

國立臺灣大學電機資訊學院資訊網路與多媒體研究所

博士論文

Graduate Institute of Networking and Multimedia
College of Electrical Engineering & Computer Science


National Taiwan University

Doctoral dissertation

基於使用者興趣量表與節奏控制的

家庭音樂影片剪輯系統

MV-Style Home Video Editing System Based on
User Interests and Rhythmic Control

The logo of National Taiwan University is a circular seal. It features a central emblem with a book and a torch, surrounded by the university's name in Chinese characters: '國立臺灣大學' at the top and '愛國愛學勵品' at the bottom.

彭維廷

Peng Wei-Ting

指導教授：洪一平 博士

Advisor: Hung Yi-Ping, Ph.D.

中華民國 99 年 6 月

June, 2010

摘要

本論文目的是讓一般家庭使用者在最輕鬆的情況下，輸入他所拍攝的家庭影片以及一段他喜歡的音樂，系統就會自動結合此段影片與音樂並生成一段有節奏性的 MV(Music Video)。與以往的自動生成影片系統相比，本系統的特色在於使用一些剪接理論與美學的觀念，並且將其轉化成可行之演算法。此外，我們也加入心理學方面的研究，嚐試從使用者在觀賞影片時的生理反應，包括眼睛運動與表情，作為我們標記每段影片重要性的依據，並將其分析的數據轉成影片摘要的結果。最後將系統進一步用 UI 來呈現，嚐試讓使用者可以參與修改電腦最後分析的結果。也加入與以往商用剪接軟體不同的操作想法，企圖在剪接表現上創造不同的可能。

關鍵字 —興趣量表，媒體美學，影片摘要，臉部表情，眼球運動。

Abstract

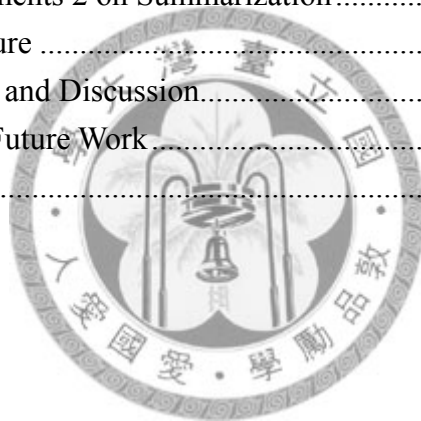
In this dissertation, we propose a novel home video editing system for generating music videos (MV) based on rhythmic control and the user interests. With the aid of rhythmic control from editing theories, the developed system is able to generate appealing and rhythmic music videos. We construct a module called “Interest Meter” to analyze variations of viewer’s blink rate, eye movement and facial expression when s/he watches unorganized raw home videos. This system transforms user’s behaviors into clues for determining important parts of video shots. Moreover, the friendly user interface allows novices to efficiently edit videos without difficulty. Experimental results show that this new editing mechanism can effectively generate music video summaries and can greatly reduce efforts of manual editing.

Keywords —Interest meter, media aesthetics, video summarization, facial expression, eye movement.

Contents

1.	Introduction.....	1
2.	Related Work.....	3
	2.1 From the perspective of information analysis.....	3
	2.2 From the perspective of audio-visual synthesis.....	5
	2.3 From the perspective of computer-human interaction.....	5
	2.4 Contributions of our system.....	6
3.	Observation and Inquiry.....	8
	3.1 Observation 1: Characteristic of Music Video.....	8
	3.2 Observation 2: Difficulty of Music Video Editing.....	10
	3.2.1 Establishing Video Rhythm is Difficult.....	10
	3.2.2 Repeat Cutting is Time Consuming Work.....	10
4.	System Framework.....	12
	4.1 Video and Music Analysis.....	12
	4.2 Interest Meter.....	13
	4.3 User Interface.....	13
5.	Video and Music Analysis.....	14
	5.1 Video analysis.....	14
	5.2 Music analysis.....	19
6.	Interest Meter.....	21
	6.1 Attention Model.....	22
	6.1.1 Head Motion Detection and Score Calculation.....	22
	6.1.2 Blinking and Saccade Detection.....	22
	6.1.3 Blinking Score Calculation.....	26
	6.1.4 Saccade Score Calculation.....	26
	6.1.5 Attention Score Calculation.....	26
	6.2 Emotion Model.....	28
	6.2.1 Facial Expression Recognition.....	28
	6.2.2 Emotion Score Calculation.....	29
	6.2.3 Interest Score Computing and Weighting Adjustment.....	30
7.	Summary Generation.....	32
	7.1 Rhythm Establishment.....	32
	7.2 Shot Trimming.....	34

	7.3 Transition Determination	35
8.	User Interface	37
	8.1 Video Editing	37
	8.1 Rhythmic Control.....	39
9.	Experimental Results	41
	9.1 Quality Estimation	42
	9.2 Evaluation of Interest Meter	43
	9.2.1 Accuracy of Iris Center Location	43
	9.2.2 Accuracy of Facial Expression Recognition	45
	9.2.3 Verification of Interest Meter	46
	9.3 User Study on Interface.....	47
	9.4 Experiments 1 on Summarization	52
	9.5 Experiments 2 on Summarization	54
	9.5.1 Procedure	55
	9.5.2 Results and Discussion.....	56
10.	Conclusions and Future Work	58
11.	Bibliography.....	60



List of Figures

Figure 1	Shot lengths in MVs.	9
Figure 2	Examples of rhythmic control.	10
Figure 3	The conventional process of shot cutting.	11
Figure 4	The proposed system framework.....	12
Figure 5	Different exposure situations.....	16
Figure 6	The velocity and acceleration of pan motion.....	19
Figure 7	The framework of Interest Meter.....	21
Figure 8	Detection of center eyeball.....	23
Figure 9	Iris refinement.....	24
Figure 10	Blink detection.....	25
Figure 11	The illustration of attention score.....	28
Figure 12	The process of facial expression recognition.....	29
Figure 13	The calculation of emotion score.....	30
Figure 14	Illustration of rhythm establishment.....	34
Figure 15	The illustration of how to choose an optimal shot clip.....	35
Figure 16	The User Interface of our system.....	37
Figure 17	The illustration of delete shot and retrieve it.....	38
Figure 18	The Illustration of putting the shot into temporal.....	39
Figure 19	Using mouse control to select the clip in shot and change its length.....	40
Figure 20	An example of rhythmic control.....	40
Figure 21	Keyframes of every shot in two videos.....	43
Figure 22	The ground truth and recognition result of facial expression.....	46
Figure 23	The average scores of participants when watching video 1.....	47
Figure 24	The average scores of participants when watching video 2.....	47
Figure 25	The experimental results of average scores of user experience.....	51
Figure 26	Editing time of participants.....	52
Figure 27	The rhythm scores of different editing system.....	54
Figure 28	The satisfaction and rhythm scores of three methods.....	57

List of Tables

Table 1	The mean brightness of three different exposure situations.....	15
Table 2	Function buttons.....	38
Table 3	Detection results of underexposed and blur shots.....	43
Table 4	False alarm for ordinary night scenes	43
Table 5	Performance comparison in terms of accuracy of eye detection.....	45
Table 6	The background of participants.....	48
Table 7	Within-Subjects and Between-Subjects effects.....	52
Table 8	The Evaluation data of Experiment 1.....	53
Table 9	The Evaluation data of Experiment 2.....	55



Chapter 1

Introduction

Digital cameras have become prevalent in modern households, and making a home video has become one of the most common ways to record important events such as parents' and children's birthday parties, friends' weddings, or family trips. These videos, which strengthen the ties between families and friends, can not only be stored on DVDs but also shared via community websites such as YouTube [1].

A common way to edit a home video is matching a video with a song that serves as background music, in order to synthesize them into a “music video (MV)”— a short video accompanied by a complete piece of music. As people have no time to watch the whole home videos, making a loosely organized and compact MV becomes a more attractive choice. MVs with appropriate length are suitable for browsing, and video summaries with background music are able to convey life experience.

Although shooting a home video is often enjoyable, editing videos is often tedious and troublesome. To conduct good editing, in addition to choosing appropriate and convenient software, user's basic background knowledge and media aesthetics are also essential [2][3][4]. Commercial video editing software such as

Adobe Premier [5], Sony Vegas [6], or Apple iMovie [7], is equipped with a variety of editing tools. However, for novice home users who are not major in filmmaking and editing, these tools can be more confusing than be helpful.

This paper proposes a novel MV editing system with a simple interface to make editing an MV-style home video easier than ever. Users only need to go through a few simple steps in order to create a rhythmic MV. We construct a module called “Interest Meter (IM)” based on psychological analysis to facilitate this task. IM monitors user’s reactions when watching a raw home video, such as his/her facial expressions, blinks, eye movements, and head motions, and identify which parts of video clips s/he might be interested in. These clips would then be chosen into the final output. Experimental results show that this MV editing system can significantly shorten the tedious editing process, and users will be able to easily make appealing home MVs with good video rhythm.

The rest of the dissertation is organized as follows. Related work is first reviewed in Chapter 2. Chapter 3 discusses the most frequent editing problems experienced by users and describes the foundation of the tailor-made design. The system’s framework is introduced in Chapter 4, and Chapter 5 describes the video and music analysis in our system. Chapter 6 presents the design of Interest Meter. The scheme of video summary generation is described in Chapter 7. Chapter 8 introduces the design of the user interface. Chapter 9 verifies that the system and IM are practical, and the conclusion and suggestions for future work are given in Section 10.

Chapter 2

Related Work

Shooting video is fun but editing is proven frustrating. Hence, users incline to put video footage on the shelf without further intention to elaborately editing. To ease video editing, video summarization has been studied for years. In this section, we survey related work from three different perspectives.

2.1 From the perspective of information analysis

Money and Agius [8] provide an extensive literature survey on video summarization. They classify related literature into three categories: (1) internal summarization techniques, (2) external summarization techniques, and (3) hybrid summarization techniques.

By definition, internal summarization techniques analyze internal information from video streams, which was produced during the production stage of the video lifecycle. These techniques extract low-level image, audio and text features to facilitate summarization, and are the most common summarization techniques [9][10][11][12][13]. External summarization techniques analyze external information

during any stage of the video lifecycle. There are two types of external information: (1) user-based information, which is information directly from users; and (2) contextual information, which is information not sourced directly from users or video streams. As for Hybrid summarization techniques, they analyze both internal and external information during any stage of the video lifecycle.

External information is collected when users view and interact with video content, and then this information is analyzed to develop video summaries. Money et al. [14] develop a video summarization technique by analyzing a range of user physiological response measures, including electro-dermal response (EDR), respiration amplitude (RA), respiration rate (RR), blood volume pulse (BVP) and heart rate (HR). Joho et al. [15] present an approach on affective video summarization based on viewer's facial expressions. Our previous work [16] analyzes variations of viewer's eye movement and facial expression when he or she watches the raw home video, and transforms these behaviors into clues of determining important part of each video shot. Compare with other similar work, we propose a framework to explore the impact of user's viewing behaviors on video editing. In our investigation, when viewers watch videos, they don't always have significant facial expression. That's the reason we also combine eye movement to determine important parts of videos. In this work, we enhance our previous work by integrating a module called Interest Meter with a friendly user interface. Moreover, we have significant improvement on eye movement information extraction, which achieves superior accuracy to other methods.

2.2 From the perspective of audio-visual synthesis

Typically speaking, there are two methods to synthesize music with a video: video-centric and music-centric. In a video-centric method, the music is dubbed in based on the content features of the video. For example, Mulhem et al. [9] developed a pivot vector space method that can automatically pick the best audio clip from a database to mix with a given video shot. In a music-centric method, various video clips are edited to match a complete song. This is the approach of the music videos commonly seen in the music industry. To produce an MV, a song's beat and tempo have to be analyzed before editing to produce a video rhythm that perfectly matches the music. Foote et al. [10] presented methods for the automatic creation of music videos. Hua et al. [11] proposed another segment-based matching method for home videos. Yoon et al. [12] used computable characteristics of a video and music to promote coherent matching. Wang et al. [13] proposed both video-centric and music-centric algorithms to synthesize music with video.

2.3 From the perspective of computer-human interaction

In addition to categorizing editing processes of music videos by audio-visual synthesis methods, from the perspective of computer-human interaction we can also classify them as follows: (1) manual, (2) fully automatic, and (3) semi-automatic. Most commercial editing software programs [5][6][7] are manual. Although they provide a wide variety of functions, editing a video on a manual software system can still be difficult even for experts, and much more so for a novice. Fully automatic video editing systems such as editing systems [10][11][12] and the automatic editing

software Power Direct [17] can render music videos through their built-in algorithms. Although they take much less time, users are not able to make changes when they are not satisfied with the results.

Wang et al. [13] proposed a dynamic-programming based algorithm for the automatic or semi-automatic generation of personalized music videos, but they did not include a user interface to help users edit the video. Shipman et al. [18] proposed the Hyper-Hitchcock program, which includes a user interface and various semi-automatic techniques to generate hyper video summaries. However, this approach is still not able to generate music videos. The semi-automatic software program MuVee [19] contains an automatic editing algorithm and a user interface so that users can adjust the output results of a music video.

2.4 Contributions of our system

We propose a novel home video editing system for generating music-centric MV. With external summarization techniques which analyze the user's attention and emotional response, our system can automatically select important parts of raw video shots that users are interested in. The main contributions of our system are as follows:

- A transfer function is used to determine the length of each shot in a music video, and establishes a video rhythm to enable music-video composition. This algorithm is not only faster than others, but is also more intuitive.
- A psychometric model is incorporated into the video summarization program to create a novel human-centric system.

- We have significant improvements in eye feature extraction, and our method achieves superior accuracy compared to other methods.
- The editing tools and design of the user interface can make video editing more convenient, allowing the user to avoid repetitive tasks. We also provide different video editing processes including manual and automatic levels.



Chapter 3

Observation and Inquiry

Prior to designing our system, we first examine the factors necessary for a successful MV. In addition to appealing composition and emotionally engaging content, what else is pivotal? In addition, we also have to identify the difficulties a novice may encounter so that our design can address these problems.

3.1 Observation 1: Characteristic of Music Video

Three MVs were analyzed in this study. Figure 1 shows relationship between music tempos and video shot lengths. The first song is Rihanna's "Umbrella," the MTV Video of the Year in 2007. The music is a rather fast-tempo song, and its MV has 504 shots within 4 minutes and 4 seconds, with the average shot length 0.48 seconds. The second song is Taylor Swift's "Love Story," which reached number 2 on the list of Top Downloaded Songs in iTunes Store. This MV features romantic images and a fast, light tempo. The video is made up of 190 shots in 3 minutes and 54 seconds, with the average shot length 1.23 seconds. These two songs express amorous ambiance of love. The last song is Norah Jones' "Come Away with Me," which hit number one on the U.S. Billboard 200 and won Album of the Year at the

Grammy Awards in 2002. The smooth tempo of this song is suitable for a family-trip MV. There are 90 shots in this 3-minute-11-second MV, with each shot lasting 2.1 seconds on average.

We will describe in later sections how the music tempos in Figure 1 were calculated. This figure shows that the shot length and music tempo tend to be inversely proportional to one another. That is, the faster the tempo of the song, the shorter the shot is. This is called “Rhythmic Control” in editing theory [2].

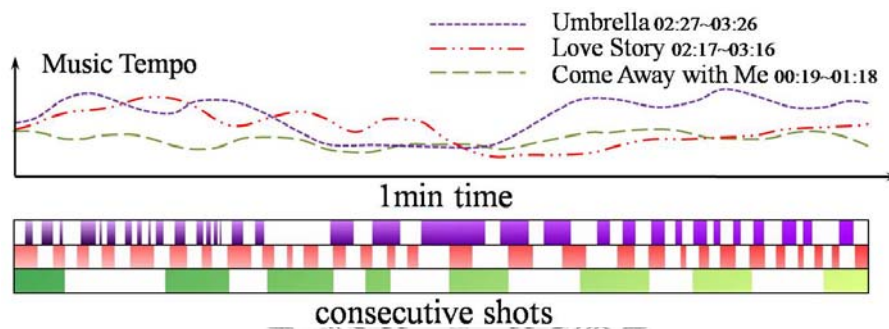


Figure 1. Shot lengths in three MVs and their corresponding music tempo curves.

Video rhythm refers to the pace of the cuts, and is similar to music tempo [4]. To create a more complex effect, brief shots can be juxtaposed with lengthy shots. Combining shots of different lengths can be used to establish different video rhythms. In video editing, we can vary lengths of consecutive shots to drive different rhythms. Based on temporal variations of music tempo, video rhythm can be altered by changing lengths of successive shots. When music tempo and video rhythm parallel one another, the entire video structure becomes unified and stable. The implementation of rhythmic control is described in later sections. Figure 2 shows how to perform rhythmic control where the shot length is manipulated for different rhythms.

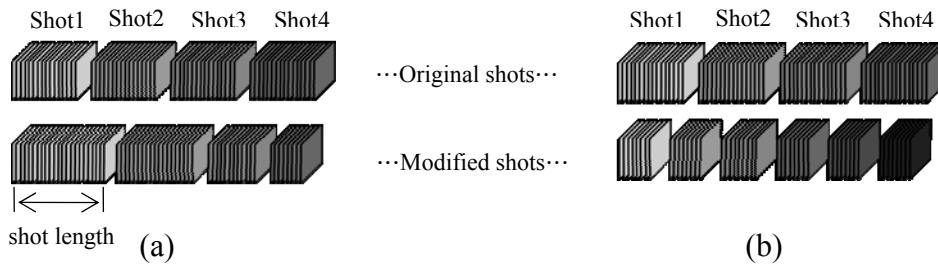


Figure 2. Examples of rhythmic control. (a) The first row shows successive shots that have the same length, and the second row shows shots with adjusted lengths. This process varies visual rhythm. (b) The first row shows that we can cut loosely successive shots to moderate video rhythm, and the second row shows that we can cut tightly successive shots to speed up video rhythm.

3.2 Observation 2: Difficulty of Music Video Editing

Although a music video is usually of a reasonable length, it still requires a lot of editing work to cut the video to the right length and to make it vividly express the event by matching the tempo of the chosen song. The followings are the most common difficulties encountered by a novice user.

3.2.1 Establishing Video Rhythm is Difficult

Manipulating shot lengths to establish video rhythm is a feature of MVs, and these techniques are not easy for a novice to grasp. For instance, in shot transition, the shot boundary has to fall exactly on the musical beat for a good video rhythm. To a professional editor, a beat is a plot point – where something happens or changes in a scene.

3.2.2 Repeat Cutting is Time Consuming Work

To make a music video, users of conventional non-linear video editing software must first cut video clips according to time lines and then link up various edited shots before rendering a final MV. The “cut” refers to establishing the “in” and “out”

points from a raw shot and combining a cut shot clip to make a music video. Figure 3 demonstrates the process of video shot clip cutting. In this procedure, changing any of the shot clips or shot lengths is troublesome work, as the raw shots have to be cut again and each shot length has to be adjusted to match the total length of the song. Users quickly tire of this process after a few iterations.

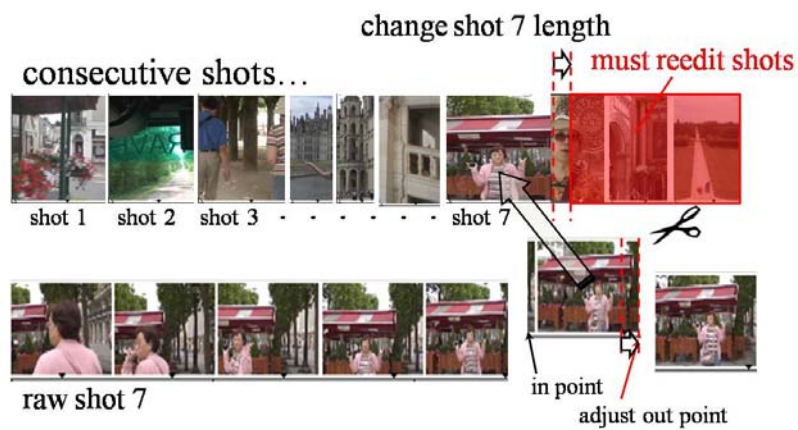


Figure 3. The conventional process of shot cutting. Cut the in and out point from a raw shot and insert a shot clip to music video.



Chapter 4

System Framework

Figure 4 demonstrates the system framework, including video, music analysis, Interest Meter and user interface.

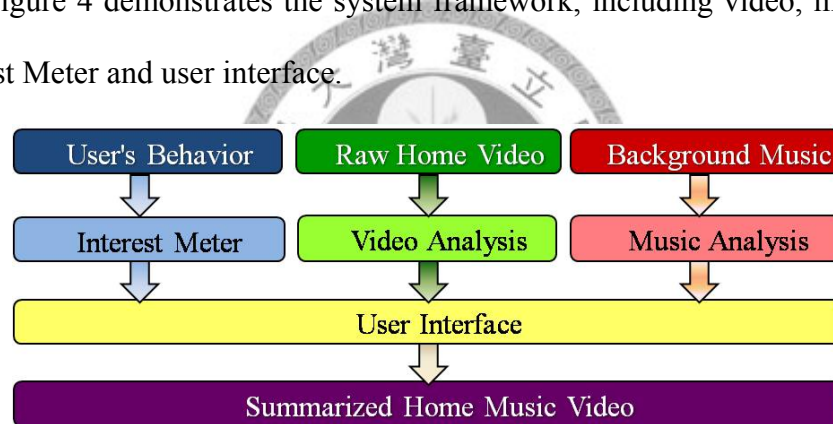


Figure 4. The proposed system framework.

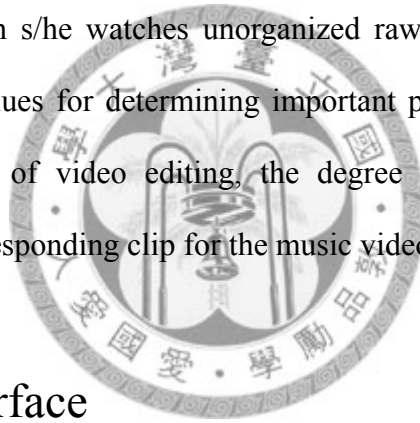
4.1 Video and Music Analysis

Our system incorporates elaborate video clips with music by carrying out analyses from both the video and music perspectives. In the video analysis, we first drop bad video shots that are poorly lit or blurred. Then, we segment the video into shots and display these shots on a user interface. For the background music, we estimate the tempo information based on the frequency of onsets (onsets generally occur when there is significant energy changes). We integrate them on the basis of

guidelines of media aesthetics. More details of video and music analyses can be found in [20].

4.2 Interest Meter

Argyle [21] indicated that users will show their interest with the following reactions: laughing, more fixed gazes, fewer blinks, and lively shoulder movements and nods of the head. Based on this theory, we construct a component called “Interest Meter” to analyze variations of viewer’s blink rate, eye movement and facial expression when s/he watches unorganized raw home videos. It transforms these behaviors into clues for determining important parts of each raw video shot. From the perspective of video editing, the degree users’ interest indicates the importance of the corresponding clip for the music video.



4.3 User Interface

In most cases, it is difficult for a novice to learn the various functions of commercial editing software. Thus, we design a user interface to display the analytical results of a home video and the background music, enabling users to easily construct a video with good rhythm simply by dragging and dropping. If they wish to change the content of shot clips, users can simply select the clips they like rather than laboriously cutting the video. These methods should enable a novice to complete the editing task within a short period of time.

Chapter 5

Video and Music Analysis

Given the input video V and the background music M , this chapter describes video and music analysis in our system. The results of these processes are the material for our automatic video editing method.

5.1 Video analysis

In order to extract the most favorable part of an input video V , we perform the following video analyses, including frame quality estimation, shot change detection, motion analysis, and face detection.

Quality Estimation. Firstly, we filter out ill-quality frames of the input video. Blur, overexposure, and underexposure effects are detected in this work.

When blur occurs, edges in video frames become indistinguishable [40]. In our system, we use a Laplacian filter to obtain edge intensities, and utilize them to achieve blur detection. The total edge strength over all pixels of a particular frame is regard as a threshold to determine whether this frame is blurred or not.

To detect overexposure or underexposure frames, we calculate mean brightness

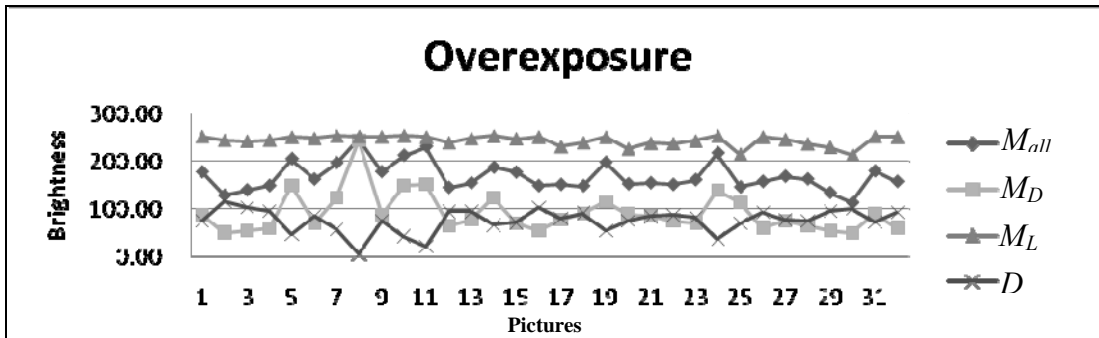
M_{all} of all pixels in a frame. Moreover, M_L and M_D values are calculated, which are top 10% lightest pixels and top 10% darkest pixels, respectively. In overexposure frames, most pixel values are over lit and have high brightness. So we consider a frame overexposure if both M_{all} and M_D exceed some predefined thresholds.

On the contrary, most pixels are dark in underexposure frames, and M_{all} should be low in this case. However, night scenes images share similar light conditions with underexposure ones. To distinguish them, we further consider the difference D between M_L and M_D . For night scenes, there would be some bright pixels such as bulbs or street lamps. Therefore, the difference between M_L and M_D can be used to distinguish night scenes images from underexposure ones.

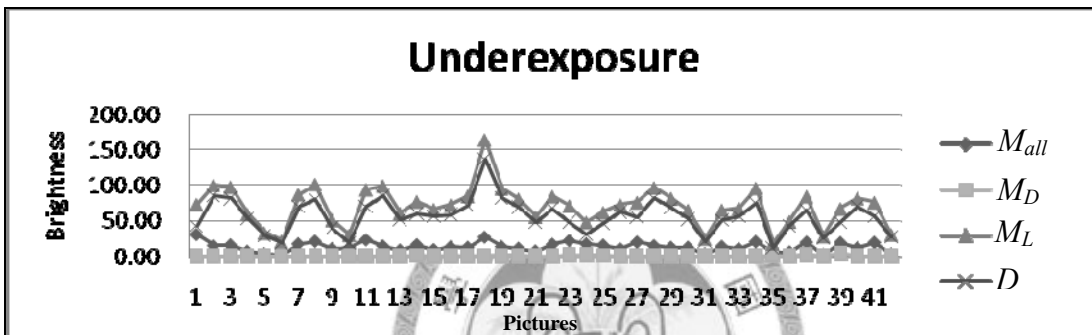
In this work, we respectively collect pictures which are in overexposure, underexposure and night scene situations. The value of M_{all} , M_D , M_L and D are calculated separately and shown in Figure 5. The mean brightness and its standard deviation of three different exposure situations are listed in Table 1. The thresholds used to classify exposure situations can be decided by these values.

Table 1. The mean brightness and it's standard deviation of three different exposure situations.

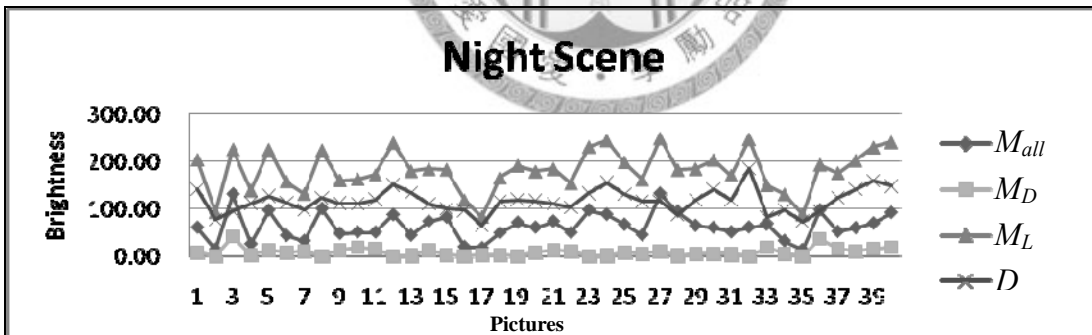
#	Overexposure				Underexposure				Night Scene			
	M_{all}	M_D	M_L	D	M_{all}	M_D	M_L	D	M_{all}	M_D	M_L	D
	v1	v2	v3	v4	v5	v6	v7	v8	v9	v10	v11	v12
Mean	168.4	90.3	243.3	74.9	13.9	0.5	70.5	56.6	63.1	8.4	180.0	116.8
STD	30.3	41.9	10.7	24.8	6.6	0.6	28.3	24.4	29.3	8.2	42.0	23.2



(a)



(b)



(c)

Figure 5. The value of M_{all} , M_D , M_L and D in different exposure situations: (a) overexposure (b) underexposure and (c) night scene.

For the frame f^i , we calculate the corresponding brightness information described above, i.e., M_{all}^i , M_D^i , M_L^i and D^i , and then examine f^i by the following algorithm:

Definitions:

D^i : difference between M_L^i and M_{all}^i

T_{all_high} : Threshold of mean brightness for testing overexposure. The reasonable value is 150~170 (Determined from v1).

T_{all_low} : Threshold of mean brightness for testing night scene and underexposure. The reasonable value is 60~70 (Determined from v9).

T_{D_high} : Threshold of 10% darkest pixels for testing overexposure. The reasonable value is 70~90. (Determined from v2)

T_{L_low} : Threshold of 10% lightest pixels for testing night scene and underexposure. The reasonable value is 170~190 (Determined from v7 and v11).

$T_{distance}$: Threshold of D for testing night scene and underexposure. The reasonable value is 90~100 (Determined from v8 and v12).

Algorithm:

```

01      if  $M_{all}^i > T_{all\_high}$ 
02          if  $M_D^i > T_{D\_high}$ 
03              then  $f^i \leftarrow$  overexposure, dropped the frame  $f^i$ 
04              else  $f^i \leftarrow$  normal
05      elseif  $M_{all}^i < T_{all\_low}$ 
06          if  $M_L^i < T_{L\_low}$ 
07              if  $D^i > T_{distance}$ 
08                  then  $f^i \leftarrow$  night scene
09                  else  $f^i \leftarrow$  underexposure, dropped the frame  $f^i$ 
10              else  $f^i \leftarrow$  normal
11      else  $f^i \leftarrow$  normal

```


How a video shot be dropped due to bad quality is decided by the following strategy. If a frame is detected as being blurred, overexposure, or underexposure, we label this frame as a bad-quality frame. If the maximum duration of successive good frames in a shot is shorter than one second, this shot is dropped from the input video.

Video Segmentation. To segment the home video V into clips, we just use the most well-known histogram-based shot change detection [41], since shot changes mostly occur with sudden cuts, instead of special transitions like dissolve or fade. We calculate hue-saturation-value (HSV) histograms for frames, and determine shot changes by detecting sudden changes of histograms between two consecutive frames. After dropping ill-quality frames and detecting shot change, we segment the input video V into N_{shot} filtered shots:

$$V_{good} = \{shot_i : i = 1, \dots, N_{shot}\} \quad (1)$$

Motion Analysis: We perform motion analysis to determine camera motion types (pan, tilt, zoom, or still) with an optical flow approach [42]. In addition to camera motion types, directions and magnitudes, we advocate that camera motion acceleration should be considered in video editing. In our work, velocity of camera motion is defined as the rate of change of position between adjacent frames. It is measured by pixels per frame. Motion acceleration is the change in velocity over frame. If motion acceleration varies frequently and significantly (see Figure 6), the corresponding video segments are usually annoying and are less likely to be selected in the automatic editing phase. Acceleration means that camera tends to speed up or slow down. Larger acceleration means more vibration.

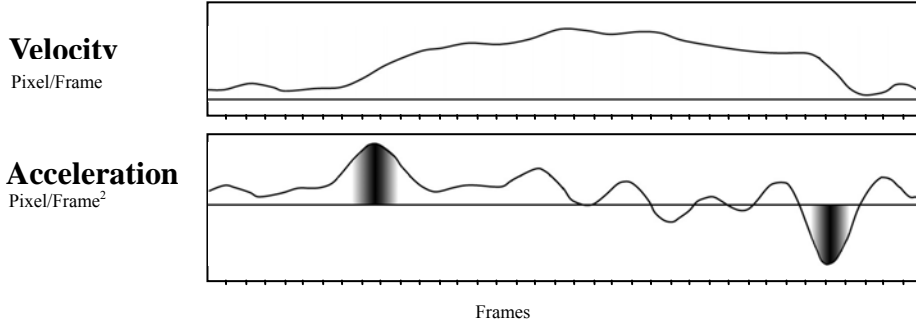


Figure 6. Top: The velocity curve of pan motion. Bottom: The acceleration curve for the same motion. The most annoying part of the clip is in the beginning and end of the motion. The darker regions mean larger acceleration.

5.2 Music analysis

To coordinate visual and aural presentation, shot changes need to be in conformity to music tempo. So we estimate music tempo of the background music M at this stage.

Onset Detection: We first detect onsets based on energy dynamics. Onsets generally occur when there is significant energy change. We apply the Fourier transform with a Hamming window $w(m)$ to M . The k th frequency bin of the n th audio frame, $F(n, k)$, of the background music M can be described as:

$$F(n, k) = \sum_{m=-\frac{N}{2}}^{\frac{N}{2}-1} M(hn + m)w(m)e^{-\frac{2j\pi mk}{N}}, \quad (2)$$

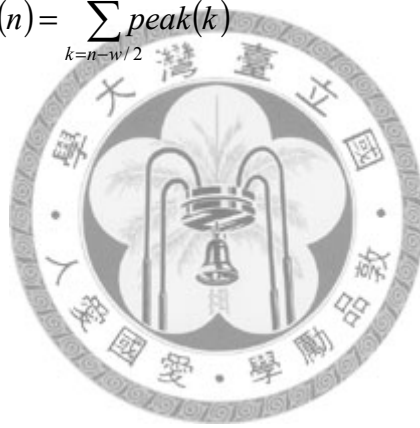
where N is the windows size, and h is the hop size. If the sampling rate of the background music M is 44100Hz, N and h are set as 2048 and 441 in our system. Spectral flux [26] is one of the onset functions that can measure the changes of magnitudes between frequency bins:

$$Flux(n) = \sum_{k=-\frac{N}{2}}^{\frac{N}{2}-1} H(|F(n,k)| - |F(n-1,k)|), \quad (3)$$

where $H(x) = (x+|x|)/2$ is the half-wave rectifier function. Then a peak at the n th frame is selected as an onset if it fulfills the peak-peaking algorithm in [27].

Tempo Estimation: Let $peak(n)$ represent an onset detection function. If the n th frame conveys a peak, the output of $peak(n)$ is one. Otherwise, the output is zero. Finally, we formulate the tempo of the n th frame of the background music M as the sum of $tempo(n)$ over a local window with size w :

$$tempo(n) = \sum_{k=n-w/2}^{n+w/2} peak(k) \quad (4)$$



Chapter 6

Interest Meter

We define the Interest Meter by two models: the attention model and the emotion model, where attention describes the visual focus of the user and emotion describes the inner state of the user. We establish the attention model with head motion detection, blinking detection and saccade detection. In emotion model, facial expression recognition is the main part. We use fuzzy logic to calculate the attention score and determine the emotion score based on facial expression recognition results. With the information, we can further calculate interest score. Figure 7 demonstrates the framework of the Interest Meter.

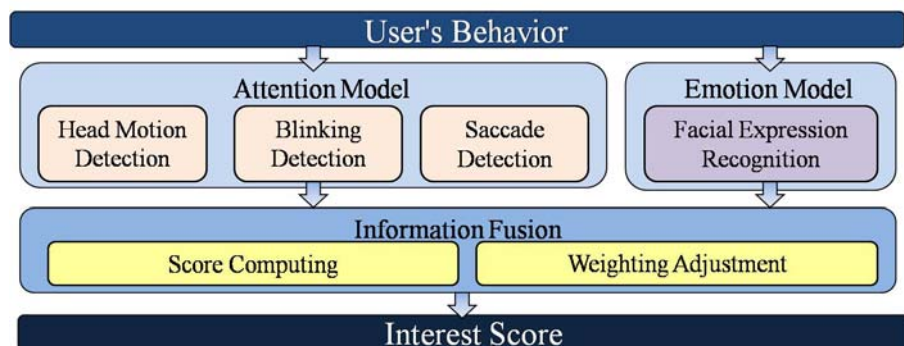


Figure 7. The framework of Interest Meter.

6.1 Attention Model

When watching videos, users spontaneously express their feelings by facial expressions. For example, when something funny happens in videos, most users smile or laugh at what they see. To gain information from such user interests, we adopt facial expression analysis in our work.

6.1.1 Head Motion Detection and Score Calculation

To calculate the head motion score, we calculate the displacement of face positions between two consecutive frames. The score for head motion can be expressed as

$$S_m(t) = e^{-\frac{(m(t))^2}{\sigma^1}}, \quad (5)$$

where $m(t)$ is the displacement of the face position at time t from the previous face position at time $t-1$, and σ^1 is a control factor.

6.1.2 Blinking and Saccade Detection

For blinking and saccade detection, we consider three visual features: the center of the eyeball, the two corners of the eye and the upper eyelid.

To find the center of the eyeball, the opening operator is first applied to eliminate the highlight that may be caused by the reflection on the cornea (Figure 8(b)), and then the iris is estimated by convolving the gray eye image with a Gaussian-shaped filter to find the center of the darker region (Figure 8(c)). Vezhnevets et al. [28] propose a similar function for the same purpose. We define the function as

$$G(x, y) = Ae^{-\frac{(x-x_0)^2+(y-y_0)^2}{2(\sigma^2)^2}}, \quad (6)$$

where the coefficient A is the amplitude, (x_0, y_0) is the center, and σ^2 controls the width of the Gaussian shape. We rescale the eye image to a fixed size before convolution. The parameter σ^2 can be chosen according to the expected iris size. After convolution, the pixel with the lowest response is considered to be the approximate iris center (Figure 8(d)).

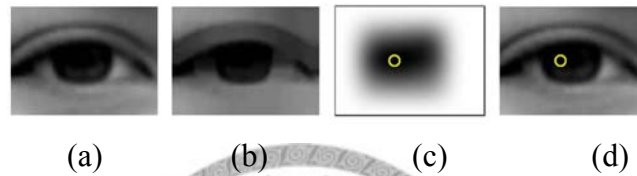


Figure 8. Detection of center eyeball. (a) Original eye image; (b) Opening operator is applied on (a); (c) Gaussian filter is applied on (b); (d) approximate iris center.

The above method yields an approximate estimate of the iris center, but the result may not be the precise location. In order to refine the iris center determination and estimate the iris radius, we further identify the circular shape of the iris. First, the edge map of the eye image after eliminating the highlight is obtained using the Canny edge detection method. To find the sample points of the iris boundary on the edge map, the algorithm begins at the approximate iris center as a starting point, and then the intersections are obtained along rays extending radially away from the starting point. The diameter of the iris is always smaller than the length between the two corners [29], so the length of each ray is limited to half the length between the two corners.

An example set of sample points is shown in Figure 9(a). We restrict the directions of the rays because the iris is likely to be occluded by the eyelids and

eyelashes. The range of angles is adjustable to accommodate different users, but it is initially defined to include the ranges -45° to 45° and 135° to 225° . One ray is traced per 5 degrees, resulting in at most 108 candidate sample points. In a real situation, however, there are large outliers due to eye blinks. In order to eliminate these outliers, an upper eyelid point is obtained by tracing a vertical ray from the starting point and finding an intersection, and then those sample points above the two links between the upper eyelid point and the two eye corners are excluded (Figure 9(b)).

The candidate sample points may still contain outliers. A circle is fit to the candidate sample points using the Random Sample Consensus (RANSAC) paradigm [30]. Unlike a least-squares fitting approach, this paradigm reduces the influence on the accuracy of these outliers. We introduce two restrictions on the RANSAC fitting process to increase the robustness of the inliers selection process. First, only candidate circles that include the starting point within the covered areas are considered. Second, based on the structure of the eye, the ratio of the iris diameter to the length between the two eye corners is about 1:3, so only candidate circles with reasonable ratios (about 1:3) are considered. The inliers and outliers are shown as green and red crosses, respectively, in Figure 9(c), and the final circle fit is shown in Figure 9(d).

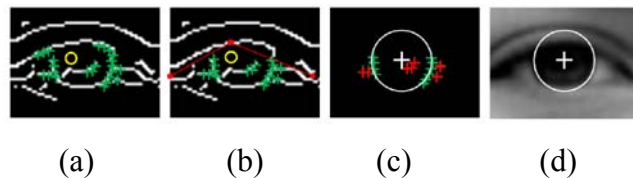


Figure 9. Iris refinement.

To detect the corners of the eye we modify the method proposed in [22], which utilizes Gabor wavelets to localize possible corners. The color distribution of the

sclera region can be distinguished from the flesh tone in the face. The largest wedge looks for the flesh tone, and the smallest wedge looks for the sclera tone. A right or left corner is detected if the average value of the pixels in each wedge satisfies the comparative color tone criteria.

Based on the positions of the eyeball and the two corners of the eye, we estimate the eye movement by comparing the relative distances between the eyeball center and the eye corners over time. If the velocity of the eyeball movement between the current frame and the previous frame is larger than a threshold, this indicates a saccade.

An eye blink is detected when the iris center is occluded by the upper eyelid, and a blink is defined as the user opening his/her eyes after closing them. Whether or not the iris center is occluded determines the status of the eye at each frame. Let $Blink(t)$ represent the status of the eye at time t .

$$Blink(t) = \begin{cases} Open & \text{if } H_i \geq H_e \\ Closed & \text{otherwise,} \end{cases} \quad (7)$$

where H_i and H_e are the distances from the upper boundary of the eye region to the iris center and the upper eyelid point, respectively. Figure 10 shows the two eyes states. As the eye changes from the open to closed states, we determine that a blink occurs.

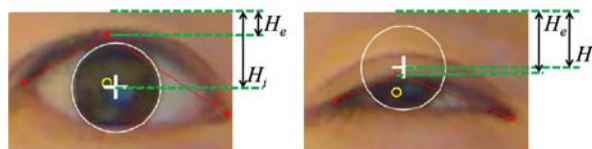


Figure 10. Blink detection.

6.1.3 Blinking Score Calculation

To calculate the blinking score, we first define a blinking detection function $b(t)$. If a blink is detected at time t , then $b(t)=1$; otherwise $b(t)=0$. The blinking score can be expressed as

$$S_b(t) = \begin{cases} 1, & \text{if } \sum_W b(t) \leq 1 \\ 0, & \text{else} \end{cases}, \quad (8)$$

where W is a one-second sliding window. More than one blinking event in this one second window indicates abnormal blinking.

6.1.4 Saccade Score Calculation

Goldstein et al. [31] classified eye movements into three categories: fixations, smooth pursuits and saccades. They reported that a movement velocity larger than 200 degrees/second corresponds to a saccade. In this work, we take saccades into account because they indicate shifts of viewer attention. The more saccades occur while viewing a shot, the less interesting this shot is for viewers.

We also analyze saccade based on a one-second sliding window W . If a saccade is detected at time t , then the saccade detection function $s(t)=1$; otherwise $s(t)=0$. The score of saccade can be expressed as:

$$S_s(t) = \begin{cases} 1, & \text{if } \sum_W s(t) = 0 \\ 0, & \text{else} \end{cases}, \quad (9)$$

6.1.5 Attention Score Calculation

The fuzzy system, proposed by Takagi and Sugeno [[23]], is a paradigm for an alternative design methodology which can be applied in developing both linear and non-linear systems for embedded control. The advantage of fuzzy logic is that we

can describe systems using simple English-like rules. It does not require system modeling or complex math equations governing the relationship between inputs and outputs.

Based on this theory, we use fuzzy logic to calculate the attention score $S_a(t)$ at time t , and the *fuzzy if-then rule* can be expressed as:

$$\left. \begin{array}{l} \text{IF } (S_m(t) \text{ is high}) \text{ AND } (S_b(t) \text{ is } 1) \text{ THEN } (S_a(t) \text{ is } FS_1(t)) \\ \text{IF } (S_m(t) \text{ is high}) \text{ AND } (S_s(t) \text{ is } 1) \text{ THEN } (S_a(t) \text{ is } FS_1(t)) \\ \text{IF } (S_b(t) \text{ is } 1) \text{ AND } (S_s(t) \text{ is } 1) \text{ THEN } (S_a(t) \text{ is } FS_1(t)) \\ \text{otherwise } S_a(t) \text{ is } FS_2(t) \end{array} \right\} \quad (10)$$

where S_m is the head motion score, S_b is the blinking score and S_s is the saccade score. The notation $S_a(t)=FS_1(t)$ means that the user is attentive to the object. In general, attention accumulates over time but can be immediately lost. Based on this observation, the value of attention in the present frame should change depending on the value of the previous adjacent frame. Therefore, we can define $FS_1(t)$ and $FS_2(t)$ as follows:

$$S_a(t) = \left. \begin{array}{l} FS_1(t) = S_a(t-1) + \gamma, \quad \text{attention} \\ FS_2(t) = \alpha \times S_a(t-1), \quad \text{inattention} \\ S_a(0) = 0 \end{array} \right\} \quad (11)$$

In attention situation, the attention score increases stably with a slope of γ . On the other hand, the attention score would decrease by α ($\alpha < 1$) times the original attention score when the user is inattentive. Figure 11 shows an example of the attention score.

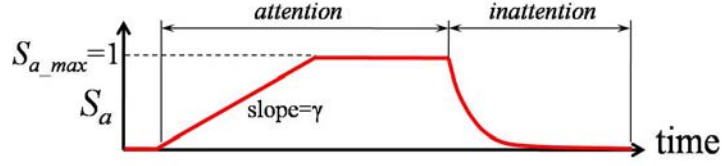


Figure 11. The illustration of attention score.

6.2 Emotion Model

When watching videos, users spontaneously express their feelings through facial expressions. For example, when something funny happens in a video, most users smile or laugh at what they see. Thus, we adopt facial expression analysis to gain information from such user expressions. For facial expression recognition, instead of analyzing six-class expressions [24][25], we classify human expressions into two categories: positive expressions and neutral expressions. A positive expression is defined as a positive human reaction that implies the user is interested in this object, including smiling and laughing. Expressions other than positive expressions are classified as neutral expressions.

6.2.1 Facial Expression Recognition

We adopt a manifold learning and fusion classifier to integrate multi-component information for facial expression recognition. Our work employs a total of nine facial components to determine expression. Given a face image I , a representative feature is constructed by learning the mapping $M : R^d \times c \rightarrow R^t$ based on facial components. Essentially, the mapping M encodes the probability of each expression in facial components, and can be defined as

$$M(I) = [m_1(I_1), m_2(I_2), \dots, m_c(I_c)] \quad (12)$$

where c is the number of components, $m_i(\cdot)$ is an embedding function of the component i , and I_i is a d -dimensional sub-image of the i^{th} component. By learning the geometry of the training data, an embedding function $m_i(I_i)$ can be obtained by projecting I_i onto the learned manifold. In our framework, a probabilistic representation of $m_i(I_i)$ can be written as

$$m_i(I_i) = \frac{1}{D^p + D^n} \{D^p, D^n\}, \quad (13)$$

where D^p is the shortest distance between I_i and the positive training data, and D^n is the shortest distance between I_i and the neutral training data. Based on these formulations, the multi-component information is then encoded in a t -dimensional feature vector $M(I)$, where t is $2 \times 9 = 18$ in this case.

To characterize the significance of components from the embedded features, a fusion classifier $F : R^t \rightarrow \{\text{Positive, Neutral}\}$ is constructed based on a probabilistic SVM classifier. This method allows our system to recognize users' emotions. Figure 12 shows the process of facial expression recognition.

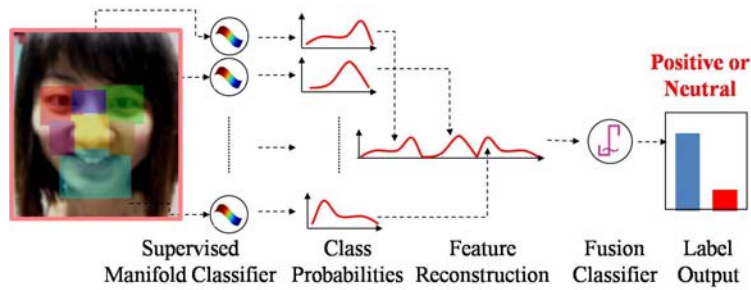


Figure 12. The process of facial expression recognition.

6.2.2 Emotion Score Calculation

This method determines the emotion score based on the facial expression recognition results. We use the probability of a positive emotion from the facial

expression recognition as the emotion score $S_e(t)$, which ranges from 0 to 1. A positive emotion score represents a shot that is more important than neutral one for the viewer at time t . Figure 13 illustrates the emotion score over time.

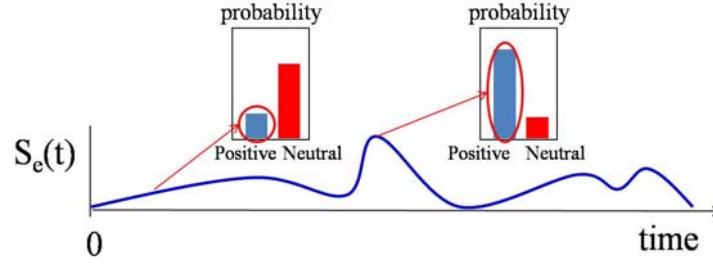


Figure 13. The calculation of emotion score.

6.2.3 Interest Score Computing and Weighting Adjustment

The interest score can be described as follow

$$S_i = W_a \times S_a + W_e \times S_e, \quad (14)$$

where W_a and S_a are the attention weight and attention score, W_e and S_e are the emotion weight and emotion score, and S_i is the interest score.

When the positive probability is higher than the neutral probability in the facial expression recognition result, we prefer the emotion score to represent the interest score. In this case, we increase the weight of the emotion score and decrease that of the attention score. When the attention score increases, the user starts to concentrate. In this case, we mostly use the attention score to represent interest. The formula can be described as:

$$\begin{bmatrix} W_a \\ W_e \end{bmatrix} = (1 - \beta) \times \begin{bmatrix} W_{a_pre} \\ W_{e_pre} \end{bmatrix} + \beta \times \begin{bmatrix} a \\ b \end{bmatrix}, \text{ where } (a, b) = (1, 0) \text{ or } (0, 1)$$

$$\beta = W_b \times (1 - S_b) + W_s \times (1 - S_s) + W_m \times (1 - S_m), \quad (15)$$

where W_a and W_e are the attention and emotion weights, W_{a_pre} and W_{e_pre} are the weights from the previous frame. W_b is the blinking weight, W_s is the saccade weight, W_m is the head motion weight, and $W_b + W_s + W_m = 1$; S_b is the blinking score, S_s is the saccade score and S_m is the head motion score. When the positive probability is higher than the neutral probability in the facial expression recognition result, we set $(a,b)=(0,1)$; otherwise $(a,b)=(1,0)$. We use β to control the variance of the adjustment amount. When there are more inattentive reactions, the value of β and the adjustment amount increase.



Chapter 7

Summary Generation

With the shots in the filtered video V_{good} and the tempo information of the background music M , we are now ready to turn to our aesthetics-based editing method, which consists of three steps: rhythm establishment, shot trimming, and transition determination.

7.1 Rhythm Establishment

Since lengths of the input video and background music are not necessary the same, durations of video shots must be adjusted to match the length of the background music, and the visual rhythm caused by shot changes is desired to be synchronous with the music tempo. As we mentioned in Chapter 3.1, the easiest way to achieve this is exploiting “cut tight” and “cut loose”. That is, a fast music tempo will result in fast shot changes in the summarized video, and vice versa. Figure 14 illustrates how we use a transfer function for this purpose, and details are described as follows.

We first linearly map shot durations to the length of the background music. We define t_i^{pre} as the begin time of $shot_i$ in V_{good} after this pre-mapping process:

$$t_i^{pre} = \frac{\sum_{k=1}^{i-1} length(shot_k)}{length(V_{good})} length(M), \quad (16)$$

where function $length(\cdot)$ represents the time length of a given shot in terms of audio frames. To synchronize visual rhythm with music tempo, we try to alter the duration of each shot. Motivated by histogram equalization techniques, we try to design a transfer function that is monotonically increasing and transforms the starting time of each shot according to the music tempo. The transfer function $TF(n)$ is defined as

$$TF(n) = \sum_{k=1}^n (tempo_{max} - tempo(k) + \delta), \quad (17)$$

where n represents the n^{th} audio frame of the music, $tempo_{max}$ denotes the maximum value of all $tempo(n)$, and δ is a factor that controls the strength of the video rhythm.

Then, we define t_i^{post} as the begin time of $shot_i$ that is further mapped according to this transfer function $TF(n)$ in this post-mapping process:

$$t_i^{post} = \frac{TF(t_i^{pre})}{TF(length(M))} length(M) \quad (18)$$

After post-mapping, the visual rhythm caused by shot changes is better synchronized with the music tempo of the background music M . In order to make shot changes occur exactly at music onsets in the output music video, we further adjust t_i^{post} to align with its nearest onset peak t_i^{onset} , as shown in Figure 14.

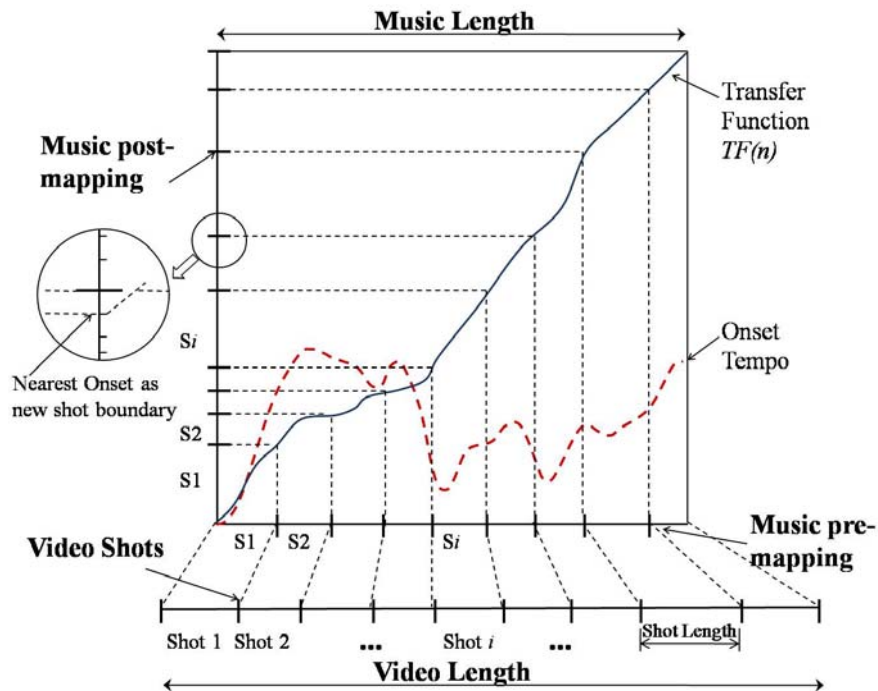


Figure 14. Illustration of rhythm establishment.



7.2 Shot Trimming

In the process of summarizing home videos, for each raw shot, the interest score S_i is calculated accordingly, and the optimal shot clip with the maximum interest score is selected to be the representative part of this raw shot. Through the processes described above, the selected shots are concatenated as the final video summary. The procedure is illustrated in Figure 15.

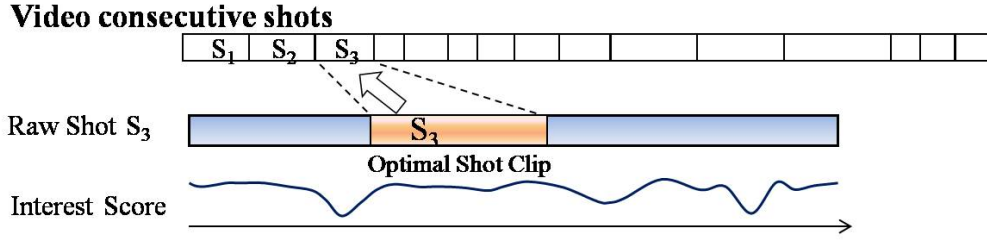


Figure 15. The illustration of how to choose an optimal shot clip based on interest score automatically.

7.3 Transition Determination

Now we are ready to consider transition effects, which were introduced in Section 3.1. Transition occurs in two different situations: at the beginning and the end of the entire video, and between adjacent shots. According to the editing theory in Section 3.1, fade-in and fade-out are applied to the beginning and the end of the entire video. For transition effects between adjacent shots, we consider the average tempo of music clip in each shot.

$$ShotTempo(i)^{music} = \frac{\sum_{n=t_i^{onset}}^{t_{i+1}^{onset}-1} tempo(n)}{t_{i+1}^{onset} - t_i^{onset}} \quad (19)$$

To classify the tempo of music in each shot, $tempo_{high}$ and $tempo_{low}$ are calculated. They are 30% highest and 30% lowest tempo of background music, respectively. Let $\kappa(i)$ represents the impact factor of music tempo in the $shot_i$.

$$\kappa(i) = \left\{ \begin{array}{ll} 0 & \text{if } ShotTempo(i)^{music} > tempo_{high} \\ 1 & \text{otherwise,} \\ 2 & \text{if } ShotTempo(i)^{music} < tempo_{low} \end{array} \right\}, \quad (20)$$

The transition duration between $shot_i$ and $shot_{i+1}$ is determined by:

$$TR(shot_i, shot_{i+1}) = \min \left(\frac{(length(shot_i^{music}) + length(shot_{i+1}^{music})) \times (\kappa(i) \times \kappa(i+1))}{\alpha}, \beta \right), \quad (21)$$

where α and β can be decided by the total shot numbers and music styles. More shot numbers in a video and heavier music, smaller magnitude of α and β are set. A music segment is claimed as heavier if the summation of $peak(n)$ in a 2-sec duration centered at the n th music frame is larger than 4. In this case, the value of β is set as 0.5, otherwise it is set as 1. The transition effect we used in our system is dissolve, which is the most common effect in film. If the music tempo in $shot_i$ is labeled as fast, the factor of $\kappa(i)$ is zero. In this case, the transition duration is also zero, and a cut transition is adopted - to generate a staccato video rhythm.



Chapter 8

User Interface

Figure 16 shows the user interface of our system, which consists of two main designs: (1) a video editing and (2) a rhythmic control.

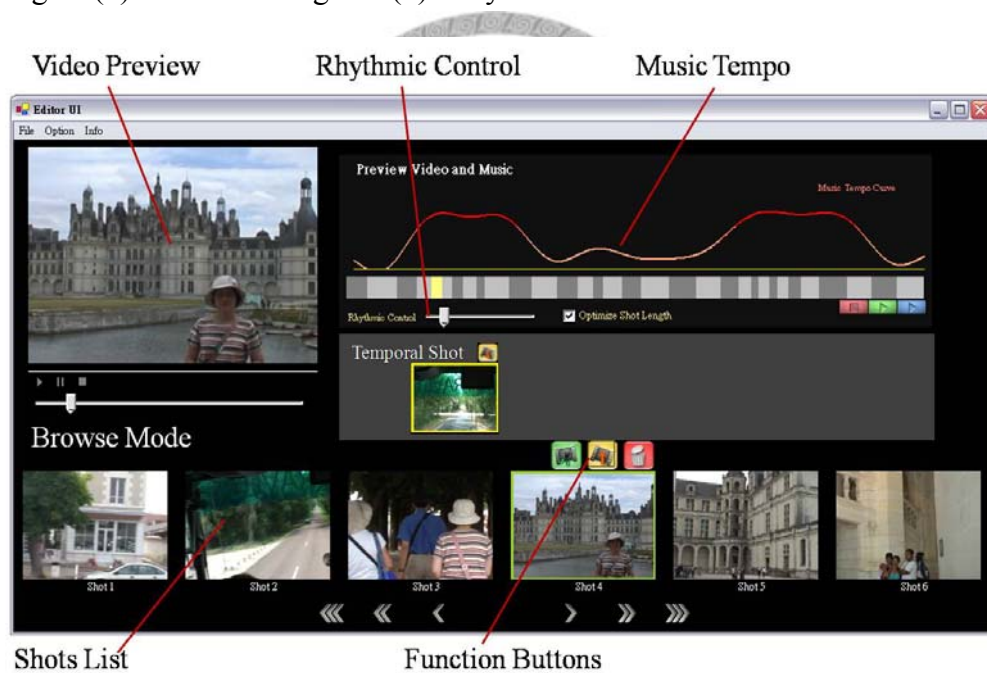


Figure 16. The User Interface of our system.







8.1 Video Editing

In general, commercial software user interfaces provide many editing tools and appear intimidating to users unfamiliar with the software. By contrast, our function

buttons are hidden when not in use, and they only reappear when users click on one of the shots in the shot list or on one of the temporal shots. We designed six function buttons, as shown in Table 2.

Clicking on “Edit Shot” switches the interface to an “edit mode” from the present “browse mode.” Users can use the mouse to choose the clips that want to piece together in a music video, as shown in Figure 19.

Table 2. Function buttons

 Edit Shot	 Back to Browse
 Put to Temporal	 Back to Shot List
 Delete	 Recover

Clicking on “Delete,” a deleted animation will be shown, and the unwanted shot is removed from the list. To retrieve the deleted shot, the user must simply click on the “Recover” icon shown on top-right of the shot, as shown in Figure 17. Clicking on “Put to Temporal Shot” removes the chosen shot from the shot list, and then places it in the temporal shot area. Clicking on “Back to Shot List” returns the clip to the shot list, as shown in Figure 18. This allows users to change the order of the shots and place shots in different positions with ease.

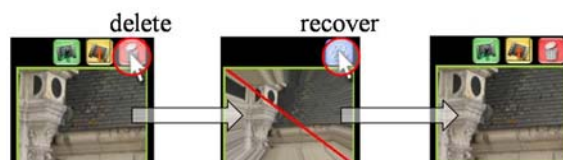


Figure 17. The illustration of delete shot and retrieve it.



Figure 18. The Illustration of putting the shot into temporal.

As mentioned above, one of the difficulties in music video editing is repeated cutting, which is necessary because it is difficult to edit the total length of shots to fit exactly with the length of the music. This software contains an effective way to replace “cutting” with “choosing.” We use a “transfer function” to constrain the total length of shots to fit the music, determine the length of each shot based on music tempo, and finally choose the proper shot clips with pre-determined lengths. If users adopt the Interest Meter to analyze their interest in each raw shot, the system will help them choose shot clips based on their interest scores, which is time-saving and human-centric.

As shown in Figure 19, the edit mode allows users to choose a piece of shot clip by simply dragging and dropping the chosen bit. They can further extend or shorten the length of the selected clip by scrolling the mouse wheel. The system automatically puts the shot clip back into the music video and adjusts the length of other clips if needed. This “choosing” method greatly decreases the amount of time required to edit a video.

8.1 Rhythmic Control

As we described in Chapter 7, we establish video rhythm by analysis the music

tempo. We use δ as a factor that controls the strength of the video rhythm. An example of rhythmic control shows in Figure 20. The first row shows successive shots that have approximately the same length, and the second row shows shots with a higher value of δ that produce stronger video rhythm. User can modify the video rhythm simply by dragging the slider in our user interface.

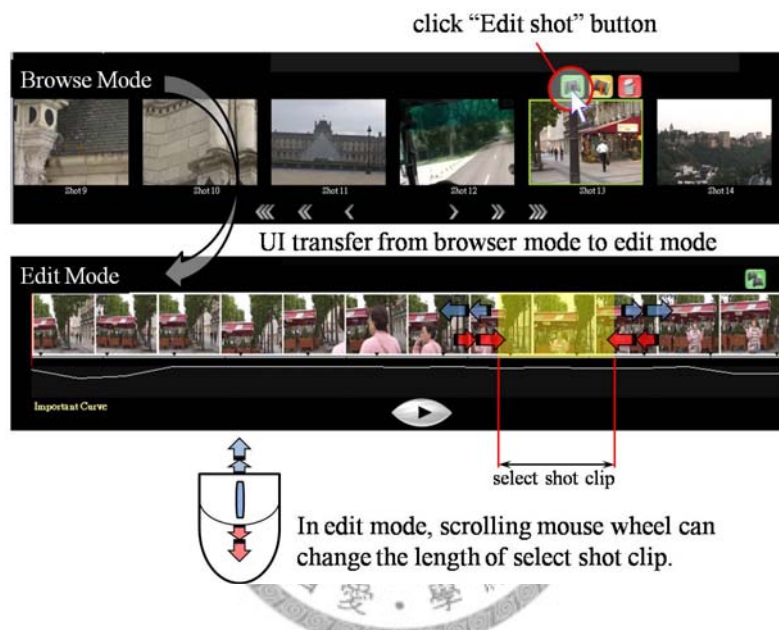


Figure 19. Using mouse control to select the clip in shot and change its length

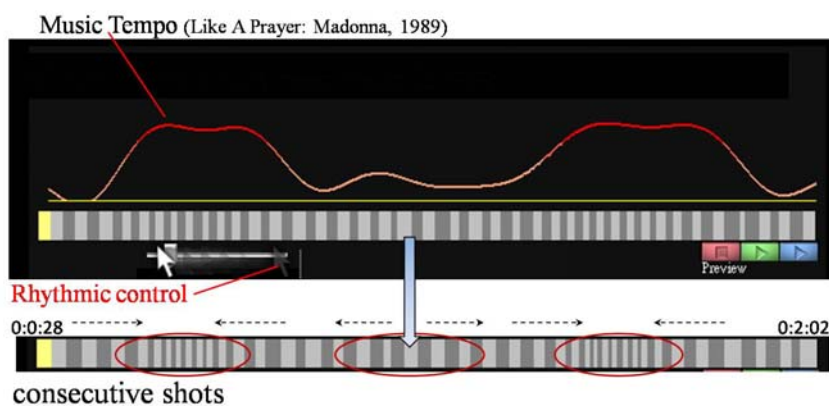


Figure 20. An example of rhythmic control.

Chapter 9

Experimental Results

The following sections describe the method used to evaluate the software and present experimental results. First, we compare different systems in terms of quality estimation. Then, we validate the performance of the eye detection and facial expression recognition methods. To measure the error of iris center location, we test the proposed method on the BioID [32] facial image database and compare the measured results with other methods in the literature that use the same database and the same accuracy measure.

We also designed a user study and invited some experts, amateurs and novices to use our system. We recorded the time they spent editing a music video using commercial software and our system. After completing the editing process, they were asked five questions about the user experience.

Finally, we designed two experiments to verify whether our system is useful. In the first experiment, two videos were used to compare our software with commercial software in terms of rhythm estimation. In the second experiment, we applied the proposed editing method to five videos and compared the results with (1) the results of randomly selected shots by automatic-editing software; and (2) the results of manual editing by a novice. The conformity between the music and video and the

clarity of the results of the three different methods were measured. This experiment was designed to verify whether the Interest Meter could effectively highlight the parts of a video that are appealing to users.

9.1 Quality Estimation

To compare these systems in terms of quality estimation, we use two videos that contain 68 shots totally, in which 53 shots are captured in night scenes. The night scene shots are likely to be underexposed (eight shots) and blurred (five shots) in all clips. The keyframe of each shot is shown in Figure 21.

Tables 3 and 4 show the results of the Experiment. Both MuVee [19] and PowerDirector [17] can detect and remove most underexposed and blurred shots. However, MuVee mis-detects fifteen ordinary night scene shots as underexposed ones. MuVee has about 28.3% false alarm rate, while our method achieves 5.7%. PowerDirector can better preserve ordinary night shots, at the cost of selecting six underexposed shots. The hit rate of PowerDirector is about 25%, while ours is 100%. Overall, our system outperforms these two commercial softwares in bad shot removal and night scenes preservation.





Figure 21. Keyframes of every shot in two videos.

Table 3. Detection results of underexposed and blur shots.

	Underexposure								Hit (%)	Blur					Hit (%)
	1-10	1-12	1-13	2-6	2-9	2-16	2-21	2-31		2-25	1-11	1-16	2-12	2-24	
S1	D	D	D	D	D	D	D	D	100	D	F	D	D	D	80
S2	D	D	F	F	F	F	F	F	25	D	D	F	F	F	40
S3	D	D	D	D	D	D	D	D	100	D	D	D	D	D	100

S1: Our system, S2: PowerDirector, S3: MuVee, F: fail, D: totally drop

Table 4. False alarm for ordinary night scenes

	Shots that caused false alarm	False Alarm (%)
S1	2-13, 2-22, 2-36	5.7 (=3/53)
S2	1-26, 1-27, 1-32, 2-22	7.5 (=4/53)
S3	1-3, 1-4, 1-7, 1-31, 1-32, 2-13, 2-17, 2-22, 2-23, 2-26, 2-27, 2-28, 2-29, 2-30, 2-34	28.3 (=15/53)

9.2 Evaluation of Interest Meter

9.2.1 Accuracy of Iris Center Location

The BioID database consists of 1521 grayscale images of 23 different subjects with a resolution of 384×286 pixels. These facial images were taken during several sessions at different places, i.e., this dataset features uncontrolled illumination and

background variations. In addition, the subject positions change both in scale and pose. In some instances the eyes are closed, or turned away from the camera. In many samples, the subjects wear glasses where the eyes are hidden by the spectacle frames or strong highlights on the glasses.

Due to these conditions, the BioID database is usually considered a challenging dataset. The ground truth of the left and right iris centers is provided with the dataset. To validate the accuracy of our iris center location method, the normalized error, indicating the error obtained by the worst eye estimation, is used to measure the error rate between the estimated iris center locations and the ground truth. This measure was proposed by Jesorsky et al. [33] and is defined as

$$e = \frac{\max(d_{left}, d_{right})}{\omega}, \quad (22)$$

where d_{left} and d_{right} are the Euclidean distance between the determined eye positions and the ground truth, and ω is the Euclidean distance between the eyes in the ground truth. Since the distance between the two inner eye corners is roughly equal to the eye length, $e \leq 0.25$ (a quarter of the interocular distance) roughly corresponds to the distance between the iris center and the eye corners, $e \leq 0.1$ corresponds to the range of the iris, and $e \leq 0.05$ corresponds to the range of the pupil. Besides the maximum (worst eye) normalized error, we also measure the minimum (best eye) normalized error to give upper and lower bounds to the accuracy and to present an average between the best and worst case estimates.

Table 2 compares the results with those of other methods, indicating a normalized error smaller than $e \leq 0.05$, 0.1 , and 0.25 respectively. It can be seen that with a normalized error smaller than $e \leq 0.25$, our method achieved superior accuracy

with respect to the other methods.

Similarly, for the iris location ($e \leq 0.1$), our method also exceeds these methods in accuracy. In the case of eye center location ($e \leq 0.05$), which is more accurate, our method still exceeds the others in accuracy except for the method proposed by Valenti and Gevers [34]. This can be explained by the fact that they train a classifier to find the best possible choice out of all candidate iris centers. The result is easily influenced by the training set, which is not clearly described in the paper. If the training sets and the test set are obtained from the same data set, of course the accuracy would be expected to be superior. When the basic approach is applied without classification, their method only has a 77.15% accuracy rate for $e \leq 0.05$, which is lower than that of our method.

Table 5. Performance comparison in terms of accuracy of eye detection.

Method	Accuracy ($e \leq 0.05$)	Accuracy ($e \leq 0.10$)	Accuracy ($e \leq 0.25$)
Our	80.36%	97.05%	99.65%
Valenti [34]	84.10%	90.85%	98.49%
Türkan [35]	19.00%	73.68%	99.46%
Asteriadis [36]	74.00%	81.70%	97.40%
Bai [37]	37.00%	64.00%	96.00%
Campadelli [38]	62.00%	85.20%	96.10%
Cristinacce [39]	56.00%	96.00%	98.00%

9.2.2 Accuracy of Facial Expression Recognition

Although facial expression recognition is still a difficult problem, this method achieves a recognition accuracy of 94.7% in the case of static pictures, with a recognition accuracy of about 80% for videos. Figure 22 shows a portion of the

recognition results and the ground truth from one subject.

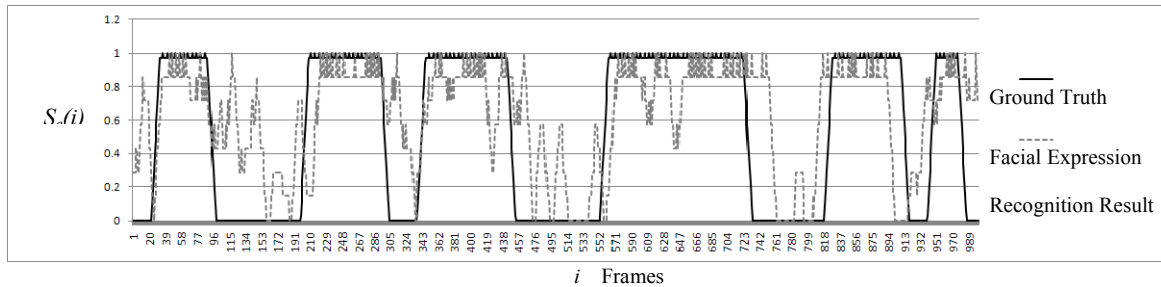


Figure 22. The ground truth and recognition result of facial expression.

9.2.3 Verification of Interest Meter

We invited 6 subjects (4 males and 2 females) who are volunteers in the experiment. Participants are from 20 to 35 years old. All participants were unaware of the specific purpose of the experiment. We prepared two testing videos, which are all about 2 minutes. Video 1 is composed of normal and funny segments, while video 2 is composed of normal and attentive segments. The difference between two testing videos is that the funny segments of video 1 may trigger participants' emotion reactions.

When participants watched two videos, Interest Meter analyzed their viewing behaviors and calculated attention, emotion and interest scores for each frame. This experiment is designed to verify whether the Interest Meter measures user's interest well. Figures 23 and 24 show the average results of participants. In Figure 23, we can find that the emotion scores are lower in normal segments and are higher in funny segments. Every participant has his own subjective cognition, though they watch the same videos. Responses of participants were not necessary the same in each frame, and we can still examine the difference between funny and normal

segments.

In Figure 24, the differences between attentive and normal segments are not obviously than that in video 1, especially the emotion score, but we also can find lower attention scores in normal segments and higher scores in attentive segments. Test results show that the Interest Meter can appropriately measure user’s interest based on viewers’ viewing behaviors.

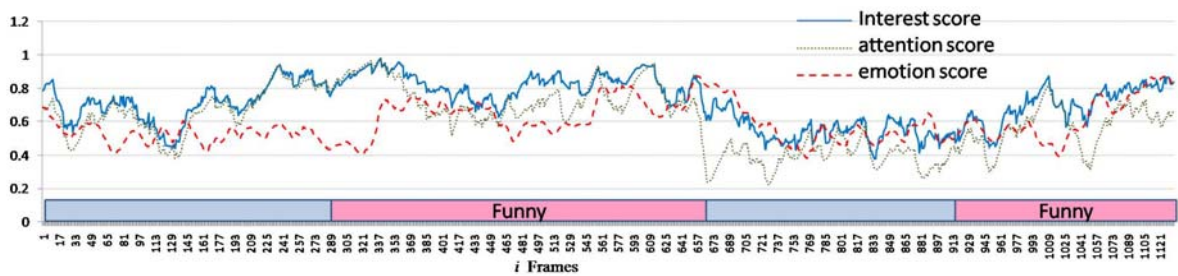


Figure 23. The average scores of participants when watching video 1.

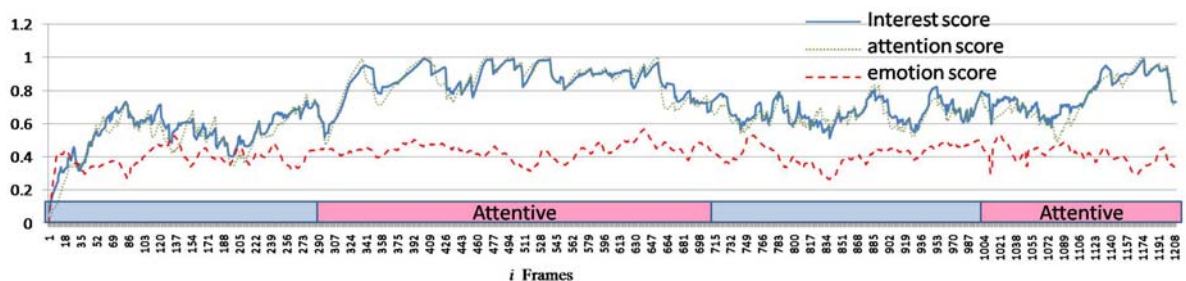


Figure 24. The average interest scores of participants when watching video 2.

9.3 User Study on Interface

To estimate whether our interface is user-friendly, we prepared a video and two songs for the user study. Participants were asked to perform editing tasks separately with our system and the commercial software that they were accustomed to. We recorded the time they spent using the two different systems. After editing the videos,

the participants were required to answer five questions, producing scores from 1 to 5. Higher scores mean that the users were more satisfied, and our conclusions are drawn based on the average total scores.

The videos in this experiment were processed manually without using the Interest Meter.

Question 1: In the user interface, do you think the design and display of our editing tools are convenient to you?

Question 2: In shot editing, do you think the concept of “choosing” video clips is more useful than “cutting”?

Question 3: In rhythmic control, do you think the control of the rhythm is intuitive and that the visual rhythm matches the music tempo?

Question 4: Do you think that this software can help you save a lot of time in editing home videos?

Question 5: Overall, do you find this software helpful to end-users in editing home videos?

Table 6. The background of participants.

No	Gender	Age	Class	Mode
1	male	41	Expert	1
2	male	29	Expert	2
3	male	26	Amateur	1
4	male	45	Amateur	2
5	female	24	Novice	1
6	male	22	Novice	2

This experiment included 6 participants (5 males and 1 female) between 22 and 45 years old. In total, 2 experts, 2 amateurs, and 2 novices were involved in this user study, and the experiment was conducted using two editing modes. In Mode 1, participants were asked to use the commercial software that they were accustomed to and then our system to edit the music video. In Mode 2, they were assigned to use the software in the reverse order. Table 6 shows the background information of these participants. Before using our system to edit, each participant was given a 5 minute demonstration on how to use our system. The theme of the video was “a trip to the zoo.” The raw video is 4 minutes 56 seconds long and is composed of 17 shots. We prepared two songs to go with the video; one is 1 minute long, and the other is 1.5 minutes long.

Figure 25 shows the user responses to the five questions. Overall, participants were satisfied when using our user interface. The results of Question 3 indicate that participants were able to sense the synchrony between the video rhythm and the music tempo, which indicates that our algorithms have performed as expected. The responses to the last two questions indicate that participants approved of our editing system and agreed that it would assist in video management.

As shown in Figure 26, for novices, regardless of whether they edited in Mode 1 or 2, it takes a long time to edit a music video with commercial software. Even when the raw video is less than 5 minutes long and the song is only one minute, it still takes more than 40 minutes to edit. User 5 had difficulty learning to use our system at first, and therefore she spent a slightly longer time on her first editing task, but when she re-edited the video, she was able to use the system with ease.

Experts and amateur editors who were quite familiar with the commercial

software spent much less time editing the music videos than novices. When using commercial software, the average editing time was 19 minutes with 17 minutes for re-editing. When using our system, these times were reduced to 10 and 8 minutes, respectively, almost doubling editing speed.

We further performed an ANOVA (ANalysis Of VAriance) for editing time. The model was a 2 System (commercial software and our system) \times 2 Editing Procedure (first editing and reediting) mixed model ANOVA. Table 7 shows the results of the ANOVA for within-subjects and between-subjects effects. F -ratio = (found variation of the group averages) / (expected variation of the group averages) and P is the significance level, whereas "large" F and "small" $P(<0.05)$ indicates statistically significant.

As shown in Table 7, we found that there are significant difference between our system and commercial software. After respectively using our system and commercial software twice (first editing and reediting), we obtain $F=8.625$ and $P=0.032<0.05$ in the first editing, and $F=47.316$ and $P=0.001<0.05$ in the reediting. As expected, this results show that our system can effectively reduce the effort a user spends on editing a music video, especially in reediting procedure.

For commercial software, there were not significant difference between two editing procedures ($F_{within}=8.477$, $P_{within}=0.062>0.05$). It means the time for two editing procedures doesn't make significant difference in commercial software. Moreover, there were not significant difference between participants from different class ($F_{between}=6.458$, $P_{between}=0.082>0.05$).

We also compare the same situation in two editing procedures when using our system. The results show that there were not significant difference between two

editing procedures ($F_{within}=0.668, P_{within}=0.474>0.05$). Comparing with commercial software, the difference is smaller in our system. When using our system, participants spend less time in the reediting procedure than that in the first editing. The main reason is that our system can reduce a lot of editing work to cut a video to the right length and automatically matches the tempo of the second song in the editing procedure.

There were not significant difference between different class of participants ($F_{between}=2.939, P_{between}=0.196>0.05$). Comparing with commercial software, the difference is also smaller in our system. It reflects that the gap between novice and amateur is smaller than that of commercial software. The above results show that our system can effectively reduce the efforts of editing a music video.

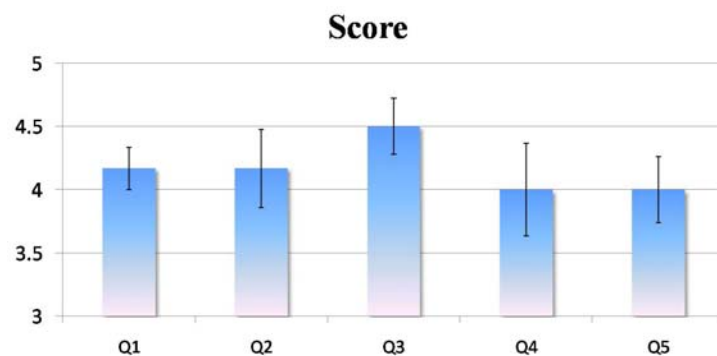


Figure 25. The experimental results of average scores of user experience (+/- standard error of the mean).

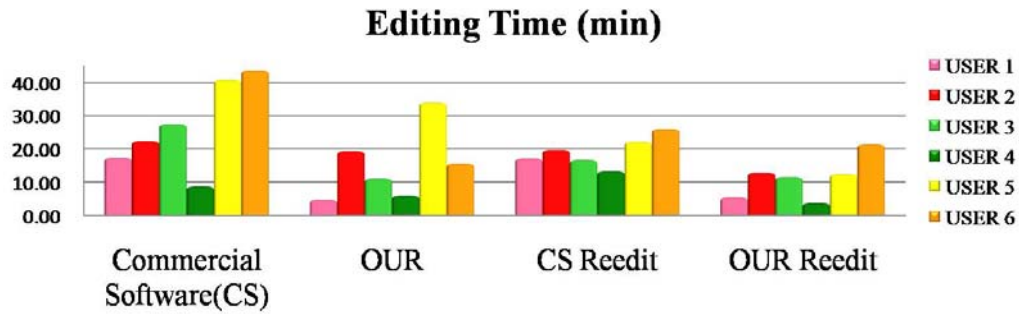


Figure 26. Editing time of participants.

Table 7. Within-Subjects and Between-Subjects effects.

Measure Source	<i>F-ratio</i>		<i>P</i>
Commercial Software & OUR system in First editing	Within-subjects	8.625	0.032
Commercial Software & OUR system in Reediting	Within-subjects	47.316	0.001
First & Reediting using Commercial Software	Within-subjects	8.477	0.062
	Between-subjects	6.458	0.082
First & Reediting using OUR system	Within-subjects	0.668	0.474
	Between-subjects	2.939	0.196

9.4 Experiments 1 on Summarization

The aim of this experiment is to verify how well our system establishes video rhythm, compared with other automatic commercial video editing software. We invited participants to watch summarized music videos generated by three different systems: PowerDirector [17], MuVee [19], and our system. In order to obtain accurate results, the participants were not informed in advance which video was generated by which system. After watching the videos, they were asked to give a

rhythm score from 1 to 10 to each music video. Higher rhythm scores meant that the video rhythms were well matched to the music tempo in the summarized videos. The videos in this experiment were edited automatically without using the Interest Meter.

There were 17 participants (11 males and 6 females) aged between 18 and 34 years old. The experiment lasted about an hour for each participant. In order to evaluate how well the video shots matched the music tempo, two entirely different music types were used for cross verification: Pop and New Age music. MuVee offers a variety of video styles for post-production, and we selected the “classic” type to process the video content. The details of Experiment 1 are shown in Table 8.

Table 8. The Evaluation data of Experiment 1.

	Video	Music
	Content	Duration
1	Travel	3m 40s
2	Night Scenery	3m 08s

Figure 27 shows the results of Experiment 1. Our system established a better video rhythm than the two other sets of software. Between the two commercial software programs, MuVee performed better than PowerDirector. The connection between the video and music could be perceived in the MV created by MuVee, as the shots changed with the beat. However, in addition to this feature, our system can also analyze music and use the “transfer function” to change the shot lengths, making the MV more intensely attractive than ever. This is not achievable by other commercial software. The results also indicate that our system performs better when the chosen background music has a strong tempo.

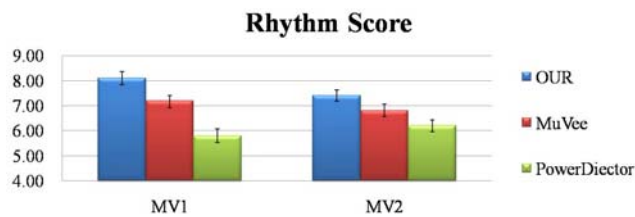


Figure 27. The rhythm scores of different editing system (+/- standard error of the mean).

9.5 Experiments 2 on Summarization

The aim of this experiment was to test whether the Interest Meter could help our system select shot clips that interest users. All participants were invited to view our test raw videos and let the system record their eye information and facial expressions. The information collected during the viewing process was used to produce personal summarized videos by the methods discussed above. Then, the participants were required to watch the summarized videos which were generated by three different methods and assign a satisfaction scores for these videos.

The videos in this experiment were edited automatically using the Interest Meter.

This experiment included 8 participants (6 males and 2 females) between 20 and 28 years old to participate in this experiment. All participants were invited to view these videos without understanding the specific purpose of the experiment. The experiment lasted about one and a half hours for every participant. This experiment differs from the first experiment in that the participants in Experiment 2 included the video providers or the subjects of the videos. In other words, the setting of this experiment is more similar to the real-life situation of a home video. We evaluated the proposed method based on five video sequences, each of which lasted about 7 to

18 minutes. Table 9 lists the specifications of the test videos and music.

Table 9. The Evaluation data of Experiment 2.

	Video		Music
	Content	Duration	Duration
1	Travel	13m 46s	3m 10s
2	Vacation	8m 06s	2m 10s
3	Motor Riding	18m 41s	3m 50s
4	Scenery	10m 58s	2m 10s
5	Wedding	7m 26s	1m 20s

9.5.1 Procedure

Since there is no objective measure available to evaluate the quality of summarized musical videos, we compare the automatically generated summaries with (1) summaries composed by randomly selecting shots; and (2) summaries manually edited by a novice user familiar with the basic concepts of video editing. All participants were required to watch these three videos (including their personal summarization results) and assign a satisfaction score from 1 to 10 to each edited video, with higher scores indicating greater satisfaction. The participants did not know which summary was generated by which method. Detailed evaluation results are shown in Figure 28. We use two attributes to evaluate the summarized videos, clarity and rhythm. Higher clarity values mean that more meaningful clips are reserved from each shot in a summarized video. Rhythm, on the other hand, represents the synchronization between the music tempo and the video rhythm. Participants were seated at a distance of about 40 cm from the 40 cm wide screen,

and the viewing angle subtended by the screen was approximately 52 degrees.

9.5.2 Results and Discussion

The satisfaction scores show that the videos produced by the system using the Interest Meter (OUR) are significantly higher than scores of videos processed using randomly selected clips or the edited results by the novice (NOVICE). The scores of the NOVICE are higher than those of the randomly selected summaries. The main reason for this difference is that random editing loses more important clips than OUR and NOVICE, and sometimes poor-quality frames are selected. Random editing does not consider the music tempo and cannot align the music onset with shot boundaries. The results also indicate that our editing system also has higher scores than the NOVICE edited results. These results indicate that the Interest Meter can help to select shot clips from each raw shot that interests the subjects.

The rhythm scores indicate that the generated summarized musical videos achieve good tempo matching between the video and music. This indicates that participants can sense the synchrony between video rhythm and music tempo, which shows that our algorithms have achieved our goals.

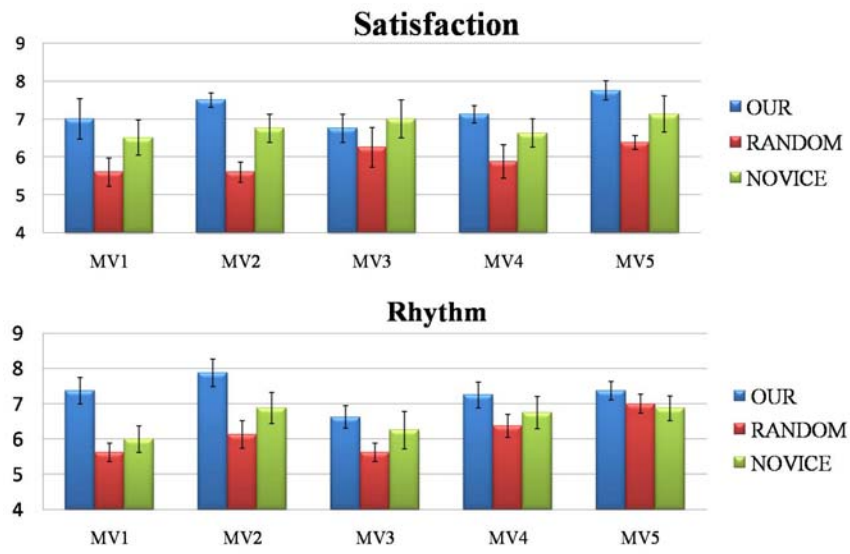


Figure 28. The satisfaction and rhythm scores of three methods (+/- standard error of the mean).



Chapter 10

Conclusions and Future Work

This work reports the development of a novel human-centric system for home music video generation based on rhythmic control and user interest. This software adopts editing rules for rhythm establishment and uses information on the viewer's blink rate, eye movements and facial expressions to generate home music video summaries. By analyzing the user's attention and emotional response, our system can automatically select important parts of raw video shots that users are interested in. The results in satisfactory performance match the user's interests. This work can be considered one of the first video summarization systems based on a human-centric concept. Our research proposes that future video summarization research would benefit from focusing more attention on user based sources of information.

We also design a user interface to facilitate interaction. Video display allows users to easily construct a video rhythm simply by dragging and dropping. If users wish to change the content of shot clips, they can also simply select the clips they like rather than cutting the video in a lengthy process. Our experimental results show that this new type of editing method can effectively generate home video summaries.

Additional work is necessary to make output videos more compelling and more

similar to professional MVs. For example, there are different types of MV expression styles. The important question of how to construct different styles in a semi-automatic MV system according to the background music and video content remains to be addressed. The ultimate goal of this study is not to design an all-encompassing music video editing system, but rather to propose some innovative functions that may inspire software programmers.



Bibliography

- [1] YouTube. <http://www.youtube.com/>
- [2] H. Zettl. *Sight, sound, motion: applied media aesthetics*, Wadsworth, 1998.
- [3] R.M. Goodman and P. McGrath. *Editing digital video : the complete creative and technical guide*, McGraw-Hill/TAB Electronics, (2002).
- [4] G. Chandler. *Cut by cut : editing your film or video*, Michael Wiese, (2006).
- [5] Adobe Premiere Pro. <http://www.adobe.com/products/premiere/>.
- [6] SONY Vegas Pro 9. <http://www.sonycreativesoftware.com/vegaspro>.
- [7] Apple iMovie'09. <http://www.apple.com/ilife/imovie/>
- [8] A. Money and H. Agius, Video summarisation: a conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation* 19, 2(2008), 121 - 143.
- [9] P. Mulhem, M.S. Kankanhalli, H. Hassan, and J. Yi. Pivot vector space approach for audio-video mixing, *IEEE Multimedia* 10, 2 (2003), 28 - 40.
- [10] J. Foote, M. Cooper, and A. Girgensohn, Creating music videos using automatic media analysis, *In Proc. ACM Multimedia*, (2002), 553 - 560.
- [11] X. Hua, L. Lu, and H. Zhang, Automatic music video generation based on temporal pattern analysis, *In Proc. ACM MultiMedia*, (2004), 472 - 475.
- [12] J.C. Yoon, I.K. Lee and S. Byun, Automated music video generation using multi-level feature-based segmentation, *Multimedia Tools and Applications* 41, 2(2009), 197 - 214.

- [13] J. Wang , E. Chng , C.S. Xu , H.Q. Lu, and Q. Tian, Generation of personalized music sports video using multimodal cues, *IEEE Transactions on Multimedia* 9, 3(2007), 576 – 588.
- [14] A. Money and H. Agius, Analysing user physiological responses for affective video summarisation. *Displays* 30, 2(2009), 59 – 70.
- [15] H. Joho, J.M. Jose, R. Valenti and N. Sebe, Exploiting facial expressions for affective video summarisation, In *Proc. International Conference on Image and Video Retrieval*, (2009).
- [16] W.T. Peng, W.J. Huang, W.T. Chu, C.N. Chou, W.Y. Chang, C.H. Chang, Y.P. Hung, A user experience model for home video summarization, In *Proc. International Multimedia Modeling Conference*, (2009), 484 – 495.
- [17] CyberLink PowerDirector, CyberLink Corporation Inc., <http://www.cyberlink.com/>
- [18] F. Shipman, A. Girgensohn and L. Wilcox, Authoring, viewing, and generating hypervideo: an overview of Hyper-Hitchcock, *ACM Transactions on Multimedia Computing, Communications, and Applications* 5, 2(2008), 1 – 19.
- [19] MuVee AutoProducer, MuVee Technologies Pte. Ltd, <http://www.muvee.com/en>.
- [20] W.T. Peng, Y.H. Chiang, W.T. Chu, W.J. Huang, W.L. Chang, P.C. Huang, and Y.P. Hung, Aesthetics-based automatic home video skimming system, In *Proc. International Multimedia Modeling Conference*, (2008), 186 – 197.
- [21] M. Argyle, *Bodily communication*, Methuen & Co. Ltd, 1988.
- [22] S. Sirohey, A. Rosenfeld, Eye detection in a face image using linear and nonlinear filters, *Pattern Recognition* 34, 7(2001), 1367 – 1391.

- [23] T. Takagi and M. Sugeno, Fuzzy identification of systems and its applications to modeling and control, *IEEE Transactions on Systems, Man and Cybernetics* 15, 1(1985), 116 – 132.
- [24] P. Ekman, W.V. Friesen, *Unmasking the face*, Prentice-Hall, (1975).
- [25] W.Y. Chang, C.S. Chen, and Y.P. Hung, Analyzing facial expression by fusing manifolds, In *Proc. Asian Conference on Computer Vision Conference*, (2007), 621 – 630.
- [26] P. Masri. “Computer modeling of sound for transformation and synthesis of musical signal,” Ph.D. dissertation, University of Bristol, UK, (1996).
- [27] S. Dixon, “Onset detection revisited,” *Proceedings of International Conference on Digital Audio Effects*, (2006).
- [28] Vezhnevets, V. and Degtiareva, A Robust and accurate eye contour extraction, *In Proc. Graphicon*, (2003), 81 – 84.
- [29] Yuille, A., Hallinan, P., and Cohen, D. Feature extraction from faces using deformable templates. *International Journal of Computer Vision* 8, 2(1992), 99–111.
- [30] M. Fischler and R. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 24, 6 (1981), 381 – 395.
- [31] R.B. Goldstein, E. Peli, S. Lerner, and G. Luo, Eye movements while watching a video: Comparisons across viewer groups. *Vision Science Society*, (2004).
- [32] BioID Technology Research. The BioID Face Database. <http://www.bioid.com>, (2001).
- [33] O. Jesorsky, K. J. Kirchbergand, and R. Frischholz. Robust face detection using the hausdorff distance. *In Proc. Audio and Video Based Person Authentication*, (2001), 90 – 95.

- [34] R. Valenti and T. Gevers. Accurate eye center location and tracking using isophote curvature. *In Proc. IEEE Computer Vision and Pattern Recognition*, (2008), 1 – 8.
- [35] M. Türkan, M. Pardás, and A. E. Cetin. Human eye localization using edge projection. *In Proc. VISAPP*, (2007), 410 – 415.
- [36] S. Asteriadis, N. Nikolaidis, A. Hajdu, and I. Pitas. An eye detection algorithm using pixel to edge information. *In Proc. Int. Symposium on Control, Communications and Signal Processing*, (2006).
- [37] L. Bai, L. Shen, and Y. Wang. A novel eye location algorithm based on radial symmetry transform. *In Proc. Pattern Recognition*, (2006), 511 – 514.
- [38] P. Campadelli, R. Lanzarotti, and G. Lipori. Precise eye localization through a general-to-specific model definition. *In Proc. BMVC*, (2006).
- [39] D. Cristinacce, T. Cootes, and I. Scott. A multi-stage approach to facial feature detection. *In Proc. BMVC*, (2004), 277 – 286.
- [40] H. Tong, M. Li, H.J. Zhang, and C. Zhang, “Blur detection for digital images using wavelet transform.” *Proceedings of IEEE International Conference on Multimedia & Expo*, (2004), 17 – 20.
- [41] A. Hanjalic, “Shot-boundary detection: unraveled and resolved?” *IEEE Transactions on Circuits and Systems for Video Technology* 12, (2002), 90 – 105.
- [42] F. Dibos, C. Jonchery, G. Koepfler, “Camera motion estimation through quadratic optical flow approximation.” *Université de PARIS – DAUPHINE*, (2005).