.6220	2.5280	1.50	1.65
Total	114	1.3594	0.34874	0.03266	1.2947	1.4241	0.00	2.00

Table 13 ANOVA result of age factor

	Sum of Squares	df	Mean Square	F	Sig.
Between Groups	0.408	4	0.102	0.834	0.506
Within Groups	13.335	109	0.122		
Total	13.743	113			

**Figure 62** Results group by video clip**Figure 63** The mean result of each image loss produced by aspect ratio different

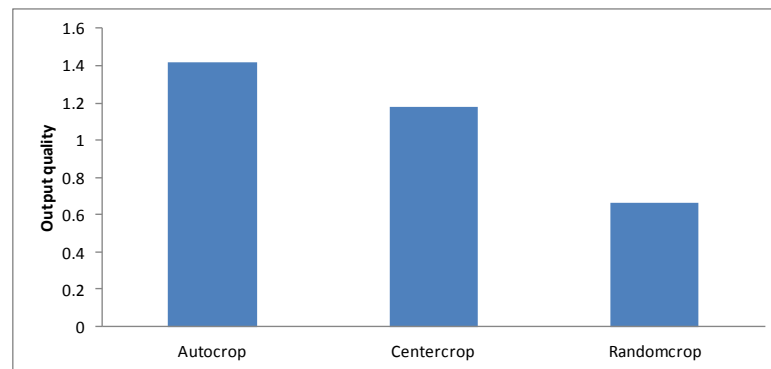


Figure 64 The mean result of all method over all image loss

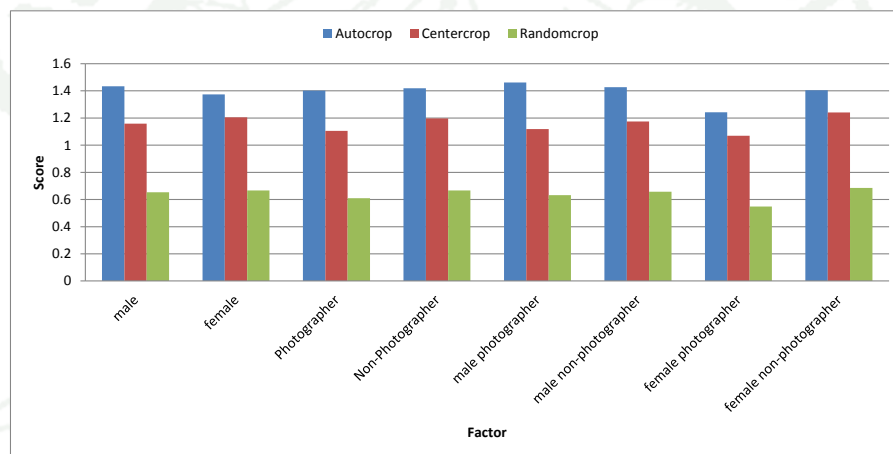


Figure 65 Results group by factor

The t-test is used to compare questionnaire results. The results shown P values between populations that have and have not background in photography are 0.07 for auto crop, 0.93 for center crop and 0.47 for placebo. Therefore, we conclude that differences in background knowledge of evaluators do not affect the result. The average result of each video clip shows in Figure 62, the average result group by factor shows in Figure 65. The average score of all reference method over all image loss is shown in Figure 64.

Table 14 Statistical details of satisfaction scores of each video cropping method

<i>Aspect ratio</i>	<i>Image Loss(%)</i>	<i>Method</i>	<i>DESCRIPTIVES</i>			<i>Paired Sample Test</i>	<i>T-TEST</i>			<i>Winner</i>
			Mean	Standard Error	Std Dev		Standard Error	t	p	
1.5:1	36.17	Autocrop	1.23	0.03	0.77	Autocrop-Centercrop	0.05	-3.73	0	Centercrop
		Centercrop	1.41	0.03	0.69	Autocrop-Randomcrop	0.05	3.31	0	
		Randomcrop	1.05	0.04	0.81	Centercrop-Randomcrop	0.05	7.19	0	
1.33:1	43.4	Autocrop	1.34	0.03	0.73	Autocrop-Centercrop	0.05	-1.65	0.1	Unclear
		Centercrop	1.42	0.03	0.64	Autocrop-Randomcrop	0.05	11.86	0	
		Randomcrop	0.7	0.03	0.76	Centercrop-Randomcrop	0.05	15.61	0	
1.18:1	49.79	Autocrop	1.33	0.03	0.75	Autocrop-Centercrop	0.05	2.39	0.02	Autocrop
		Centercrop	1.21	0.03	0.73	Autocrop-Randomcrop	0.06	9.52	0	
		Randomcrop	0.79	0.04	0.79	Centercrop-Randomcrop	0.05	7.85	0	
1:1	57.45	Autocrop	1.41	0.03	0.69	Autocrop-Centercrop	0.05	5.72	0	Autocrop
		Centercrop	1.13	0.03	0.72	Autocrop-Randomcrop	0.05	14.89	0	
		Randomcrop	0.64	0.03	0.72	Centercrop-Randomcrop	0.05	9.32	0	
1:1.18	64.26	Autocrop	1.47	0.03	0.71	Autocrop-Centercrop	0.05	8.09	0	Autocrop
		Centercrop	1.04	0.03	0.74	Autocrop-Randomcrop	0.05	18.17	0	
		Randomcrop	0.55	0.03	0.68	Centercrop-Randomcrop	0.05	9.73	0	
1:1.33	68.04	Autocrop	1.42	0.03	0.73	Autocrop-Centercrop	0.05	8.16	0	Autocrop
		Centercrop	0.99	0.03	0.53	Autocrop-Randomcrop	0.05	16.06	0	
		Randomcrop	0.57	0.03	0.48	Centercrop-Randomcrop	0.05	8.66	0	
1:1.5	71.66	Autocrop	1.54	0.03	0.63	Autocrop-Centercrop	0.05	12.59	0	Autocrop
		Centercrop	0.93	0.03	0.74	Autocrop-Randomcrop	0.04	26.08	0	
		Randomcrop	0.4	0.03	0.62	Centercrop-Randomcrop	0.05	10.66	0	

2.2.Comparison with “cropped by expert” version experiment result

Precision results for our algorithm are shown in Table 15; the average precision is 0.82. The example result with different precision is shown in .Figure 67. When we look for the locations of errors, we found that most video clips with highly different errors are also hard for humans to crop as well. For example, the original frame as shown in Figure 66(A); the result of our algorithm is shown in Figure 66 (B); and commercial pan & scan version of the same frame is shown in Figure 66 (C). Both results are likely to be equally correct. If we removed all clips which have this situation (undecidable case), the precision of our method is 0.91. Figure 67E shows an example of precision= 0.91, it demonstrates that it is hard to notice the difference compared with precision=1 in Figure 67B. We conclude that, ignoring undecidable cases, our algorithm can produce equivalent results to experts' cropping.



Figure 66 Example of scenario that there is more than one correct position to crop:
(A) original DVD frame; (B) our result and (c) VCD frame

Table 15 Cropping precisions of our algorithm on selected motion pictures

<i>Motion picture</i>	<i>Precision</i>	<i>Precision (ignoring undecidable cases)</i>
Transformers: Revenge of the Fallen (2009)	0.84	0.92
Raiders of the Lost Ark (1981)	0.81	0.91
Indiana Jones and the Temple of Doom (1984)	0.80	0.90
Indiana Jones and the Last Crusade (1989)	0.83	0.91
Mercury Rising (1998)	0.82	0.91
Star Trek (2009)	0.84	0.91
Average	0.82	0.91

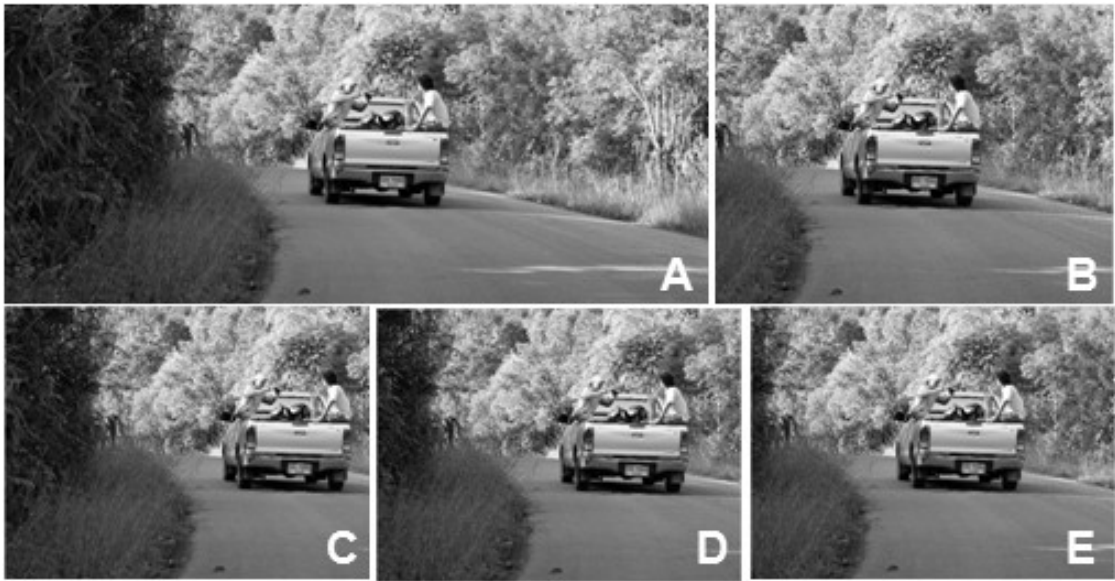


Figure 67 Comparison results in different precisions

Even though the accuracy evaluation suggests that our algorithm could replace the manual method, we further investigate the quality of cropping compared with the manual method. We select some scenes where precision is less than 40%, since it is different enough for audiences to distinguish and it is possibly an “undecidable” situation. We ask audiences to compare 40 cropped versions by expert with results from our algorithms in the controlled-environment. The example of the web page is shown in Figure 68. From a total of 160 respondents, 63% prefer results from our algorithms and the remaining 37% prefer the VCD version, where significant difference is confirmed by t-test ($p < 0.05$). Detailed results are shown in Figure 69.



Figure 68 Example of webpage for evaluation between result from expert and result from our algorithm

Table 16 Cropping accept rate for undesirable cases

<i>Motion picture</i>	<i>Descriptive</i>		<i>Paired T-test between our algorithm and VCD version</i>			
	Accept rate		Std Dev	Std Error	t	p
	Our Algorithm	VCD Version				
Transformers: Revenge of the Fallen (2009)	70.40%	29.59%	0.39	0.02	14.10	0
Raiders of the Lost Ark (1981)	60.76%	39.23%	0.48	0.01	10.09	0
Indiana Jones and the Temple of Doom (1984)	54.01%	45.58%	0.50	0.02	4.42	0.02
Indiana Jones and the Last Crusade (1989)	67.65%	32.34%	0.48	0.01	13.32	0
Mercury Rising (1998)	57.33%	42.66%	0.50	0.01	4.95	0
Star Trek (2009)	65.95%	34.04%	0.48	0.01	8.11	0
Average	62.68%	37.24%				



Figure 69 Example of undesirable cases

3. Video indexing based on N-gram evaluation results

After the best practical values of parameters k and l are determined, we used it to set our algorithm to compare performances (precision and recall) with a baseline algorithm LSH-E (Dong *et al.*, 2008) and a benchmark VK (Wan-Lei *et al.*, 2010). Computation time for signature generation is also measured. Locally sensitive hashing embedding on color moment (LSH-E), which is histogram based signature detection, is our baseline. VK (visual keyword), which is a high-level object recognition, is our benchmark. VK usually requires very high computation power because it is object-recognition algorithm.

Figure 70 depicts benchmark and baseline comparison results. Note that the averages of results are in the rightmost sets. It shows that precision, recall, and F from our method are higher than LSH-E; however, it cannot compete with the VK method in average. The average result of all 24 queries is 0.85 for accuracy, 0.72 for precision and 0.58 for recall, the average of precision and recall shown that our method is 10% better than baseline method.

For specific details of results in each query are described as follows. Our method produces comparably equal performance to other algorithms for queries 1, 13, 18 and 24. Queries 1, 13, and 24 are described in the dataset as *"simple scene or complex scene with minor editing and variations"*. The query 18 is described as *"bad video quality shot with a cell phone camera"*. Our results are better than the baseline but worse than the benchmark for queries 10, 15, 22 and 23, which are described as *"extensive editing, modification with a lot of unrelated frames attached to the beginning and the end"*, comparison of average results of all approaches are demonstrated in Table 17.

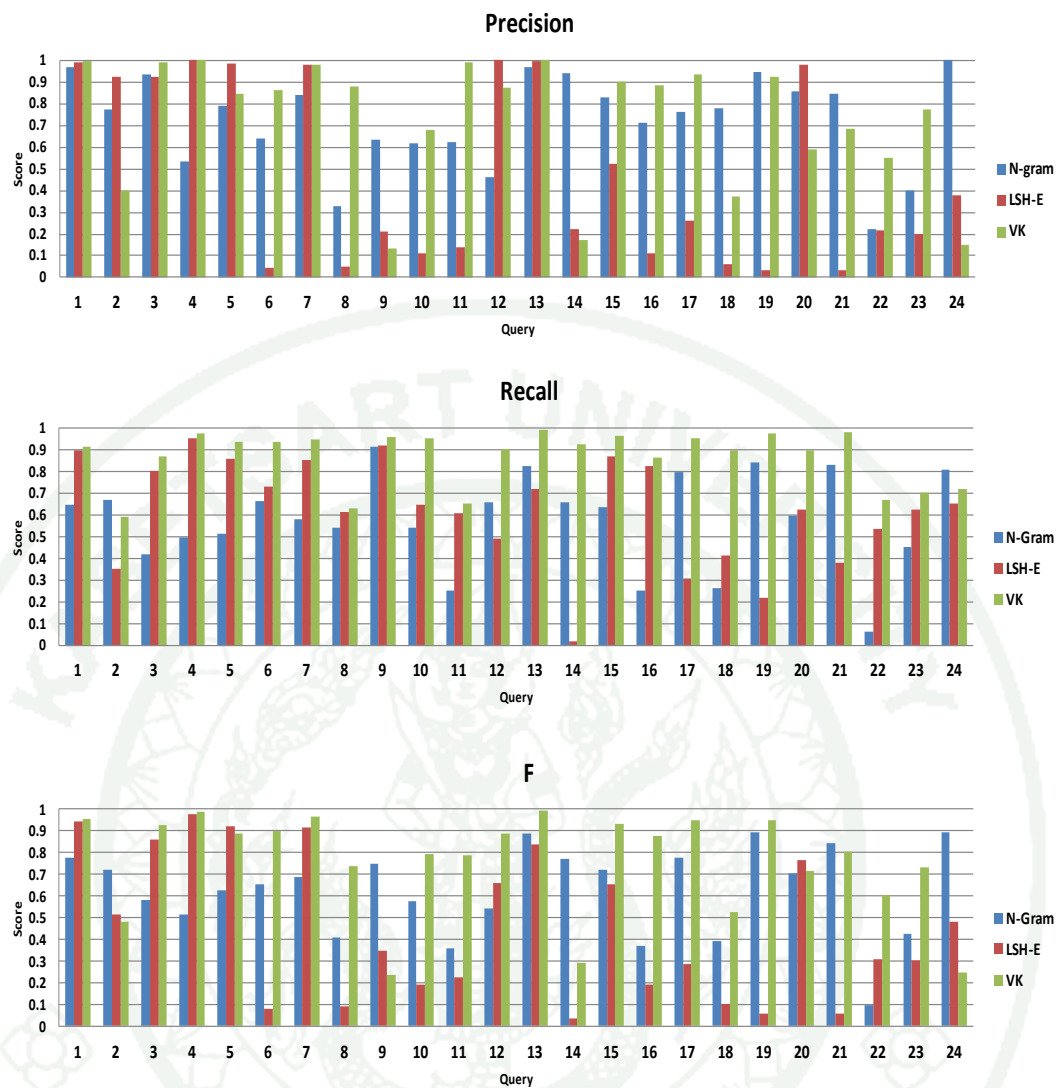


Figure 70 Precision, recall and F-measure compared with LSH-E and VK.

Table 17 Video indexing based on N-gram average results

Algorithm	Precision	Recall	F-measure
N gram	0.726	0.580	0.624
LSH-E	0.474	0.622	0.450
VK	0.732	0.868	0.757

4. Video indexing based on spectrogram evaluation results

In this experiment, performance results are visualized on recall/precision plots since satisfaction of each user varies to types of applications. Therefore, a user may evaluate the performance by either recall or precision.

The results of this algorithm are compared with three algorithms: sliding window in (Xian-Sheng *et al.*, 2004), spatial-temporal distribution signature based (STD) in (BaoFeng *et al.*, 2010) and dynamic time warping on signature based on key-frame without spectrogram (DTW) in (Chih-Yi *et al.*, 2006). Performance on those three algorithms are shown in Recall/Precision graphs in Figure 71, Figure 72, and Figure 73, respectively. Note that the nearer to top-right corner of recall/precision graphs (recall=1 and precision=1), the better results are. Precision of sliding window and STD algorithm is inconsistent. Precision of many queries begin to degrade since recall=0.1. DTW algorithm produces higher precision than STD in most queries except the query numbers 6, 18, and 22. We further investigate this algorithm. Instead of using only key frame, we all frames as signature. The result is shown as DTW+. In Figure 74, the overall precision is improved and only two queries get low-precision results. However, this algorithm is very exhaustive since its signature is long. The length of DTW+ signature also longer than that of DTW. Finally, result of the proposed algorithm is shown in Figure 75. Note that its overall precision is higher than of all reference algorithms. Results are very consistent for almost all queries except for queries 18 and 22. However, queries 18 and 12 are low-resolution videos and extensively edited videos, respectively. Precisions of all topics when recall =1 generated by all five algorithms are shown in Figure 76. It is clearly shown that precisions of this algorithm are better than those of reference algorithms in all queries. Average precision of all queries of this algorithm are 19.6%, 16.6%, 11.7%, and 4.4% as high as sliding window, STD, DTW, and DTW+ algorithms, respectively.

The overall precision over all recall levels are concluded in Table 18. Average precision of this algorithm is higher than all reference algorithms. The DTW+ produced only 0.014 lower than this algorithm, but it is confirmed by paired t-test

($t=14.75, p=0$) that it is significantly different. Standard deviations also show this algorithm produced the most consistent results.

The average total signature length produced by all five algorithms are also shown in Table 18. Length of ordinal feature is 9 ($k=9$) times of video length. Sliding window needs to compare the whole length since it does not compact the signature. DTW+ compacts data from 9 dimensions to 1 dimension so the total length is reduced 9 times of the original. DTW compacts up to 18.1 times. This algorithm compacts further up to 61.4 times. The STD generates smallest signature because it is not based on ordinal feature; however it also generated the lowest precision.

Table 18 Average and standard deviation of precision.

Methods	Sliding window	STD	DTW	DTW+	This algorithm
Average of Precision	0.890	0.881	0.937	0.972	0.986
SD. of precision	0.169	0.175	0.133	0.091	0.061
Average of total signature length	33991	1	1871	3777	553

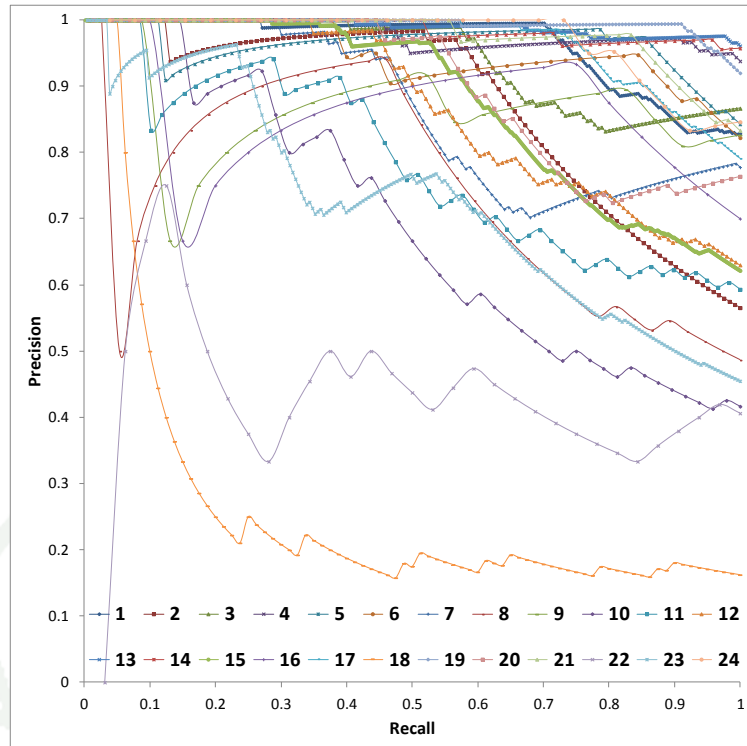


Figure 71 Ranking performance of sliding window algorithm

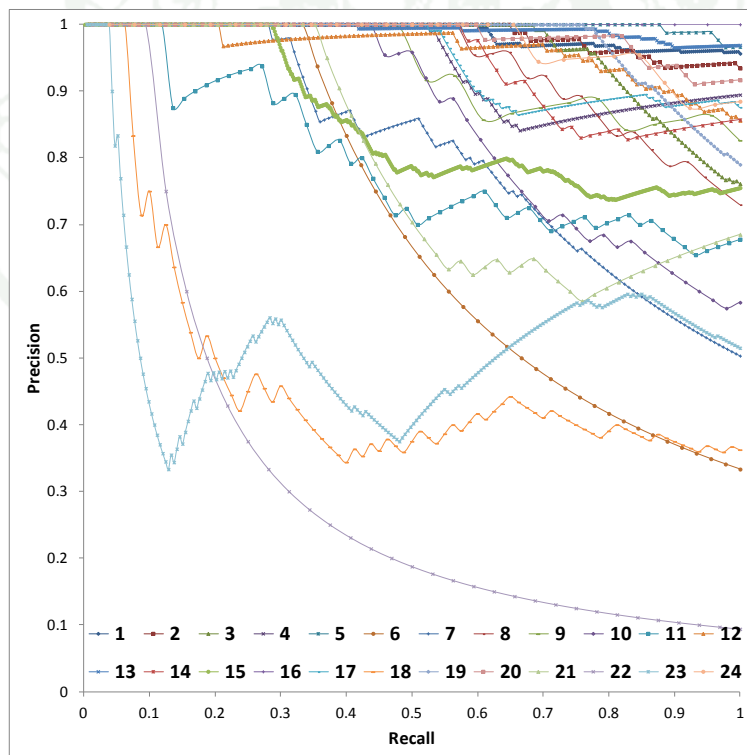


Figure 72 Ranking performance of STD algorithm

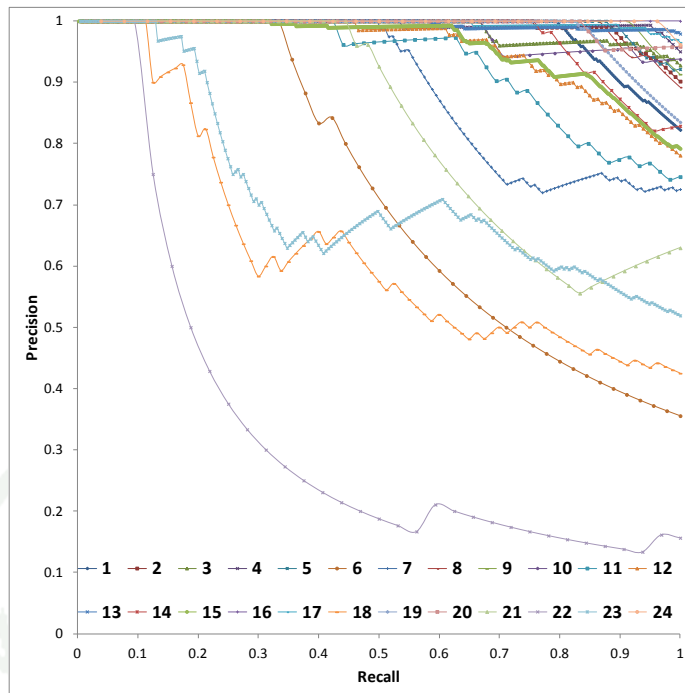


Figure 73 Ranking performance of DTW algorithm

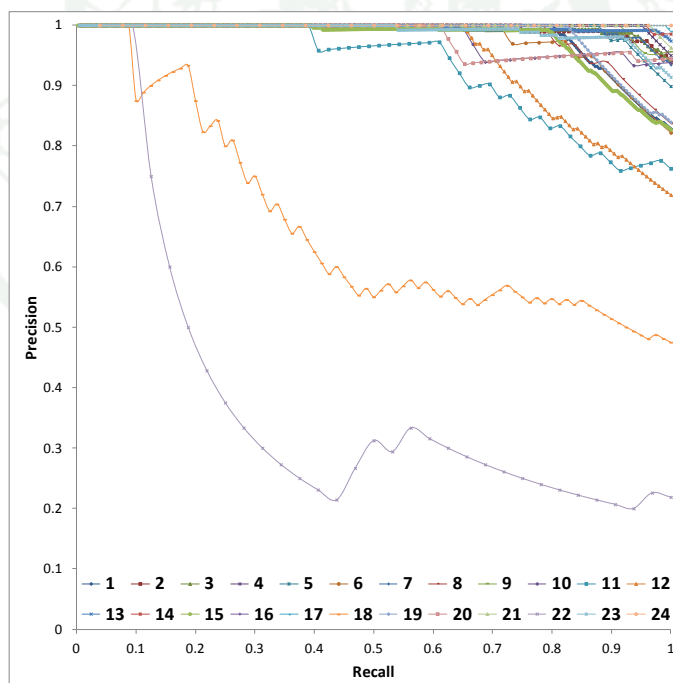


Figure 74 Ranking performance of DTW+ algorithm

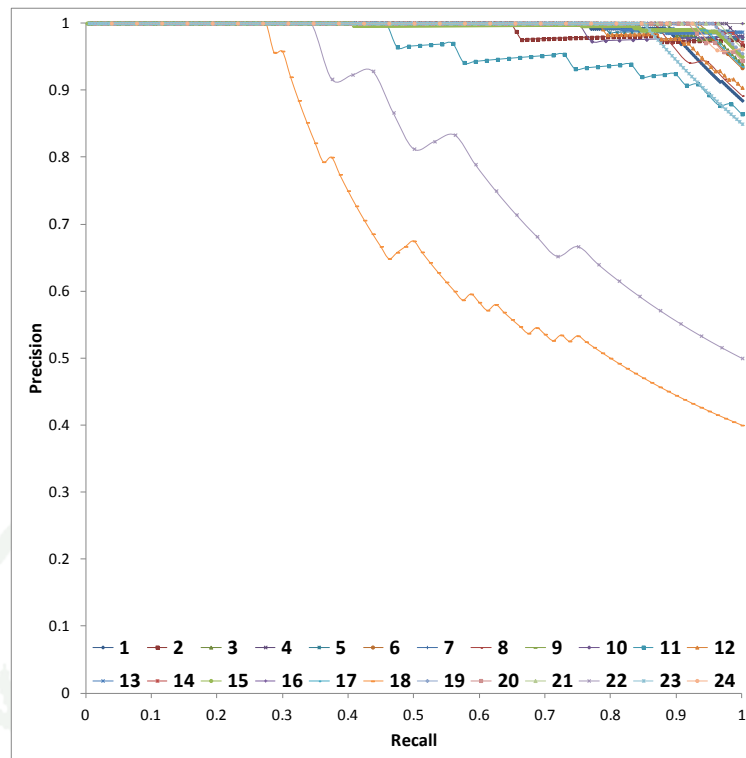


Figure 75 Ranking performance of proposed algorithm

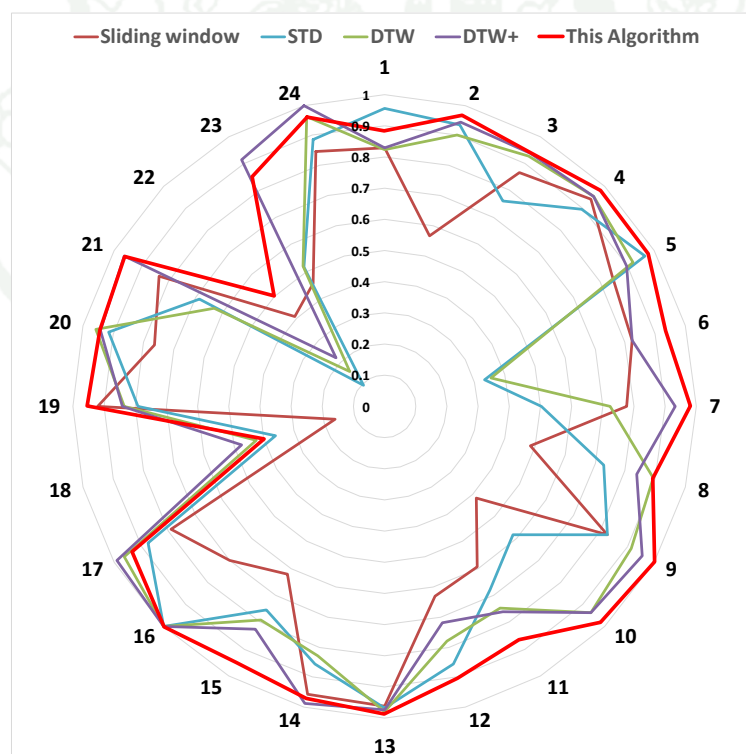


Figure 76 Radar plot of all algorithms when recall=1

We noticed that some types of video modification have major effects on performance. Those types are 1) minor-edited videos (queries #1, #13, #16, and #24), 2) extensively-edited videos (queries #10, #15, #22, and #23), and 3) low quality videos (query #18). Precisions at recall=1 of these groups are shown in Figure 77. Precision compare between video groups at recall =1. Precision of minor-edited videos from all five algorithms are almost similar except Sliding Window. The proposed algorithm produces the highest precision among all reference algorithms. This algorithm also produces the highest precision in extensively-edited videos. This algorithm cannot outperform DTW and DTW+ algorithms in the low quality video; however, our method produces much shorter signatures.

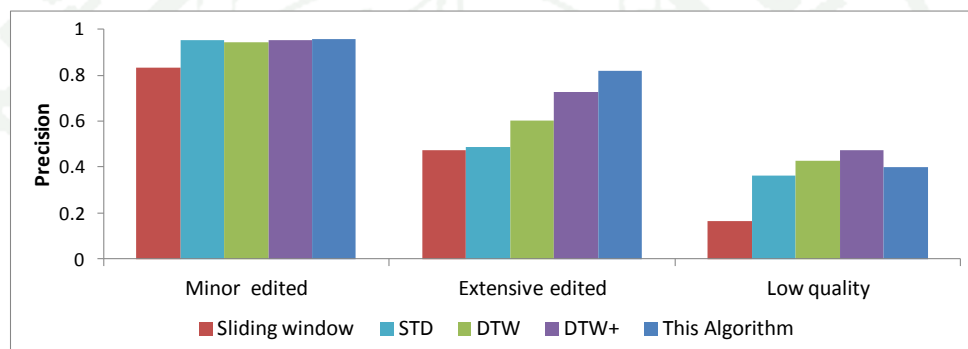


Figure 77 Precision compare between video groups at recall =1

Discussion

1. Image cropping algorithm discussion

By observation, we found that this algorithm did not perform well in two situations: low-contrast image and noisy image. When the image has low contrast, such as a foggy scene or motion blur, it affects the sharpness of the image. Both focus and face maps are unclear and the suggested cropping area tended to be too large; therefore, it cannot allow high amounts of loss. For noisy image, general face detection algorithms try to find all possible faces and it is sometimes too sensitive to detect false-positive objects. We are investigating how to compensate weights based on sensitivity of importance map generator.

2. Video cropping algorithm discussion

The robustness of auto-cropping algorithms depends on cropped area constraint; the greater the loss of picture, the more complicated the cropper's task is. In our experiment, we tested algorithm performance with different percentage loss of image area. For less than 10% loss, such as crop from American widescreen (1.85:1) to HDTV format (16:9), there is no need for an automatic algorithm since the results from center crop, and our algorithm are not significantly different ($p=1$). However, at 30-50% loss such as anamorphic widescreen (2.35:1) to HDTV, the performance of our algorithm starts to show significant improvement and results show that placebo and fixed window do not perform well, so a more sophisticated algorithm is required. Finally, for more than 50% loss, such as anamorphic widescreen to standard square screen (1:1), results show that the performance of our algorithm was significantly preferred to other methods.

Theoretically, when the aspect ratio difference is less than 50% you can randomly crop and the result will always intersect with the ground truth image and the precision calculated by equation 14 is always greater than 0. The less the aspect ratio difference, the higher the crop precision is. However, when the aspect ratio difference is larger than 50%, random crop or center crop is not acceptable because

there is a chance that the crop result will not intersect with the ground truth image. This scenario requires a sophisticated algorithm and our proposed algorithm was demonstrated to be the best for this scenario.

3. Video indexing based on N-gram discussion

Our method is almost four times as faster as VK method for signature generation process and 16 times as faster as query process. For LSH-E, the signature generation time is pretty much the same with our method. However, LSH-E requires ranking with all items in database while our method uses binary search. From the experiments we concluded that our proposed algorithm is fast and good for query with simple scene or complex scene with minor editing or low quality videos. However, our algorithm is still intolerant to heavily-modified video, which requires high-level object detection to analyze.

4. Video indexing based on spectrogram discussion

The proposed algorithm produces the highest precision among all reference algorithms. This algorithm also produces the highest precision in extensively-edited videos. This algorithm cannot outperform DTW and DTW+ algorithms in the low quality video; however, our method produces much shorter signatures. From the experiments, every algorithm achieves high precision at recall < 0.2 ; therefore, every algorithm is suitable for applications that do not require high recall rate. However, for applications that require high precision and high recall, our algorithm outperforms other algorithms for edited videos. Moreover, our signature is also more compact than almost every reference algorithm.

CONCLUSION AND RECOMMENDATION

Conclusion

Image cropping algorithm, video cropping algorithm and video indexing algorithm that based on cinematic features were proposed in this thesis. Cinematic associated with each algorithm is shown in Table 19.

An automatic image cropping algorithm based on inference of the photographer's intention. The focal-point and face location are used as clues about the location in the picture that the photographer is most interested in. These features are used to generate an importance map. An automatic algorithm is designed based on the pan-and-scan method where a genetic algorithm selects the location that has maximal information per unit area from the importance map. To zoom in on the target area, a wrapper for the pan-and-scan-based algorithm is applied with successively smaller cropping size until a data-loss threshold is reached. Experiments were performed to compare user satisfaction with the cropped images proposed by this algorithm as compared with cropped images from three cropping algorithms used in Google's commercial products, using images from the National Geographic website. The result shows that this algorithm is 30.3%, 23.0% and 6.0% better than Picasa1, Picasa2, and Picasa3 algorithms, respectively. Users without photography backgrounds prefer this algorithm to Picasa1 and Picasa2 and equally satisfied with cropping results from algorithm and Picasa 3, even though this algorithm allows 7% higher data loss. On the other hand, users with background in photograph prefer images cropped by this algorithm to images from any of Picasa's at the same level of data loss.

Automatic real-time video cropping for heterogeneous display devices using cinematic features was proposed. Cinematic features are evident when the cameraman uses some techniques such as zooming, panning, and focusing, during the shooting. We proposed algorithms to analyze the footprints of those techniques in order to determine an importance map, which collects possible area of interest of the

cameraman. The importance map is preprocessed at the server, and then transmitted synchronously with the associated video stream. The client, which may have various sizes of display, will use the importance map to determine the best cropping policy on its display device. In experiments, we found that the uncompressed overhead size is 2.1% of the original video. The volunteers' evaluations indicate that our algorithm outperforms others, especially when the aspect ratio of the original movie and client's display device differ greatly. When we compared the results of our algorithm with a commercial pan-and-scan version of the same movie with the same target aspect ratio, when cropping positions of both versions are more than 40% different, our results are preferable to test audiences. For less than 40% difference, audiences felt the quality of both versions was similar. Therefore, we believe that manual cropping for commercial video could be replaced with our automatic algorithm. Our study also showed that a cropping algorithm becomes more important when the aspect ratio difference is more than 43%, such as cropping a widescreen movie to fit onto portrait phone display. However, our current implementation has some limitations. In particular, our video cropping algorithm assumes that the video can be cropped using a single window; if the important object is both located in the leftmost and the rightmost part of the screen at the same time then the algorithm will not perform accurately.

The video copy detection method based on signature approach. The luminance velocity was proposed to represent the video signature. The query method was adapted from text document similarity detection algorithm using N-Gram algorithm. This method extracts shot transitions from video clips and used them as signature. The evaluation was done with an internet video dataset that contains 12,790 near-duplicate videos which are results of 24 queries onto popular internet video sites. The result indicates that our method is fast and effective to detect near-duplicate videos, especially for simple scene or complex scene with minor editing and low quality or low bit-rate videos.

The spectrogram signature was proposed to extend the ranking capability. Signature is generated by the following steps. First, ordinal feature is extracted from each frame. Second, ordinal features of all frames are transformed by short-time

Fourier transform. The transformed results are used to generate a spectrogram. Signature similarity measurement is designed by using pairwise similarity matrix and sequence alignment. The alignment process is done by using dynamic time warping algorithm. The result was clearly shown that our method is effective for ranking the similarity on large diversity of near-duplicate videos. Our work can be applied to video query by example system.

Table 19 Cinematic used in each part of this thesis

Applications	Cinematic				
	Focusing	Acting	Tracking	Panning	Transitioning
Image cropping	X				
Video cropping	X	X	X	X	
Video indexing					X

Recommendation

Further study research would have to collaborate with other features such as object recognition and audio. Video that has rapid movement need more sophisticated motion analysis such as wavelet analysis. Face recognition of main actors and actresses should also be taken in consideration. Future work could include content-aware resizing in any particular shot and the video auto cropping algorithm for 3D video should be investigated.

In some situation video cannot be cropped or it will lose the important information, in this case the method to detect this type of video shot must be investigated. And finally our proposed method is to show the potential to be improved to use data directly from MPEG stream instead of using bitmap image; in this case the computation required can be significantly reduced.

LITERATURE CITED

- A, A.H., K. ho Hyun B and R.B. A. 2000. Comparison of Sequence Matching Techniques for Video Copy Detection, p. 194-201. *In Proc. SPIE.*
- Abrams, J.J. (2009) **Star Trek**. pp. 127 min.
- Allen, J.B. and L. Rabiner. 1977. A Unified Approach to Short-Time Fourier Analysis and Synthesis. **Proceedings of the IEEE** 65 (11): 1558-1564.
- Apple.com. 2012. **iPhone 3gs Technical Specification**. Available Source: <http://www.apple.com/iphone/iphone-3gs/specs.html>,
- Ardizzone, E., M. La Cascia and D. Molinelli. 1996. Motion and Color-Based Video Indexing and Retrieval, p. 135-139 vol.133. *In Pattern Recognition, 1996., Proceedings of the 13th International Conference on.*
- Baina, J. and J. Dublet. 1995. Automatic Focus and Iris Control for Video Cameras, p. 232-235. *In Image Processing and its Applications, 1995., Fifth International Conference on.*
- BaoFeng, L., C. HaiBin and C. Zheng. 2010. An Efficient Method for Video Similarity Search with Video Signature, p. 713-716. *In Computational and Information Sciences (ICCIS), 2010 International Conference on.*
- Bay, M. (2009) **Transformers: Revenge of the Fallen**. pp. 150 min.
- Becker, H. (1998) **Mercury Rising**. pp. 111 min.
- Bolosky, W.J., S. Corbin, D. Goebel and J.R. Douceur. 2000. Single Instance Storage in Windows 2000, p. 2. *In Proceedings of the 4th conference on USENIX Windows Systems Symposium - Volume 4.* USENIX Association, Seattle, Washington.
- Bradski, G. 2000. OpenCv_Library. **Dr. Dobb's Journal of Software Tools**
- Byers, Z., M. Dixon, K. Goodier, C.M. Grimm and W.D. Smart. 2003. An Autonomous Robot Photographer, p. 2636-2641 vol.2633. *In Intelligent Robots and Systems, 2003. (IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on.*
- Cao, Z. and M. Zhu. 2009. An Efficient Video Similarity Search Strategy for Video-on-Demand Systems, p. 174-178. *In Broadband Network & Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on.*

- Chee Hwa, C., Z. Ying and W. Changyun. 2003. Visual Servo Auto-Focusing Using Recursive Weighted Least-Squares for Machine Vision Inspection, p. 777-780. *In Electronics Packaging Technology, 2003 5th Conference (EPTC 2003).*
- Chen, D.-Y. and Y.-S. Luo. 2011. Content-Aware Video Resizing Based on Salient Visual Cubes. **J. Vis. Comun. Image Represent.** 22 (3): 226-236.
- Chen, L.-Q., X. Xie, X. Fan, W.-Y. Ma, H.-J. Zhang and H.-Q. Zhou. 2003. A Visual Attention Model for Adapting Images on Small Displays. **Multimedia Systems** 9 (4): 353-364.
- Cheng, W.H., C.W. Wang and J.L. Wu. 2007. Video Adaptation for Small Display Based on Content Recomposition. **Circuits and Systems for Video Technology, IEEE Transactions on** 17 (1): 43-58.
- Cheol-Hee, P., P. Jong-Ho, Y. Young-Hwan, S. Hyoung-Kyu and C. Yong-Soo. 2000. Auto Focus Filter Design and Implementation Using Correlation between Filter and Auto Focus Criterion, p. 250-251. *In Consumer Electronics, 2000. ICCE. 2000 Digest of Technical Papers. International Conference on.*
- Chih-Yi, C., L. Cheng-Hung, W. Hsiang-An, C. Chu-Song and C. Lee-Feng. 2006. A Time Warping Based Approach for Video Copy Detection, p. 228-231. *In Pattern Recognition, 2006. ICPR 2006. 18th International Conference on.*
- Chun-Hung, S. and H.H. Chen. 2006. Robust Focus Measure for Low-Contrast Images, p. 69-70. *In Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on.*
- Chung Nam, C., J. Jung Hun, Y. Joon Shik, S. Jeong Ho and P. Joon Ki. 1999. Region Selectable Auto-Focusing System by Digital Psf Estimation, p. 92-93. *In Consumer Electronics, 1999. ICCE. International Conference on.*
- Colmenarez, A.J. and T.S. Huang. 1997. Face Detection with Information-Based Maximum Discrimination, p. 782-787. *In Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on.*
- Dalton, C.J. 1996. Wide-Screen Production Engineering-an Overview from Source to Display, p. 366-372. *In Broadcasting Convention, International (Conf. Publ. No. 428).*
- Diansheng, C., G. Yunguo and L. Huanli. 2010. Auto-Focusing Evaluation Functions in Digital Image System, p. V5-331-V335-334. *In Advanced Computer Theory and Engineering (ICACTE), 2010 3rd International Conference on.*

- Dong, W., Z. Wang, M. Charikar and K. Li. 2008. Efficiently Matching Sets of Features with Random Histograms, p. 179-188. *In Proceedings of the 16th ACM international conference on Multimedia*. ACM, Vancouver, British Columbia, Canada.
- Drude, S. and M. Klecha. 2006. System Aspects for Broadcast Tv Reception in Mobile Phones, p. 269-270. *In Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on*.
- Evans-Pughe, C. 2005. Movies on the Move [Mobile Phone Video Content Reception]. *IEE Review* 51 (10): 44-47.
- Fan, X., X. Xie, H.-Q. Zhou and W.-Y. Ma. 2003. Looking into Video Frames on Small Displays, p. *in Proceedings of the eleventh ACM international conference on Multimedia*. ACM, Berkeley, CA, USA.
- Faria, G., J.A. Henriksson, E. Stare and P. Talmola. 2006. Dvb-H: Digital Broadcast Services to Handheld Devices. *Proceedings of the IEEE* 94 (1): 194-209.
- Feng, L. and J. Hong. 2005. A Fast Auto Focusing Method for Digital Still Camera, p. 5001-5005 Vol. 5008. *In Machine Learning and Cybernetics, 2005. Proceedings of 2005 International Conference on*.
- Flickner, M., H. Sawhney, W. Niblack, J. Ashley, H. Qian, B. Dom, M. Gorkani, J. Hafner, D. Lee, D. Petkovic, D. Steele and P. Yanker. 1995. Query by Image and Video Content: The Qbic System. *Computer* 28 (9): 23-32.
- Frintrop, S., E. Rome and H.I. Christensen. 2010. Computational Visual Attention Systems and Their Cognitive Foundations: A Survey. *ACM Trans. Appl. Percept.* 7 (1): 1-39.
- Geographic, N. 2013. **National Geographic Photo of the Day**. Available Source: <http://photography.nationalgeographic.com/photo-of-the-day/>,
- Google. 2013a. **Youtube.Com**. Available Source: <http://www.youtube.com>,
- Google. (2013b) **Picasa**, Google.
- Grinias, I. and G. Tziritas. 2002. Robust Pan, Tilt and Zoom Estimation, p. 679-682 vol.672. *in Digital Signal Processing, 2002. DSP 2002. 2002 14th International Conference on*.
- Gsmarena.com. 2012a. Nokia N80 Technical Specification.

- Hamzah, R.A., R.A. Rahim and Z.M. Noh. 2010. Sum of Absolute Differences Algorithm in Stereo Correspondence Problem for Stereo Matching in Computer Vision Application, p. 652-657. **In Computer Science and Information Technology (ICCSIT), 2010 3rd IEEE International Conference on.**
- He, L.-w., M.F. Cohen and D.H. Salesin. 1996. The Virtual Cinematographer: A Paradigm for Automatic Real-Time Camera Control and Directing, p. 217-224. **In Proceedings of the 23rd annual conference on Computer graphics and interactive techniques.** ACM.
- Herrero, C. and P. Vuorimaa. 2004. Delivery of Digital Television to Handheld Devices, p. 240-244. **In Wireless Communication Systems, 2004. 1st International Symposium on.**
- Hyeong-Min, N., B. Keun-Yung, J. Jae-Yun, C. Kang-Sun and K. Sung-Jea. 2010. Low Complexity Content-Aware Video Retargeting for Mobile Devices. **Consumer Electronics, IEEE Transactions on** 56 (1): 182-189.
- Imagga. 2013. **Cropp.Me**. Available Source: <http://cropp.me/>,
- IMDb. (2010) **Manhunter(1986)**, Internet Movie Database Ltd.
- Itti, L. and C. Koch. 2001. Computational Modelling of Visual Attention. **Nat Rev Neurosci** 2 (3): 194-203.
- Jie, H., Z. Rongzhen and H. Zhiliang. 2003. Modified Fast Climbing Search Auto-Focus Algorithm with Adaptive Step Size Searching Technique for Digital Camera. **Consumer Electronics, IEEE Transactions on** 49 (2): 257-262.
- Jordan, N. and R. Schatz. 2006. Broadcast Television Services Suited for Mobile Handheld Devices, p. 55-55. **In Digital Telecommunications, , 2006. ICDT '06. International Conference on.**
- Jun, W., M.J.T. Reinders, R.L. Lagendijk, J. Lindenberg and M.S. Kankanhalli. 2004. Video Content Representation on Tiny Devices, p. 1711-1714 Vol.1713. **in Multimedia and Expo, 2004. ICME '04. 2004 IEEE International Conference on.**
- Kampe, A. and H. Olsson. 2005. A Dvb-H Receiver Architecture, p. 265-268. **In NORCHIP Conference, 2005. 23rd.**
- KANG Z M, Z.L. 2003. Implementation of an Automatic Focusing Algorithm Based on Spatial High Frequency Energy and Entropy, p. 552-555. **in ACTA Electronic SNICA.**
- Katz, S.D. 1991. **Film Directing Shot by Shot : Visualizing from Concept to Screen.** Michael Wiese Productions in conjunction with Focal Press, Studio City, CA.

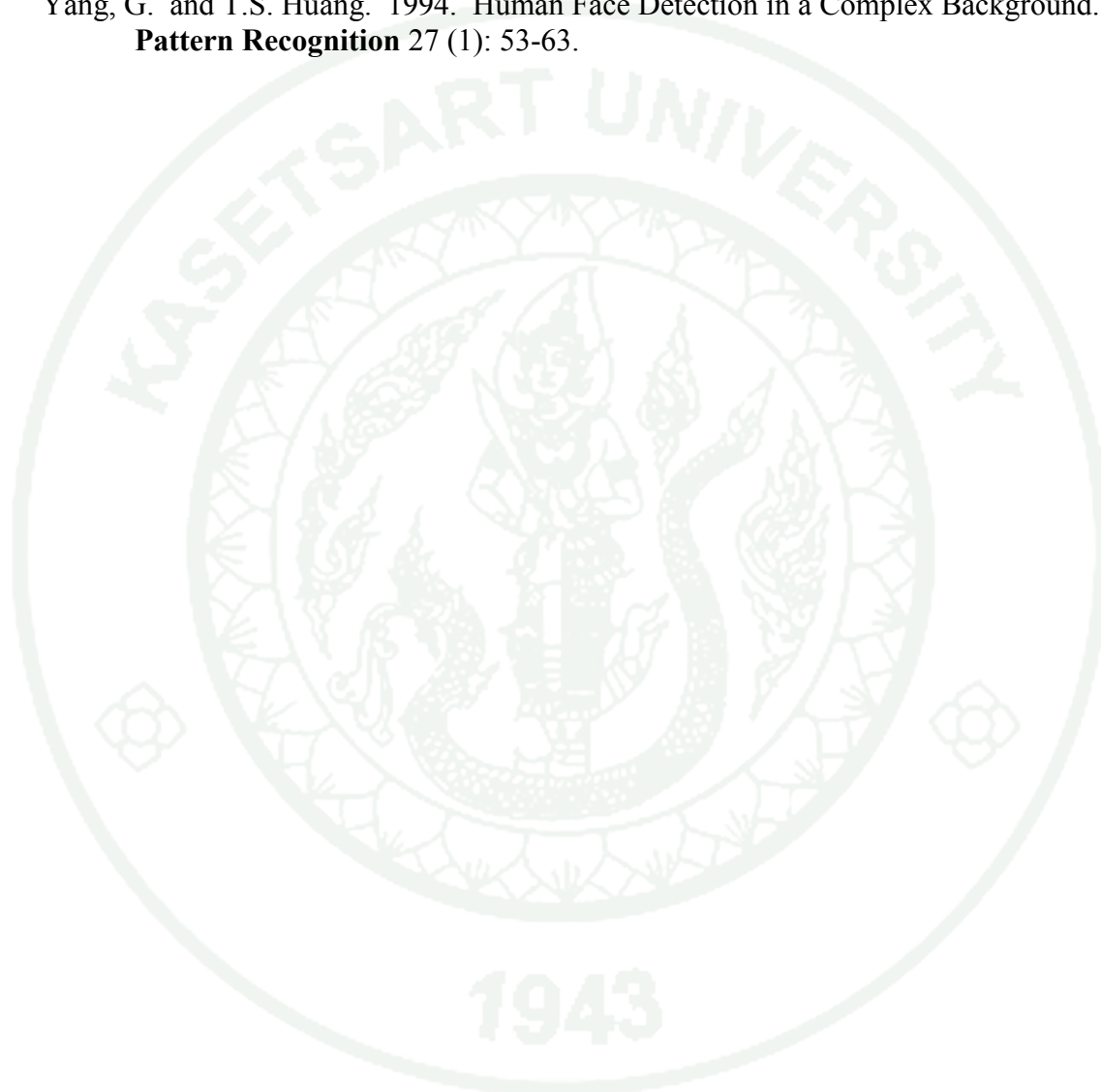
- Keogh, E.J. and M.J. Pazzani. 2000. Scaling up Dynamic Time Warping for Datamining Applications, p. 285-289. **In Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining.** ACM, Boston, Massachusetts, USA.
- Kharitonenko, I. and X. Zhang. 2000. Digital Focus Detector for Mobile Video Communicators. **Consumer Electronics, IEEE Transactions on** 46 (1): 233.
- Kopf, S., F. Lampi, T. King and W. Effelsberg. 2006. Automatic Scaling and Cropping of Videos for Devices with Limited Screen Resolution, p. **in Proceedings of the 14th annual ACM international conference on Multimedia.** ACM, Santa Barbara, CA, USA.
- Kopf, S., T. Haenselmann, J. Kiess, B. Guthier and W. Effelsberg. 2011. Algorithms for Video Retargeting. **Multimedia Tools Appl.** 51 (2): 819-861.
- Kornfeld, M. 2004. Dvb-H - the Emerging Standard for Mobile Data Communication, p. 193-198. **In Consumer Electronics, 2004 IEEE International Symposium on.**
- Lee, S.Y., S.S. Park, C.S. Kim, Y. Kumar and S.W. Kim. 2006. Low-Power Auto Focus Algorithm Using Modified Dct for the Mobile Phones, p. 67-68. **In Consumer Electronics, 2006. ICCE '06. 2006 Digest of Technical Papers. International Conference on.**
- Electronics, L. 2014. **Lg Optimus Vu** Available Source: <http://www.lg.com/th/mobile-phone/lg-P895-Optimus-Vu>,
- Lienhart, R., C. Kuhmunch and W. Effelsberg. 1997. On the Detection and Recognition of Television Commercials, p. 509-516. **In Multimedia Computing and Systems '97. Proceedings., IEEE International Conference on.**
- Liu, F. and M. Gleicher. 2005. Automatic Image Retargeting with Fisheye-View Warping, p. **in Proceedings of the 18th annual ACM symposium on User interface software and technology.** ACM, Seattle, WA, USA.
- Liu, F. and M. Gleicher. 2006. Video Retargeting: Automating Pan and Scan, p. **in Proceedings of the 14th annual ACM international conference on Multimedia.** ACM, Santa Barbara, CA, USA.
- Liu, H., X. Xie, W.-Y. Ma and H.-J. Zhang. 2003. Automatic Browsing of Large Pictures on Mobile Devices, p. **in Proceedings of the eleventh ACM international conference on Multimedia.** ACM, Berkeley, CA, USA.
- Ma, Y.-F. and H.-J. Zhang. 2003. Contrast-Based Image Attention Analysis by Using Fuzzy Growing, p. 374-381. **In Proceedings of the eleventh ACM international conference on Multimedia.** ACM, Berkeley, CA, USA.

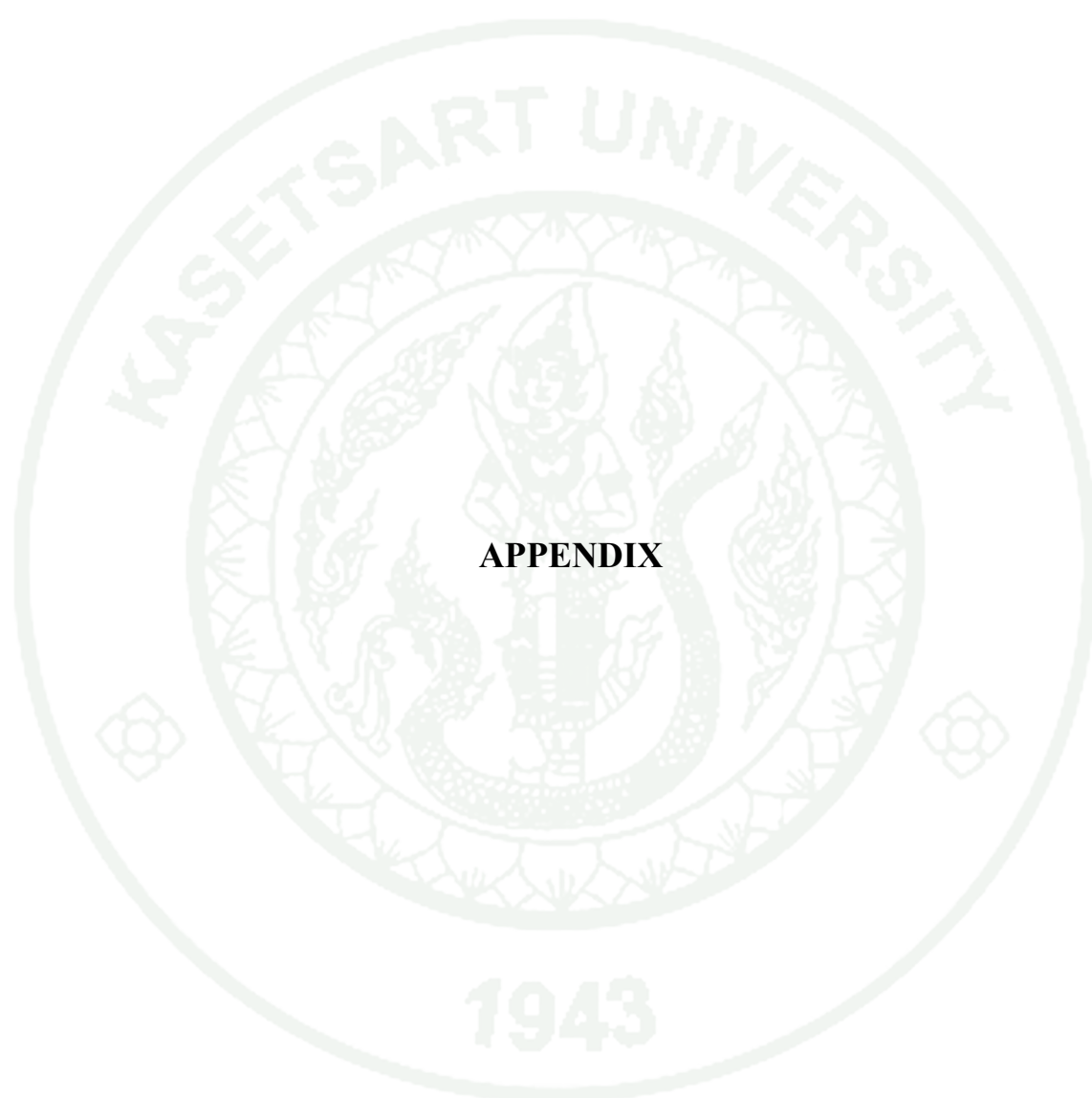
- Mamer, B. 2003. **Film Production Technique : Creating the Accomplished Image**. 3rd ed. Wadsworth/Thomson Learning, South Melbourne ; London.
- Ming-Hsuan, Y., D.J. Kriegman and N. Ahuja. 2002. Detecting Faces in Images: A Survey. **Pattern Analysis and Machine Intelligence, IEEE Transactions on** 24 (1): 34-58.
- Ming-Sui, L., S. Mei-Yin, C.C.J. Kuo and Y. Akio. 2007. Techniques for Flexible Image/Video Resolution Conversion with Heterogeneous Terminals. **Communications Magazine, IEEE** 45 (1): 61-67.
- Mingju, Z., Z. Lei, S. Yanfeng, F. Lin and M. Weiyang. 2005. Auto Cropping for Digital Photographs, p. 4 pp. *in* **Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on**.
- Mogul, J.C., Y.M. Chan and T. Kelly. 2004. Design, Implementation, and Evaluation of Duplicate Transfer Detection in Http, p. 4. *In* **Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1**. USENIX Association, San Francisco, California.
- Mohan, R. 1998. Video Sequence Matching, p. 3697-3700 vol.3696. *in* **Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on**.
- Ooi, K., K. Izumi, M. Nozaki and I. Takeda. 1990. An Advanced Auto-Focus System for Video Camera Using Quasi Condition Reasoning, p. 348-349. *In* **Consumer Electronics, 1990. ICCE 90. IEEE 1990 International Conference on**.
- Piamsa-nga, P. and N. Babaguchi. 2004. Motion Estimation and Detection of Complex Object by Analyzing Resampled Movements of Parts, p. 365-368 Vol. 361. *In* **Image Processing, 2004. ICIP '04. 2004 International Conference on**.
- Prince, S. 2004. **Movies and Meaning : An Introduction to Film**. 3rd. Pearson/Allyn and Bacon, Boston.
- SAMSUNG. 2014. **Galaxy Gear**. Available Source: <http://www.samsung.com/th/consumer/mobile-devices/galaxy-gear/>,
- Sánchez, J., X. Binefa, J. Vitrià and P. Radeva. 1999. Local Color Analysis for Scene Break Detection Applied to Tv Commercials Recognition, p. 237-244. *In* D. Huijsmans and A. M. Smeulders, eds. **Visual Information and Information Systems**. Springer Berlin Heidelberg.

- Sang Ku, K., P. Sang Rae and P. Joon Ki. 1998. Simultaneous out-of-Focus Blur Estimation and Restoration for a Digital Auto-Focusing System, p. 430-432. ***In Consumer Electronics, 1998. ICCE. 1998 Digest of Technical Papers. International Conference on.***
- Santella, A., M. Agrawala, D. DeCarlo, D. Salesin and M. Cohen. 2006. Gaze-Based Interaction for Semi-Automatic Photo Cropping, p. ***in Proceedings of the SIGCHI conference on Human Factors in computing systems.*** ACM, Montr\&\#233;al, Qu\&\#233;bec, Canada.
- Setlur, V., S. Takagi, R. Raskar, M. Gleicher and B. Gooch. 2005. Automatic Image Retargeting, p. ***in Proceedings of the 4th international conference on Mobile and ubiquitous multimedia.*** ACM, Christchurch, New Zealand.
- Shang, L., L. Yang, F. Wang, K.-P. Chan and X.-S. Hua. 2010. Real-Time Large Scale near-Duplicate Web Video Retrieval, p. 531-540. ***In Proceedings of the international conference on Multimedia.*** ACM, Firenze, Italy.
- Shivakumar, G.I.P.I.a.N. 1999. **Finding Pirated Video Sequences on the Internet.**
- SMPTE. 1994. **Rp 27.4-1994**
- SMPTE. 2003. **Rp 105-2003.**
- Spielberg, S. (1981) **Raiders of the Lost Ark.** pp. 115 min.
- Spielberg, S. (1984) **Indiana Jones and the Temple of Doom.** pp. 118 min.
- Spielberg, S. (1989) **Indiana Jones and the Last Crusade.** pp. 127 min.
- Subbarao, M., T. Choi and A. Nikzad. 1993. Focusing Techniques. **Journal of Optical Engineering** 32 2824-2836.
- Suh, B., H. Ling, B.B. Bederson and D.W. Jacobs. 2003. Automatic Thumbnail Cropping and Its Effectiveness, p. ***in Proceedings of the 16th annual ACM symposium on User interface software and technology.*** ACM, Vancouver, Canada.
- Sung-Hyun, Y., S. Sung-Min, H. Kuk-Tae, L. Hyoung-Soo and S. Bo-Ik. 2005. On-Chip Voice-Coil Motor Driver for Mobile Auto-Focus Camera Applications, p. 101-104. ***In Asian Solid-State Circuits Conference, 2005.***
- Tae-Kyu, K., S. Jae-Sik, L. Suk-Hwan, K. Ki-Ryong and K. Duh-Gyoo. 2006. Fast Auto-Focus Control Algorithm Using the Vcm Hysteresis Compensation in the Mobile Phone Camera, p. III-III. ***In Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on.***

- Turetsky, R.J. and D.P.W. Ellis. 2003. Ground-Truth Transcriptions of Real Music from Force-Aligned Midi Syntheses, p. 26-30. **In the Fourth International Conference on Music Information Retrieval.**
- Union, I.T. 2012. **Recommendation Itu-R Bt.500-13.**
- Viola, P. and M. Jones. 2001. Rapid Object Detection Using a Boosted Cascade of Simple Features, p. I-511-I-518 vol.511. **in Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on.**
- Viola, P. and M.J. Jones. 2004. Robust Real-Time Face Detection. **Int. J. Comput. Vision** 57 (2): 137-154.
- Wan-Lei, Z., W. Xiao and N. Chong-Wah. 2010. On the Annotation of Web Videos by Efficient near-Duplicate Search. **Multimedia, IEEE Transactions on** 12 (5): 448-461.
- Watkinson, J. 2001. **Convergence in Broadcast and Communications Media : The Fundamentals of Audio, Video, Data Processing, and Communications Technologies.** Focal Press, Oxford.
- Weibin, T., L. Yinghao, C. Zhicong, C. Lihuan, Y. Tao and G. Donghui. 2008. Auto-Focusing System for Microscope Based on Computational Verb Controllers, p. 84-87. **In Anti-counterfeiting, Security and Identification, 2008. ASID 2008. 2nd International Conference on.**
- Wheeler, L.J. 1969. **Principles of Cinematography : A Handbook of Motion Picture Technology.** (4th. Fountain press, London.
- Wright, S. 2006. **Digital Compositing for Film and Video.** 2nd ed. Focal Press, Burlington, Mass. ; Oxford.
- Wu, V.K.Y. and C. Polychronopoulos. 2012. Efficient Real-Time Similarity Detection for Video Caching and Streaming, p. 2249-2252. **In Image Processing (ICIP), 2012 19th IEEE International Conference on.**
- Wu, X., A.G. Hauptmann and C.-W. Ngo. 2007. Practical Elimination of near-Duplicates from Web Video Search, p. 218-227. **In Proceedings of the 15th international conference on Multimedia.** ACM, Augsburg, Germany.
- Xian-Sheng, H., C. Xian and Z. Hong-Jiang. 2004. Robust Video Signature Based on Ordinal Measure, p. 685-688 Vol. 681. **In Image Processing, 2004. ICIP '04. 2004 International Conference on.**
- Xiao, W., N. Chong-Wah, A.G. Hauptmann and T. Hung-Khoon. 2009. Real-Time near-Duplicate Elimination for Web Video Search with Content and Context. **Multimedia, IEEE Transactions on** 11 (2): 196-207.

- Xie, Q., Z. Huang, H.T. Shen, X. Zhou and C. Pang. 2012. Quick Identification of near-Duplicate Video Sequences with Cut Signature. **World Wide Web** 15 (3): 355-382.
- Xue, Y.-z., H. Du, Z. Liu and Z.-y. Zhang. 2011. Video Retargeting Using Optimized Crop-and-Scale. **Journal of Shanghai University (English Edition)** 15 (4): 331-334.
- Yang, G. and T.S. Huang. 1994. Human Face Detection in a Complex Background. **Pattern Recognition** 27 (1): 53-63.





APPENDIX



Publication

N-Gram Signature for Video Copy Detection

Paween Khoenkaw and Punpiti Piamsa-nga

Department of Computer Engineering, Faculty of Engineering, Kasetsart University,
Jatujak, Bangkok, 10900, THAILAND
{g4885027, pp}@ku.ac.th

Abstract.

Typically, video copy detection can be done by comparing signatures of new content with of known contents in database. However, this method requires high computation for both database generation and signature detection. In this paper, we proposed an efficient and fast video signature for video copy protection. The video features of a scene are extracted and then transformed to be a signature as a bit-wise string. All string signatures then are stored and manipulated by n-gram based text retrieval algorithm, which is proposed as a replacement with computation-intensive content similarity detection algorithm. The evaluation on the CC_WEB_VIDEO dataset shows that its accuracy is 85% where our baseline algorithms achieved only 75%; however, our algorithm is around 20 times as fast as the baseline.

Keywords: Similarity Measure, Video Matching, Video Search, Video Database

1 Introduction

Exponential growth of video contents on the internet makes cyber life uncomfortable for users to select the most appropriate contents; however, redundant uploads/posts of the same contents make internet much less enjoyable. Redundant post on the internet is usually related to violation to copyright laws and website policies as well. To prevent redundant uploads, video similarity detection algorithms have been proposed [1]. There are two approaches to tackle this problem: frame-by-frame comparison method and signature based method. Although frame-by-frame comparison achieves high accuracy, it requires exhaustive computation; therefore, it is impractical for long video. The signature based method is about to represent video contents by a smaller vector and then that vector is used to be compared instead of the original content; therefore, it is faster and more practical even though it may lose accuracy.

Early signature-based technique was about to digest video contents using strong hash algorithm, such as SHA [2]. This method is efficient for comparing videos that contained exactly the same contents; however, it is not practical for real application since there are many existing formats of video and the video files usually contains other additional information [3]. Many other techniques have been proposed as following literatures.

adfa, p. 1, 2011.

© Springer-Verlag Berlin Heidelberg 2011

There are many video features that can be used as video signature. Parts of video stream such as DCT coefficients extracted from MPEG streams can be used for internet video similarity detection [4]; however, this method is limited to MPEG encoded videos. Weighted color component [5], color histogram [6] and color with applying coherence vector [7] can be used as signature to detect the television commercials. Color deterioration in motion pictures usually degrades signature quality. Therefore, there are many proposed techniques to improve robustness of color. For example, Xian-Sheng et. al. proposed to down-sample video frame to create the temporal shape of relative intensity [8, 9] and Lifeng et. al. proposed to use local average of gray-level values for signature creation [10]. However, in many applications, color deterioration is a main trouble to video signature detection [8].

Other proposed techniques are based on high-level feature extraction, such as motion [3], shot durations, event length [11], camera behaviors [12-15], and object recognition [16, 17]. These types of signatures can represent its source videos; however, it suffers many problems. Color histogram is simple to extract but it is not robust to color deterioration. Motion is robust to color deteriorated but it required complex calculation to extract motion signatures, such as motion estimation and template matching. Combined features to create signature can gain more accuracy than the previous proposed ones but it required more computation power to prepare.

The previous research projects on signature matching were based on similarity measurement method between pair of signatures. Usually a signature is represented by a multidimensional vector of real numbers, which requires very high computation power in similarity measurement process.

In temporal dimension, it is very usual that the same video content may have different frame rates. Therefore, video similarity detection algorithm must be tolerant to small frame rate change. Typically, most algorithms use average values within sliding windows as signature representation or use dynamic programming in signature matching process [18]. However, this method requires high computation.

However, the longest processing time in signature detection is about similarity ranking. All methods in literatures need to compute similarity score between query and all records in database. The results are the highest ranked results. Therefore, the processing time is much worse if the database is larger.

In this paper, we propose an n-gram signature for video copy detection. All video sequences are transformed into a string representation, where each character represents features extracted from video. Then, all comparisons take advantages of text comparison, which is much faster than using multidimensional vectors of real numbers. N-gram comparison is alignment free, which is robust to a small number of insertions or deletions of a few characters in the string; therefore, it should be robust to small frame rate changes. We tested our proposed method on the a video database corpus (CC_WEB_VIDEO) [19]. We found that its accuracy is 85% where our benchmark is 75%. However, run time using by our algorithm is only 5% of the benchmark.

In following sections, we give brief introduction of N-gram in section 2. The proposed algorithm is described in section 3. Section 4 has details of experiments and their results. Finally, we conclude our wok in section 5

2 N Gram

N-Gram is a contiguous sequence of n items from a given sequence of data stream, such as text, speech and video. It is widely used in sequence matching algorithm on natural language processing [20, 21] and biological sequence analysis [22-24]. N-Gram is generated by cutting given sequence into small pieces with size varied from 1 to N , called gram; every possible n -gram in a sequence are stored in a table along with its frequency. The similarity between two sequences is about the similarity of frequencies of common n -grams. N-gram signature is robust to insertion, deletion and noise [24]. It is fast because it does not need to compare every possible n -gram and only high-frequency n -grams are required in matching process. Therefore, it requires less storage for n -gram signature.

3 Proposed Model

Our system is depicted in **Fig. 1**. First, features of interest are extracted from video stream. Second, those features are “textualized”, which is to transform features in multidimensional vector to string representation. Third, those strings are extracted as “grams” and then stored in a database as indices. The lower part of **Fig. 1** is about video querying. The query video has to be textualized and then used it to generate n -grams. The query n -grams are then used as video representatives to match in the video archive database.

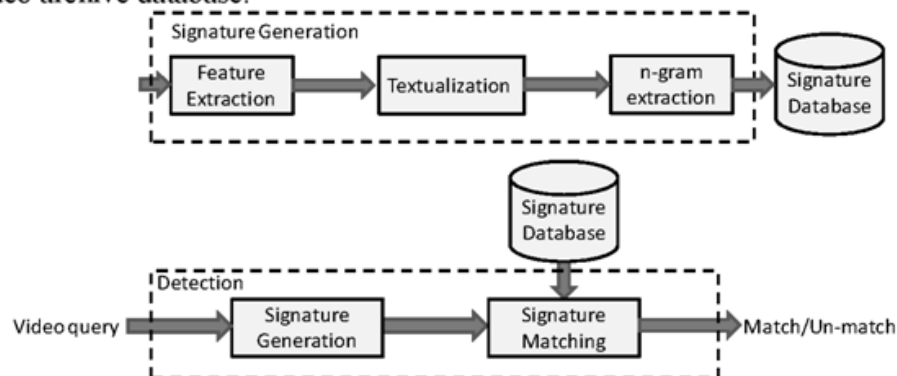


Fig. 1. proposed model of video copy detection.

3.1 Signature generation

In order to generate a good signature, features must be carefully selected. Many previous research projects used temporal differentials of feature vectors to create signatures. This kind of features can amplify “landmark event” in video such as motion of object or scene cut detection in video sequence. However, landmark event cannot be a signature but a composition of small details around landmark can. In order to avoid problems of color deterioration problem and sampling-rate heterogeneity, we propose to use “luminance velocity” as feature. The signature generation is explained

as follows. First, extract average luminance intensity of each video frame by converting image frame to gray scale and then compute average intensity of each frame to create a sequence of average luminance (I). Second, determine the “luminance velocity” by calculating the differences between adjacent elements of I using equation (1) where I_v is defined as “luminance velocity” and then its values are normalized to be in between $[0..1]$ by equation(2) where D is normalized luminance velocity vector.

$$I_v = \Delta I \quad (1)$$

$$D = \left(\frac{I_v - I_{v_{min}}}{I_{v_{max}} - I_{v_{min}}} \right) \quad (2)$$

Third, quantize the vector D into finite l levels, where $0 < l < 256$ (ASCII range), to create symbolic sequences that suitable to text processing algorithm. Practically, ASCII characters below 32 are the control code that can affect the text processing library in order to avoid that we suggest to shift the symbol to the range of the normal alphabet, where the first character is arbitrary c as described in equation (3) where sequence Q is a quantized vector.

$$Q = [D \times l] + c \quad (3)$$

Finally, a signature string Q represents a video clip. Then Q is chopped and each piece is called candidate gram. In our experiments only gram that has frequency greater than 2 can be stored in database as signature (S).

3.2 Signature matching

To match querying signature, we assume that if some parts of query are matched with some parts of video in database then the query is not an original video. We use signature generation method described in the previous section to create a signature of query. A query signature is still in the same format as of video archive, which is a set of n-grams associated with its frequencies. Actually, if an n-gram of the query is matched to the one in database, this query could assume to be video duplication; however, this brings low-recall results. We propose to use a selected set of k highest-frequent n-grams to detect the similarity. Our algorithm is described as follows and written in pseudo code shown in Algorithm 1.

Algorithm 1 Video query

<p>Input: k, query signature(S_t), signature database(S)</p> <p>Output: similar video(S_d)</p> <p>1: S_r=Select top k grams from S_t order by frequency, length(gram)</p> <p>2: FOR each <i>gram</i> of S_r</p> <p>3: S_d=S_d+ Select videos from S where grams=<i>gram</i></p> <p>4: ENDFOR</p>
--

First, generate n-grams of query video (S_t) by process in previous section. Second, select the k most-frequent n-gram where length of n-gram is priority if frequency is equal. These k n-grams are denoted as S_r . Third, every element in S_r is used to search

in database S using binary search. If a single n -gram in S_r is found in S , the result is a set of matched videos.

4 Experiments and Results

The experiments are divided into two parts: to determine the most appropriate number of quantization levels (l) and number of grams in video query (l); and compare our method with a baseline algorithm [16, 17]. We assume that our method should maintain accuracy where using less computation power. We used the well-known CC_WEB_VIDEO, which is a near-duplicate web video dataset. This dataset contains a collection of viral videos, which is searched by 24 popular queries from YouTube, Google Video and Yahoo. All 12,790 videos are varied in encoding format, frame rate, bit rate, frame resolution, color/lighting change, overlay with logo, text and content modification. All queries in dataset are shown in **Table 1** [17, 19, 25].

Table 1. Queries to build video collections

ID	Query	Number of Videos		Processing Time (Second)
		Total	Near-Duplicate	
1	The lion sleeps tonight	792	334	1.88
2	Evolution of dance	483	122	2.85
3	Fold shirt	436	183	0.39
4	Cat massage	344	161	0.21
5	Ok go here it goes again	396	89	1.87
6	Urban ninja	771	45	3.39
7	Real life Simpsons	365	154	1.07
8	Free hugs	539	37	2.89
9	Where the hell is Matt	235	23	0.84
10	U2 and green day	297	52	1.30
11	Little superstar	377	59	1.49
12	Napoleon dynamite dance	881	146	2.94
13	I will survive Jesus	416	384	0.38
14	Ronaldinho ping pong	107	72	0.65
15	White and Nerdy	1771	696	7.52
16	Korean karaoke	205	20	1.11
17	Panic at the disco I write sins not tragedies	647	201	2.71
18	Bus uncle (巴士阿叔)	488	80	1.70
19	Sony Bravia	566	202	1.42
20	Changes Tupac	194	72	1.15
21	Afternoon delight	449	54	1.45
22	Numa Gary	422	32	1.59
23	Shakira hips don't lie	1322	234	5.98
24	India driving	287	26	0.88
Sum		12790	3478	47.66

To determine a number of quantization level l , we measure the performance (accuracy, precision, recall, and F-measures) of different l values, ranged from 0 to 200. the details of this effect are shown in **Fig. 2**. We found that effect of l can be divided into three regions. The first region is where $l < 80$. In this region, the result has high recall but low precision and accuracy. The second region is where between $80 \leq l \leq 160$. In this region, the accuracy is stable but the precision and recall are highly varied. Value of l in this region can be selected depending on type of an application whether it required either precision or recall. The third region is where $l > 160$. In this region, the accuracy and precision is decreased where l is increased.

From the experiment, we consider the best value of parameter $l=150$ because it produced the highest accuracy and the highest precision. Then, we will use this parameter for the rest of experiments.

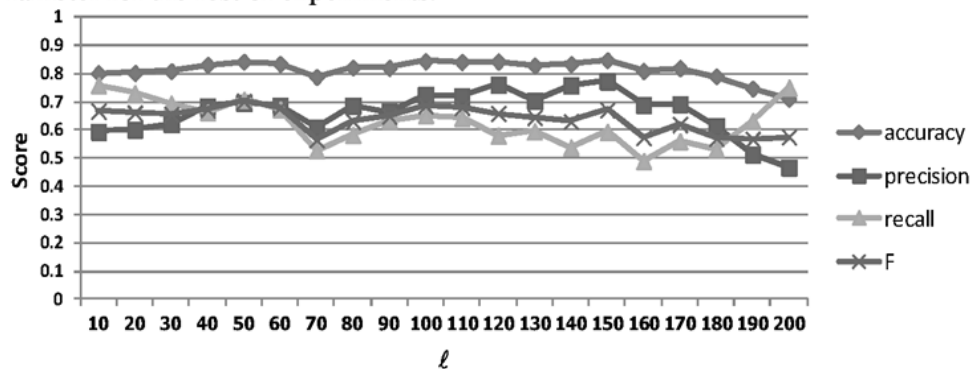


Fig. 2. Accuracy, precision, recall and F with different number of quantization levels (l)

To determine the most appropriate number of n-grams (k), we varied k from 2 to 70. The results are shown in **Fig. 3**. The result can be divided into three regions. The first region is where $k < 7$. We found that accuracy and precision is high but recall is low. Therefore, value of k in this region is appropriate to applications that require high precision. The second region is where $7 \leq k \leq 17$. In this region, accuracy is stable but precision and recall are converged. Therefore, this could be an optimal zone depending on types of applications. The third region is where $k > 17$. In this region, accuracy, precision and recall are stable. Therefore, $k > 17$ should not be used because the result will not get better and computation time varies linearly to k , shown in **Fig. 4**. From this experiment, we consider the best value of parameter $k = 8$ because it produces the highest accuracy while maintaining minimum computation and database load.

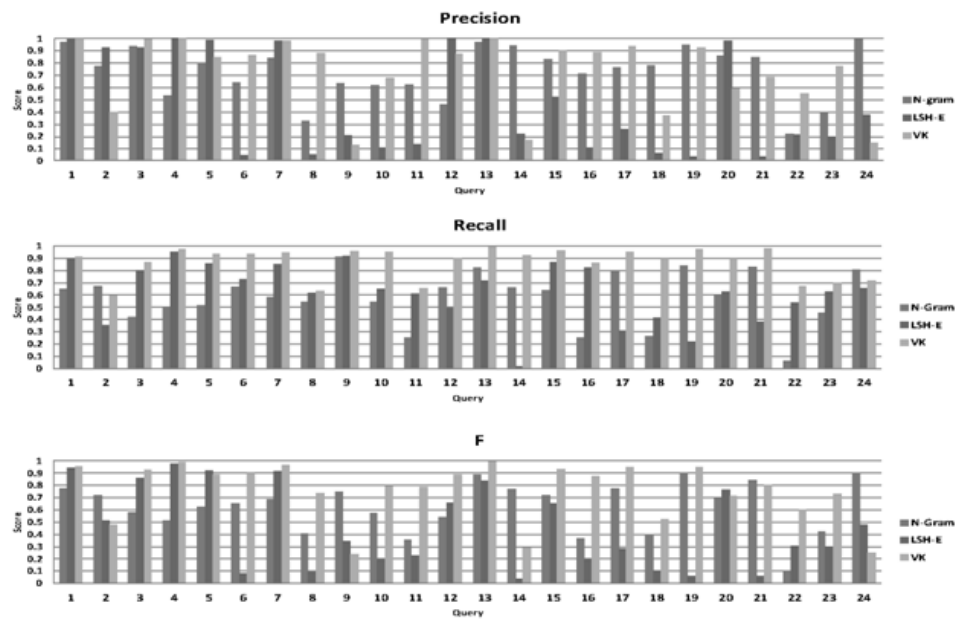


Fig. 5. Precision, recall and F-measure compared with LSH-E and VK

Table 2. Average results

Algorithm	Precision	Recall	F-measure
N gram	0.726	0.580	0.624
LSH-E	0.474	0.622	0.450
VK	0.732	0.868	0.757

The implementation was run on a Windows-2008 server of Intel Xeon E5620 machine with 96-GB memory. Signature database is implemented on MySQL server version 5.1.50-community. The signature data is stored in memory-type table. Both baseline and benchmark algorithms are run on the same environment. Our method is almost four times as faster as VK method for signature generation process and 16 times as faster as query process. For LSH-E, the signature generation time is pretty much the same with our method. However, LSH-E requires ranking with all items in database while our method uses binary search. From the experiments we concluded that our proposed algorithm is fast and good for query with simple scene or complex scene with minor editing or low quality videos. However, our algorithm is still intolerant to heavily-modified video, which requires high-level object detection to analyze.

5 Conclusion

In this paper, we proposed a video copy detection method based on signature approach. The luminance velocity was proposed to represent the video signature. The query method was adapted from text document similarity detection algorithm using

N-Gram algorithm. The evaluation was done with an internet video dataset that contains 12,790 near-duplicate videos which are results of 24 queries onto popular internet video sites. The result indicates that our method is fast and effective to detect near-duplicate videos, especially for simple scene or complex scene with minor editing and low quality or low bit-rate videos.

Our work can be adapted as a tool for eliminating redundant results in video query system and video piracy detection system. We are also currently advancing our algorithm for signature generation to gain more recall in complex scenes. Furthermore, we will enhance our method to target the online and real-time applications in the future.

6 References

1. <http://www.youtube.com>
2. Mogul, J.C., Chan, Y.M., Kelly, T.: Design, implementation, and evaluation of duplicate transfer detection in HTTP. Proceedings of the 1st conference on Symposium on Networked Systems Design and Implementation - Volume 1, pp. 4. USENIX Association, San Francisco, California (2004)
3. A, A.H., ho Hyun B, K., A, R.B.: Comparison of Sequence Matching Techniques for Video Copy Detection. (2000)
4. Wu, V.K.Y., Polychronopoulos, C.: Efficient real-time similarity detection for video caching and streaming. In: Image Processing (ICIP), 2012 19th IEEE International Conference on, pp. 2249-2252. (2012)
5. BaoFeng, L., HaiBin, C., Zheng, C.: An Efficient Method for Video Similarity Search with Video Signature. In: Computational and Information Sciences (ICCIS), 2010 International Conference on, pp. 713-716. (2010)
6. Sánchez, J., Binefa, X., Vitrià, J., Radeva, P.: Local Color Analysis for Scene Break Detection Applied to TV Commercials Recognition. In: Huijsmans, D., Smeulders, A.M. (eds.) Visual Information and Information Systems, vol. 1614, pp. 237-244. Springer Berlin Heidelberg (1999)
7. Lienhart, R., Kuhmunch, C., Effelsberg, W.: On the detection and recognition of television commercials. In: Multimedia Computing and Systems '97. Proceedings., IEEE International Conference on, pp. 509-516. (1997)
8. Xian-Sheng, H., Xian, C., Hong-Jiang, Z.: Robust video signature based on ordinal measure. In: Image Processing, 2004. ICIP '04. 2004 International Conference on, pp. 685-688 Vol. 681. (2014)
9. Cao, Z., Zhu, M.: An efficient video similarity search strategy for video-on-demand systems. In: Broadband Network & Multimedia Technology, 2009. IC-BNMT '09. 2nd IEEE International Conference on, pp. 174-178. (2009)
10. Shang, L., Yang, L., Wang, F., Chan, K.-P., Hua, X.-S.: Real-time large scale near-duplicate web video retrieval. Proceedings of the international conference on Multimedia, pp. 531-540. ACM, Firenze, Italy (2010)

11. Mohan, R.: Video sequence matching. In: Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on, pp. 3697-3700 vol.3696. (1998)
12. Shivakumar, G.I.P.I.a.N.: Finding pirated video sequences on the internet., (1999)
13. Xie, Q., Huang, Z., Shen, H.T., Zhou, X., Pang, C.: Quick identification of near-duplicate video sequences with cut signature. World Wide Web 15, 355-382 (2012)
14. Ardizzone, E., La Cascia, M., Molinelli, D.: Motion and color-based video indexing and retrieval. In: Pattern Recognition, 1996., Proceedings of the 13th International Conference on, pp. 135-139 vol.133. (1996)
15. Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Qian, H., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: the QBIC system. Computer 28, 23-32 (1995)
16. Dong, W., Wang, Z., Charikar, M., Li, K.: Efficiently matching sets of features with random histograms. Proceedings of the 16th ACM international conference on Multimedia, pp. 179-188. ACM, Vancouver, British Columbia, Canada (2008)
17. Wan-Lei, Z., Xiao, W., Chong-Wah, N.: On the Annotation of Web Videos by Efficient Near-Duplicate Search. Multimedia, IEEE Transactions on 12, 448-461 (2010)
18. Junfeng, J., Xiao-Ping, Z., Loui, A.C.: A new video similarity measure model based on video time density function and dynamic programming. In: Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on, pp. 1201-1204. (2011)
19. Wu, X., Hauptmann, A.G., Ngo, C.-W.: Practical elimination of near-duplicates from web video search. Proceedings of the 15th international conference on Multimedia, pp. 218-227. ACM, Augsburg, Germany (2007)
20. Dunning, T.: Statistical Identification of Language. Computing Research Laboratory. New Mexico State University (1994)
21. Khoo, C.S.G., Loh, T.E.: Using statistical and contextual information to identify two-and three-character words in Chinese text. J. Am. Soc. Inf. Sci. Technol. 53, 365-377 (2002)
22. Tomović, A., Janičić, P., Kešelj, V.: n-Gram-based classification and unsupervised hierarchical clustering of genome sequences. Computer Methods and Programs in Biomedicine 81, 137-153 (2006)
23. Pavlović-Lažetić, G.M., Mitić, N.S., Beljanski, M.V.: n-Gram characterization of genomic islands in bacterial genomes. Computer Methods and Programs in Biomedicine 93, 241-256 (2009)
24. Radomski, J.P., Slonimski, P.P.: Primary sequences of proteins from complete genomes display a singular periodicity: Alignment-free N-gram analysis. Comptes Rendus Biologies 330, 33-48 (2007)
25. Xiao, W., Chong-Wah, N., Hauptmann, A.G., Hung-Khoon, T.: Real-Time Near-Duplicate Elimination for Web Video Search With Content and Context. Multimedia, IEEE Transactions on 11, 196-207 (2009)

Video Similarity Measurement Using Spectrogram

Paween Khoenkaw and Pimpiti Piamsa-nga

Department of Computer Engineering, Faculty of Engineering,
Kasetsart University, Jatujak, Bangkok, 10900, THAILAND
{g4885027, pp}@ku.ac.th

Abstract—A new video similarity measurement method is developed for video copy detection. The ordinal feature is transformed to spectrogram by Short-Time Fourier Transform to represent as a video signature. Similarity between video signatures was measured by using DTW algorithm. The experiments on the CC_WEB_VIDEO dataset show that accuracy of this algorithm is 19.6%, 11.7%, and 16.6% as high as Sliding Window, DTW and STD methods, respectively in both “minor edited” and “extensively edited” video categories.

Keywords—Similarity Measure, Redundancy Detection, Near-Duplicate, Web Video, Video Matching, Video Search, Video Database, Spectrogram, Dynamic time warping

I. INTRODUCTION

Because popularity of video sharing on the Internet grows rapidly; problems of duplicated uploading and copyright violation become more important. Frequently, copied contents are just parts of other video; therefore, it is very hard for manual operation and many video similarity detection were proposed in order to eliminate the redundant video and protect violation on copyrighted video contents.

A. Video Signature

There are two approaches for video similarity measurements: frame-by-frame direct comparison and signature-based methods. Frame-by-frame comparison is very impractical for large database since it required exhaustive computation. Signature-based one is about to create a representation of video data by a small signature vector. This method is faster and practical since it compares two videos by small signatures instead of contents.

There are many signature-based techniques, such as length of video [1], motion vector [2], shot durations, camera behaviors and transition [3-6], and ordinal measurement [7-9]. Each technique has its own advantages and disadvantages. Video length is simpler but it must not be truncated or extended which cannot detect partial contents. Motion vector requires high computation and it cannot be used for majority-edited video. Analysis on shot durations, camera behaviors and transition is useful for video that has color or brightness deterioration but it cannot comprehend inserted un-related video content, such as commercial clips. Ordinal measurement is about to measure the visual correspondence between two images, which is robust to noise and color or brightness deterioration [8, 9].

B. Signature Comparison

Sliding window method was introduced to solve this problem in order to measure distances between many small video chunks rather than a full-length video. This method is feasible for comparing different length signatures and detecting a small video section that is a part of a long video, such as TV commercials [10].

However, sliding window does not allow frame deletion, frame insertion, and comparison between video clips that use different frame rate. To solve this problem, sequence alignment technique is introduced. The majority of alignment technique is based on Dynamic programming, such as Needleman-Wunsch algorithm [9], longest common subsequence [11] and Edit distance [12]. These algorithms are still effective only for short videos.

However, processing long video by dynamic programming still have pitfalls that it still takes long time if both video clips are quite long. The method proposed in [9] selects small chunks of video to compare as it does in sliding window and then add more chunks in dynamic programming style in order to compute similarity. Moreover, this method allows video insertion and deletion.

Generally, features of video are represented by vectors of real numbers, which is still computationally intensive. Therefore, the method in [13] proposed to use a sequence of discrete symbols extracted/transformed from video content and then compare them by string edit distance. This method improves accuracy, but it still requires exhaustive ranking [14].

To avoid exhaustive ranking, research in [15] proposed to use Laplace of Gaussian filter to extract a key-frame and use only key-frame as signature. Video key frame itself represents information in video shot; therefore, it eliminates redundancy. However, this method relied heavily on unreliable key-frame extraction process, bad tuning can result a signature that does not contain vital information.

We have proposed in our previous research [16] that alignment-free technique such as N-gram should be used in order to avoid exhaustive ranking. However, it is not truly successful for heavily content modified videos. In this paper, we propose a new measurement for video similarity, where the signature is more accurate and requires less computation than other video signatures.

This paper is organized as follows. We explain the proposed method in section II. Experiment and results are in

section III. Finally, conclusion and future works are described in section IV.

II. PROPOSED METHOD

We proposed a new compact signature, which is generated by transforming time-domain signature into frequency-domain. The overview of the proposed model is shown in Fig. 1. The proposed model is composed of two parts: signature generation and signature similarity measurement. Details are as follows.

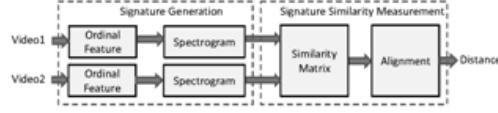


Fig. 1. Proposed model

A. Signature Generation

In this research we focus on creating video signature that is robust to color deteriorate, intensity deteriorates and different frame rate. Therefore, ordinal feature [7-9] is used since it can solve these problems.

The ordinal feature is extracted from a single video frame by partition image into k sub-image (Fig. 2A). Then, average intensity of each sub-image are calculated (Fig. 2B) and assigned an ordinal rank (Fig. 2C). This process is similar to [9]. We re-arrange ordinal rank matrix into a vector v (Fig. 2D). This process is repeated through all video frames and ordinal feature from every frame are concatenated to form the matrix M sized $k * l$ where l is a number of video frames.

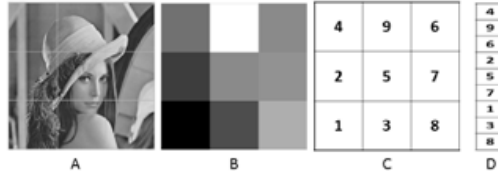


Fig. 2. The ordinal feature extraction process.

Considering row-wise vectors of matrix M , there are k vectors $x_1[n] \dots x_k[n]$ from each row from row 1 to row k . Each $x[n]$ is then used to generate spectrograms in order to reduce dimensions of signatures by transforming them into frequency domain. In order to preserve the time resolution a short-time Fourier transform: STFT is used. The STFT is a Fourier-related transform that is used to determine phase and frequency content of local section of a signal. The STFT is described in (1) where $x[n]$ is a signal to be transformed, $\omega[n]$ is window function. In this research "Hann window" is used and $X(m, \omega)$ is the Fourier transform of $x[n]\omega[n - m]$.

$$STFT\{x[n]\} \equiv X(m, \omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n - m]e^{-j\omega n} \quad (1)$$

The STFT signal is used to create time-frequency distributions called spectrogram [17]. It is widely used to visualize audio and speech signals. Spectrogram is estimated by computing the squared magnitude of the STFT of the signal using (2); the spectrogram is then used as a video signature.

By converting signal from time-domain to spectrogram, width of signal is increased from 1 into around $\frac{N}{2}$; however, length of signal is varied to window function (ω) and overlap size ($OverlapSize$). The length of this signature (l') is estimated by (3). In this research, we assume that $WindowSize \approx N$. Then, size of spectrogram is around $[\frac{N}{2} l']$.

$$spectrogram\{x[n]\}(m, \omega) = |X(m, \omega)|^2 \quad (2)$$

$$l' = \left\lceil \frac{l - WindowSize}{WindowSize - OverlapSize} \right\rceil + 1 \quad (3)$$

An example $x_l[n]$, which is a dimension original signal is shown in Fig. 3(A) and its spectrogram shown in Fig. 3(B).

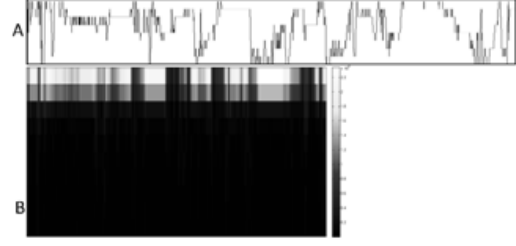


Fig. 3. A dimension of spectrogram signature example: (A) original signature; and (B) spectrogram

Then, we have a video signature which is composed of k spectrograms, where its length is shorter than time-domain signature. We describe how to compare our proposed signatures in the next section.

B. Signature Similarity Measurement

To measure similarity between two spectrogram signatures $D1$ and $D2$, we adopted a spectrogram-based method [18] for audio analysis application, which can handle signatures with different lengths.

First, similarity of pair k ($D1_k$ and $D2_k$) between two spectrograms are calculated by (4).

$$SM_k(i, j) = \frac{D1_k(i)^T D2_k(j)}{|D1_k(i)| |D2_k(j)|}, \text{ for } 0 \leq i < l'_1, 0 \leq j < l'_2 \quad (4)$$

Next, signatures from each dimension is aligned by **Algorithm 1** in order to acquire minimum cost. Costs from all dimensions are summed by (5) as a distance between two videos. The design of **Algorithm 1** is basically Dynamic Programming algorithm, where its alignment rule (Line 11) is from greedy penalty rules [8, 18].

$$d = \sum_{i=1}^k DynamicTimeWarping(1 - SM_{i, i}) \quad (5)$$

Examples of the lowest cost path determined by **Algorithm 1** are shown in Fig. 4. The lowest cost path of exactly duplicated videos is shown a perfect diagonal line. However, the more different videos, the longer cost path is.

Algorithm 1 Dynamic Time Warping

Input: similarity matrix(SM, k)
Output: minimum cost path (d)

```

1: [r c] = size(SM) // matrix dimensions
2: D(1,:) = ∞, D(:,1) = ∞
3: D(1,1) = 0
4: FOR i=1 to r
5:   FOR j=1 to c
6:     D(i+1,j+1) = SM(i,j)
7:   END FOR
8: END FOR
9: FOR i=1 to r
10:  FOR j=1 to c
11:    D(i+1,j+1) = D(i+1,j+1) + min(D(i,j), D(i,j+1), D(i+1,j))
12:  END FOR
13: END FOR
14: RETURN D(i+1,j+1)

```

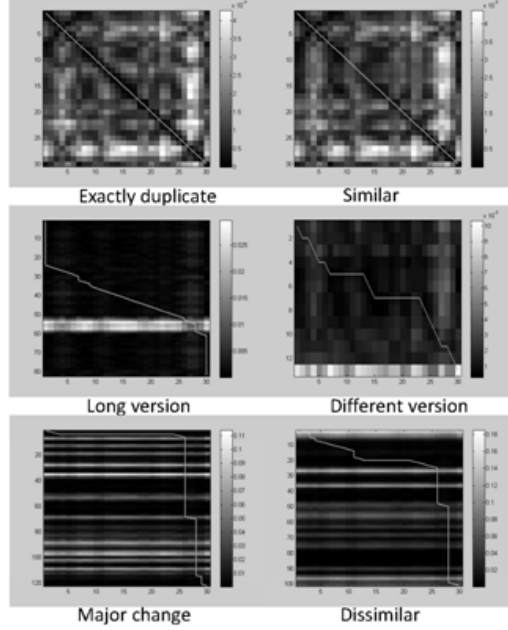


Fig. 4. The minimum cost path of different type of video comparison.

In case of two signals $x_1[n]$ and $x_2[n]$, whose lengths are l_1 and l_2 , respectively, the complexity of dynamic time warping algorithm is $O(l_1 l_2)$ [22]. Algorithm 1 performs faster by reducing size of input vectors. First, it transforms those two input signals into spectrograms $|X_1(m, \omega)|^2$ and $|X_2(m, \omega)|^2$, where their sizes are only $\lfloor \frac{N}{2} \rfloor \times \lfloor l_1' \rfloor$ and $\lfloor \frac{N}{2} \rfloor \times \lfloor l_2' \rfloor$, respectively. Then, size of similarity matrix SM is $\lfloor l_1' \rfloor \times \lfloor l_2' \rfloor$. Therefore, Dynamic time warping complexity of this algorithm is $O(l_1' l_2')$. Thus, the speedup is $O(l_1 l_2 / l_1' l_2')$.

III. EXPERIMENTS
A. Dataset

In this research, we use a benchmark dataset called CC_WEB_VIDEO: Near-Duplicate Web Video Dataset. This dataset contained a collection of viral video based on a sample of 24 popular queries from YouTube, Google Video and Yahoo! Video. All 12,790 videos are in various encoding formats, frame rates, bit rates, frame resolutions, lengths, frame insertion / frame deletion, color / lighting change, overlay with logos and text, and content modification [19-21]. Because each topic of the CC_WEB_VIDEO dataset has different number of duplicated videos; it is possible that result may be biased on number of duplications. Therefore, we normalize result before further evaluation.

The experiment is divided into two parts: 1) parameter tuning to determine the effect of each parameter used in this algorithm; and 2) evaluation of the ranking performance of this algorithm in different recall.

B. Parameter tuning

We have to do an empirical experiment in order to determine 1) the best dimension k for generating ordinal features 2) the best window size $\omega(n)$ for STFT; and 3) the best overlap size of STFT.

We determine a dimension size k of ordinal feature k in the signature generation process by varying k from 2x2 to 6x6 to find top-1 to top-20 ranking results (Fig. 5). We concluded that the 3x3 generated the highest precision.

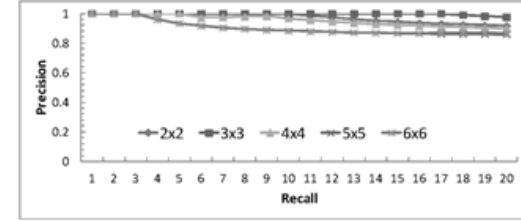


Fig. 5. The accuracy at different grid size

A window size $\omega(n)$ is determined by fixing k as 3x3 and varying size from 50 to 10,000 frames. Precision of ranking (average of top-1 ranking to top-20 ranking) is shown in Fig. 6. The parameter $\omega(n)$ also indicates time and frequency resolution of a spectrogram and signature length. Small window generates high resolution in time (long signature length) but low resolution in frequency and vice versa. From empirical results, we select 400 frames in further experiments.

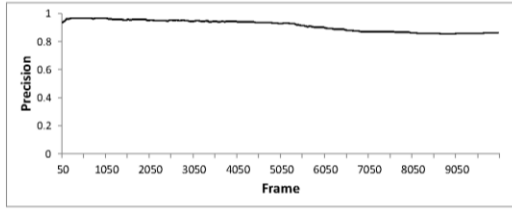


Fig. 6. Precision at different STFT window size

An *OverlapSize* is determined by fixing size $\omega(n)=400$, k as 3×3 and varying this parameter from 1 to 90% of window size. Precision of ranking (average of top-1 ranking to top-20 ranking) is shown in Fig. 7. The parameter *OverlapSize* indicates the overlap frame of STFT. Large overlap generates lower error rates but signature length is longer and vice versa. From empirical results, we select **81%** of window size in further experiments.

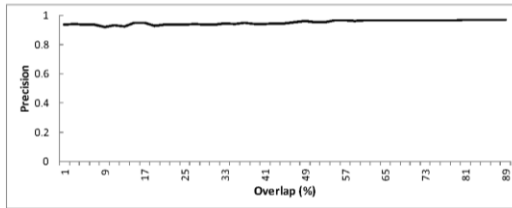


Fig. 7. Overlap window

C. Ranking performance

In this experiment, performance results are visualized on recall/precision plots since satisfaction of each user varies to types of applications. Therefore, a user may evaluate the performance by either recall or precision.

The results of this algorithm are compared with three algorithms: sliding window in [9], spatial-temporal distribution signature based (STD) in [8] and dynamic time warping on signature based on key-frame without spectrogram (DTW) in [15]. Performance on those three algorithms are shown in Recall/Precision graphs in Fig. 8(a), Fig. 8(b), and Fig. 8(c), respectively. Note that the nearer to top-right corner of recall/precision graphs (recall=1 and precision=1), the better results are.

Precision of sliding window and STD algorithm is inconsistent. Precision of many queries begin to degrade since recall=0.1. DTW algorithm produces higher precision than STD in most queries except the query numbers 6, 18, and 22. We further investigate this algorithm. Instead of using only key frame, we all frames as signature. The result is shown as DTW+. In Fig. 8(d), the overall precision is improved and only two queries get low-precision results. However, this

algorithm is very exhaustive since its signature is long. The length of DTW+ signature also longer than that of DTW. Finally, result of the proposed algorithm is shown in Fig. 8(e). Note that its overall precision is higher than of all reference algorithms. Results are very consistent for almost all queries except for queries 18 and 22. However, queries 18 and 12 are low-resolution videos and extensively edited videos, respectively.

Precisions of all topics when recall=1 generated by all five algorithms are shown in Fig. 8(f). It is clearly shown that precisions of this algorithm are better than those of reference algorithms in all queries. Average precision of all queries of this algorithm are 19.6%, 16.6%, 11.7%, and 4.4% as high as sliding window, STD, DTW, and DTW+ algorithms, respectively.

The overall precision over all recall levels are concluded in TABLE I. Average precision of this algorithm is higher than all reference algorithms. The DTW+ produced only 0.014 lower than this algorithm, but it is confirmed by paired t-test ($t=14.75$, $p=0$) that it is significantly different. Standard deviations also show this algorithm produced the most consistent results.

The average total signature length produced by all five algorithms are also shown in TABLE I. Length of ordinal feature is 9 ($k=9$) times of video length. Sliding window needs to compare the whole length since it does not compact the signature. DTW+ compacts data from 9 dimensions to 1 dimension so the total length is reduced 9 times of the original. DTW compacts up to 18.1 times. This algorithm compacts further up to 61.4 times. The STD generates smallest signature because it is not based on ordinal feature; however it also generated the lowest precision.

TABLE I. AVERAGE AND STANDARD DEVIATION OF PRECISION

Methods	Sliding window	STD	DTW	DTW+	This algorithm
Average of Precision	0.890	0.881	0.937	0.972	0.986
SD of precision	0.169	0.175	0.133	0.091	0.061
Average of total signature length	33991	1	1871	3777	553

D. Effects of video modification

We noticed that some types of video modification have major effects on performance. Those types are 1) minor-edited videos (queries #1, #13, #16, and #24), 2) extensively-edited videos (queries #10, #15, #22, and #23), and 3) low quality videos (query #18). Precisions at recall=1 of these groups are shown in Fig. 9. Precision of minor-edited videos from all five algorithms are almost similar except Sliding Window. The proposed algorithm produces the highest precision among all reference algorithms. This algorithm also produces the highest precision in extensively-edited videos. This algorithm cannot outperform DTW and DTW+ algorithms in the low quality video; however, our method produces much shorter signatures.

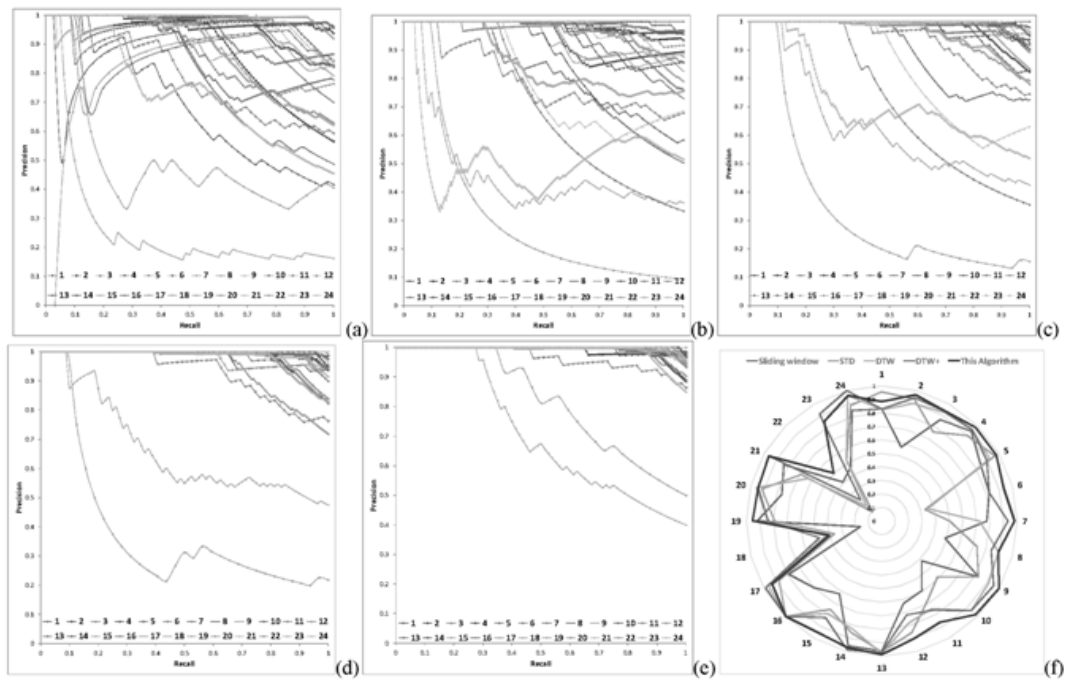


Fig. 8. Ranking performance: (a) sliding window; (b) STD; (c) DTW; (d) DTW+; (e) this algorithm; and (f) radar plot of all algorithms when recall=1

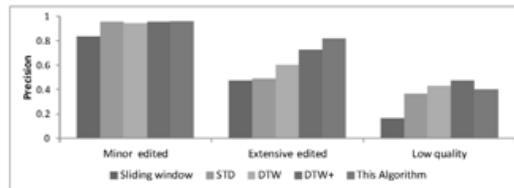


Fig. 9. precision compare between video groups at recall = 1

From the experiments, every algorithm achieves high precision at recall < 0.2 ; therefore, every algorithm is suitable for applications that do not require high recall rate. However, for applications that require both high precision and high recall, our algorithm outperforms other algorithms for edited videos. Moreover, our signature is also more compact than almost every reference algorithm.

IV. CONCLUSION

In this paper, we proposed a new video similarity measurement using spectrogram signatures. Signature is generated by the following steps. First, ordinal feature is extracted from each frame. Second, ordinal features of all frames are transformed by short-time Fourier transform. The transformed results are used to generate a spectrogram. Signature similarity measurement is designed by using pairwise similarity matrix and sequence alignment. The

alignment process is done by using dynamic time warping algorithm.

The dataset for result evaluation is CC_WEB_VIDEO. It is an internet video dataset that contains 12,790 videos in 24 queries. It contains near-duplicate video that commonly found on the internet. The result was clearly shown that our method is effective for ranking the similarity on large diversity of near-duplicate videos.

Our work can be applied to video query by example system. We are also currently advancing our algorithm for finding more compact, more efficient signature generation and detecting pirated video. The preliminary results are promising.

ACKNOWLEDGEMENT

This research was partially supported by Kasetsart University Research and Development Institute.

REFERENCES

- [1] R. Mohan, "Video sequence matching," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, 1998, pp. 3697-3700 vol.6.
- [2] A. H. A. K. ho Hyun B, and R. B. A, "Comparison of Sequence Matching Techniques for Video Copy Detection," ed, 2000.

- [3] G. I. P. I. a. N. Shivakumar, "Finding pirated video sequences on the internet.," 1999.
- [4] Q. Xie, Z. Huang, H. T. Shen, X. Zhou, and C. Pang, "Quick identification of near-duplicate video sequences with cut signature," *World Wide Web*, vol. 15, pp. 355-382, 2012.
- [5] E. Ardizzone, M. La Cascia, and D. Molinelli, "Motion and color-based video indexing and retrieval," in *Pattern Recognition, 1996., Proceedings of the 13th International Conference on*, 1996, pp. 135-139 vol.3.
- [6] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, H. Qian, B. Dom, et al., "Query by image and video content: the QBIC system," *Computer*, vol. 28, pp. 23-32, 1995.
- [7] D. N. Bhat and S. K. Nayar, "Ordinal measures for image correspondence," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, pp. 415-423, 1998.
- [8] L. BaoFeng, C. HaiBin, and C. Zheng, "An Efficient Method for Video Similarity Search with Video Signature," in *Computational and Information Sciences (ICCIS), 2010 International Conference on*, 2010, pp. 713-716.
- [9] H. Xian-Sheng, C. Xian, and Z. Hong-Jiang, "Robust video signature based on ordinal measure," in *Image Processing, 2004. ICIP '04. 2004 International Conference on*, 2004, pp. 685-688 Vol. 1.
- [10] N. Hu and R. B. Dannenberg, "A comparison of melodic database retrieval techniques using sung queries," presented at the Proceedings of the 2nd ACM/IEEE-CS joint conference on Digital libraries, Portland, Oregon, USA, 2002.
- [11] T.-S. C. Liping Chen, "A Match and Tiling Approach to Content-based Video Retrieval," in *IEEE International Conference on Multimedia and Expo*, 2001.
- [12] J. Zhou and X.-P. Zhang, "Automatic identification of digital video based on shot-level sequence matching," presented at the Proceedings of the 13th annual ACM international conference on Multimedia, Hilton, Singapore, 2005.
- [13] D. A. Adjeroh, M. C. Lee, and I. King, "A Distance Measure for Video Sequences," *Computer Vision and Image Understanding*, vol. 75, pp. 25-45, 7// 1999.
- [14] Z. Huang, H. T. Shen, J. Shao, X. Zhou, and B. Cui, "Bounded coordinate system indexing for real-time video clip search," *ACM Trans. Inf. Syst.*, vol. 27, pp. 1-33, 2009.
- [15] C. Chih-Yi, L. Cheng-Hung, W. Hsiang-An, C. Chu-Song, and C. Lee-Feng, "A Time Warping Based Approach for Video Copy Detection," in *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, 2006, pp. 228-231.
- [16] P. Khoenkaw and P. Piamsa-nga, "N-Gram Signature for Video Copy Detection," in *Recent Advances in Information and Communication Technology*, vol. 265, S. Boonkrong, H. Unger, and P. Meesad, Eds., ed: Springer International Publishing, 2014, pp. 335-344.
- [17] J. B. Allen and L. Rabiner, "A unified approach to short-time Fourier analysis and synthesis," *Proceedings of the IEEE*, vol. 65, pp. 1558-1564, 1977.
- [18] R. J. Turetsky and D. P. W. Ellis, "Ground-Truth Transcriptions of Real Music from Force-Aligned MIDI Syntheses," ed, 2003.
- [19] X. Wu, A. G. Hauptmann, and C.-W. Ngo, "Practical elimination of near-duplicates from web video search," presented at the Proceedings of the 15th international conference on Multimedia, Augsburg, Germany, 2007.
- [20] W. Xiao, N. Chong-Wah, A. G. Hauptmann, and T. Hung-Khoon, "Real-Time Near-Duplicate Elimination for Web Video Search With Content and Context," *Multimedia, IEEE Transactions on*, vol. 11, pp. 196-207, 2009.
- [21] Z. Wan-Lei, W. Xiao, and N. Chong-Wah, "On the Annotation of Web Videos by Efficient Near-Duplicate Search," *Multimedia, IEEE Transactions on*, vol. 12, pp. 448-461, 2010.
- [22] E. J. Keogh and M. J. Pazzani, "Scaling up dynamic time warping for datamining applications," presented at the Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining, Boston, Massachusetts, USA, 2000.

CIRRICULUM VITAE

NAME : Mr. Paween Khoenkaw

BIRTH DATE : September 14, 1979

BIRTH PLACE : Chiangmai, Thailand

EDUCATION	: <u>YEAR</u>	<u>INSTITUTE</u>	<u>DEGREE/DIPLOMA</u>
	2003	Rajamangala Univ.	B.Eng. (Computer Engineer)
	2005	Chiangmai Univ.	M.Sc. (Computer Science)

POSITION/TITLE :

WORK PLACE :

SCHOLARSHIP/AWARDS :