國立台灣師範大學
資訊工程研究所
博士論文

智慧型演講錄製系統
Smart Lecture Recording System

研究生：羅安鈞
指導教授：陳世旺 博士

中華民國　106　年　8　月

摘要

近年來由於數位學習（或遠距教學）的發展，從高度發達的大都市到偏遠低開發國家，都可為學習者提供了平等的機會。而演講錄製系統在收集數位學習的內容資料中發揮著至關重要的作用。然而隨著數位學習的蓬勃發展，數位內容的缺乏以及專業錄製團隊人員等正在成為一個大問題。這項研究提出了一個智慧型的演講錄製系統，可以自動錄製與人類團隊相同質量水平的內容，並減少錄製人員不足的問題。

本研究所提出的智慧型演講錄製系統由三個主要元件系統組成，分別稱為虛擬攝影師、虛擬導演和虛實對位。前兩個元件虛擬攝影師和虛擬導演是線上執行的系統，而虛實對位是屬於離線後製的元件。而虛擬攝影師元件可進一步分為三個子系統：演講者攝影師，觀眾攝影師和演講廳攝影師。所有這些子系統都是自動運作，包括選擇拍攝目標、追踪拍攝、特殊事件偵測等功能。這三個子系統拍攝的視訊將全部傳輸到虛擬導演系統，虛擬導演則選擇最具代表性的畫面錄製或直播。我們將虛擬導演的此功能稱為:選鏡。選鏡的功能主要是對來自虛擬攝影師的視訊作內容分析，並通過反傳播神經網絡特徵的機器學習過程進行畫面選擇的決策。此外，虛擬導演系統具有另一個關鍵功能:視覺指導，通過它可以模仿人類導演和現實世界中的人類攝影師之間的溝通。

在完成一段實況的演講錄製後，有時會在演講的錄音集中附加額外的內容或素材，以增加其表現力和可看性。所以本研究另外開發了一個稱為虛實對位系統的後期製作元件，用於實際拍攝影片與虛擬物件的合成。該系統以深度攝影機作為深度感測設備，協助真實世界的彩色攝影機和虛擬世界的攝影機同步對位。虛實對位系統有三個主要執行流程：時間深度融合、攝影機跟踪和虛實合成預覽。由深度相機獲取的深度影像經由時間深度融合被疊合成場景的 3D 構造。再藉由 3D 場景的結構與深度攝影機的相對關係，推導出彩色攝影機的移動軌跡。此軌跡則用於引導虛擬攝影機與真實攝影機同步移動完成對位，將虛擬物件投影並生成虛擬影像，將生成的虛擬圖像疊加在由彩色攝影機獲取的真實圖像上，所得到的圖像稱為虛實合一的預覽圖像。

本研究進行了一系列實際演講錄製實驗，而實驗數據顯示我們所提出的智慧型演講錄製系統可以模擬出近似於真正的人類團隊所採取的拍攝、選鏡技術。我們也認為這套系統可不限於演講錄製;如果可以搭配適當的訓練資料，也可以適合錄製舞台表演，音樂會，運動比賽和產品發表會等場合。

關鍵字：智慧型演講錄製系統、虛擬攝影師、虛擬導播、虛實對位、選鏡、視覺指導、虛實預覽。

Abstract

Nowadays, e-learning (or distance learning) provides equal opportunities for learners in locations ranging from highly developed metropolises to remote less-developed countries. Lecture recording systems play a vital role in collecting spoken discourse for e-learning. However, in view of the growing development of e-learning, the lack of content is becoming a problem. This research presents a smart lecture recording (SLR) system that can record orations at the same level of quality as a human team, but with a reduced degree of human involvement.

The proposed SLR system is composed of three principal components, referred to as virtual cameraman (VC), virtual director (VD), and virtual-real match moving (VRMM), respectively. The first two components, VC and VD, are online components, whereas the VRMM component is offline. The VC component is further divided into three subsystems: speaker cameraman (SC), audience cameraman (AC), and hall cameraman (HC). All these subsystems are automatic, and can take actions that include target and event detection, tracking, and view searching. The videos taken by these three subsystems are all forwarded to the VD system, in which the representative shot is chosen for recording or direct broadcasting. We refer to this function of the VD system as shot selection. The shot selection function operates based on the content analysis of the videos transmitted from the VC component. The capability of content analysis is pre-trained through a machine-learning process characterized by the counter-propagation neural network. In addition, the VD system possesses another pivotal function of visual instruction, through which it imitates the communication between a human director and human cameramen in the real world.

Having completed a live speech recording, it is often necessary to include

additional contents or materials in the shot collection of the speech in order to increase its expressivity and vitality. In this context, we develop a post-production component called the virtual-real match moving (VRMM) system for graphic/ stereoscopic image composition. The input data to this system is provided by the equipment constituting a color camera and a depth camera. There are three major processes: temporal depth fusion, camera tracking, and virtual-real synthesis preview, involved in the VRMM subsystem. During temporal depth fusion, the depth images acquired by the depth camera are fused to lead to a 3D construction of the scene. Based on the constructed scene, the pose of the color camera is determined, which is next used to direct a virtual camera to generate synthetic images of a given 3D object model. The generated images are superimposed upon the real images acquired by the color camera. The resultant images are called preview images.

A series of experiments for real lecture has been conducted. The results showed that the proposed SLR system can provide oration records close to some extend to those taken by real human teams. We believe that the proposed system may not be limited to live speeches; if it can be configured with appropriate training materials, it may also be suitable for recording stage performance, concerts, athletic competitions, and product launches.

Keywords: *Smart lecture recording system, Virtual cameraman, Virtual director, Virtual-real match moving, Shot selection, Visual instruction, Preview images.*

誌謝

在博士研究生活中，承蒙眾多人的幫助，非常抱歉無法親自一一答謝。博士班雖然時間很漫長，但是如今我非常高興當初毅然決定就讀的選擇。誠摯感謝我的指導教授陳世旺老師，陳老師好學不倦的精神深深影響著我，提醒我不斷的思考和吸收新知才能突破以往的思維，懂得如何獨立解決問題。此外，陳老師也體諒我在職就讀的身份，不斷的配合我的工作更改會談時間，甚至陪我一起討論工作研發上遇到的難題，博士班學到的知識技術也幫助我在工作升遷上更加的順利。陳老師所傳授的不僅僅只有學術領域，更不時於會談之餘教導我做人處事的道理，讓我在追求學問的過程中不忘重視人情，在我工作與生活受到挫折時，每次都能即時給我最溫暖的安慰，讓實驗室對我來說是另一個家、另一個避風港。

在此特別感謝方瓊瑤老師，方老師常常不厭其煩的告知我們研究上必須注意的事項，也感謝方瓊瑤老師在計畫口試與論文口試時協助與叮嚀我們分配相關的工作、文件與學校規章。也感謝系主任陳柏琳老師，我們的演講者錄製系統實驗中提供了不少實用的意見與方法。感謝台大傅楸善教授、交大吳炳飛教授、清大陳朝欽教授在論文與研究方法上的寶貴意見與指導，也讓我了解不少演講錄製系統上可以改進的地方。

感謝 IPCV 實驗室的所有成員，鍾允中學長、王俊民學長與張祥利學長多年來的指導，一起參與討論研究上面臨的困難，時而提供我解決的方法或技巧，讓我的研究得以順利進行與完成。感謝彥佑學弟、俊億學弟與佳儒學妹在演講者錄製系統實作上的付出與盡心。感謝軒嘉學弟、淳雅學妹、宇珊學妹的大力幫忙和支持。也感謝 CVIU 實驗室的所有學弟妹，如:雯琳、靖允、銘仁、家安在我演講錄製實驗與口試過程中揮汗如雨的幫忙，非常感動。

特別感謝我的摯友陳柏網博士，我永遠不會忘記七年前在迷惘中的兩人互相鼓勵所做的羅博與陳博約定，現在我們倆都終於順利畢業取得博士學位，達成了

約定，沒有你，我不會開始；沒有你，我堅持不到現在，你是我最好的朋友也是貴人。我也要感謝我的家人與妻子映均，持續的體諒著我，請原諒我因為忙著工作與學業，沒能好好陪伴你們。

最後，非常感謝大家一直以來對於我的鼓勵與讚美，以及批評與指教，尤其是對於我的包容，希望往後每位師長、每位夥伴、每位朋友都能夠繼續互相扶持，在接下來的人生都能一路順遂，謝謝大家。

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

1.1 Motivation

In recent years, fiber optic networks have become well developed, and wireless Internet access points have been deployed widely. At present, users are not required sit in front of a desktop computer and access content through a wired network; they can use a hand-held device (Figure 1.1) or a laptop in places that have LTE or Wi-Fi signals, such as convenience stores, cafes, and mass transit stations.

Figure 1.1. Watching video on a smartphone.

Therefore, the market for network-based multimedia digital content (especially e-learning) is growing. With the growth of e-learning systems, more and more people hope to record their presentations or speeches to preserve them or to upload them on the Internet to share with other people around the world. Taiwan has developed e-learning for nearly a decade, and the Ministry of Economic Affairs allocated 200 million NTD to further develop e-campus technologies over the last two years. Moreover, Delta Electronics & Wistron Electronics made a total R & D investment of approximately 600 million NTD. Before the end of 2016, the value of the e-learning industry is expected to reach 84.6 billion NTD; Southeast Asia is expected to be the first major area to enter the overseas market. There seems to be universal agreement that the worldwide e-learning market will show rapid and dramatic growth over the next five years. The worldwide market for self-paced e-learning reached $35.6 billion in

2011. The five-year compound annual growth rate is estimated at approximately 7.6%, so revenues are expected reach some $51.5 billion by 2016.

A definition of self-paced learning is education in which each learner studies at his or her own pace, without a fixed starting date or regularly scheduled assignment completion dates in common with other students enrolled in the same program. However, a self-paced learning course may have a fixed overall completion timeframe. In view of the growing development of e-learning, the lack of content is becoming a problem.

In general, companies, schools, and other organizations often hold a wide variety of presentations, such as product presentations (Figure 1.2), academic speech seminars, and so on. These presentations are vital assets that are worth recording for on-line audiences or future audiences who will want to review archived videos. Recording these lectures usually requires a professional filming team that can perform filming and shot selection.

Figure 1.2. Product presentation for Apple smartphones.

When recording a lecture, the filming team has three main jobs: the first job is to survey the venue, before the speech begins, so that the equipment can be arranged effectively. The second job is filming the speech; during the presentation, each cameraman controls a camera and films various shots in different situations. The third

job requires that the live video shots from different cameramen be transmitted to a control room; the director does the third job by selecting the most representative shots from those videos. The main responsibilities of the director are shot selection and visual instruction. Whether a presentation can be successfully recorded depends on whether the director has sufficient experience or visual storytelling ability. Finally, after the end of the speech, director splices the shot division, does the postproduction process, and then uploads the video to a server; the audience can download the video from the server. Although speeches recorded and uploaded to servers are highly convenient for numerous audiences, the cost is very high.

1.2 Literature Review

A great number of people have used e-learning and mobile learning services, such as open courseware (OCW) and lecture websites. When recording a lecture or a course, at least two cameramen and one director are needed. Often, to make the program's content more comprehensive, multiple cameras shoot video from different viewpoints and all video signals are sent to a video mixer (or switcher) in a control room. The director chooses the most suitable shot to broadcast from the mixer. We call this work "shot selection." The workflow of program recording is shown in Fig. 1.3.

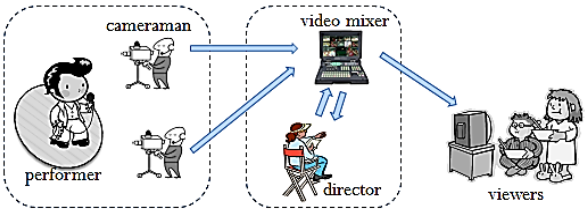


Figure 1.3. Workflow of program recording.

Considering the cost of the traditional system, Rowe [1] classified the cost into two major parts: fixed costs, like computers, cameras, and microphones, and unfixed costs like cameramen, directors, and editors. Generally, the fixed costs must be paid

once, but the unfixed costs must be paid for every production. The authors of [2] mentioned that their teams cost more than US$500 for each Microsoft Corporation lecture. Consequently, the authors of [3], [4], and [5] have discussed the reduction of unfixed costs. To reduce the costs of recording a lecture or a course, some automatic systems of recording have been proposed.

Recording lectures automatically has become popular in recent years, but most of the systems on the market still use static cameras without automatic camera control. Cruz [6] published the earliest proposal for an automatic lecture recording system; because the author only used one camera to shoot the scene, the output looked tedious. Bianchi [4] improved on this drawback; the author used several cameras that were able to detect and track the speaker automatically to shoot a lecture. In [7] and [8], the authors used a Microsoft library to construct iCam systems, and the authors of [9] discussed how to reduce the unfixed costs of recording and broadcasting speeches and presentations.

The virtual cameraman (VC) is based on the idea of automatically detecting the positions and postures of key speakers and shooting adequate views of those speakers. Onishi [10] not only detected and tracked the speaker but also recognized the actions of the speaker. For posture recognition, the authors of [11] extracted user images from various input images and used those images to train a neural fuzzy network to recognize corresponding postures. The authors of [12] let users wear magnetic sensors that were able to locate the users' hands, shoulders, and abdomens, and then recognized the users' behaviors by the location information transmitted by those sensors. The researchers involved in [13] and [14] used KINECT, a range sensor developed by Microsoft, to obtain users' 3-D skeleton information and recognized their actions using posture matching and an SVM classifier, respectively. Huang [15] tried to track the human head and arm using a single camera in cluttered environments. An earlier publication

reported the construction of a VC system [16-22]. In another article, human faces were tracked using a mean-shift algorithm [23]. However, no prior research considered utilizing content analysis to perform automatic shot selection between video shots from different cameramen.

In 1969, Paul Ekman and Wallace V. Friesen published a paper on the psychology of nonverbal behavior [24] indicating that gesture plays a vital role in terms of nonverbal communication. In addition, they classified nonverbal behaviors into several categories; each category represented a predefined meaning. However, the hand gestures of any speaker differ from the gestures of other speakers. Therefore, it is impossible to require the VC to react to every gesture. Depending on the specific needs of a use case, the meanings of hand gestures may require discussion and classification into specific categories. If gestures have been classified for a predefined scenario, the system can control the camera action appropriately.

Shot selection plays an essential role in the success of a program using multiple cameras. A video mixer is a platform that allows a director to transmit pictures, with or without additional functions such as special effects and titles. The job of the director is to convey the information of the speech or program faithfully to the audience.

Shot selection is a key task that demands experience and skill; at each moment, the director must process multiple live video feeds and choose the most suitable shot from all of them. If the most suitable shot is already being shown, the director must maintain the current feed, but if the most suitable shot is not being shown, the director must cut to the feed with the most suitable shot. The director selects the most suitable shot (called the representative shot) of all received shots by looking for clues that may interest the viewer and then transmit the representative shot, either to an on-air broadcast or to a recording medium. To make a satisfactory decision from multiple inputs, an expert director should have served as a cameraman, a video editor, and a technical adviser. An

expert director with manifold experience has the ability to analyze content and select shots that conform to photographic aesthetics and group psychology.

In automatic shot selection, Gleicher [25] used virtual videography to edit videos, whereas Okuni cut a video and extracted meaningful shots to make a new one [26]. Kumano [27] analyzed the behavior of camera motion in terms of video grammar and combined multiple pieces of footage into a more complete video. Wang [28] analyzed a video content by using a genetic algorithm to analyze the structure of the footage and the use of photography; Wang combined separate but relevant video selections into a complete movie. Machnicki [29] integrated diverse directorial functions into an automatic system and called it a "virtual director" (VD). The aforementioned research studies were off-line works; thus, they could not select shots in real time. Furthermore, those studies did not implement any communication between VC and VD to emulated the communication between real cameramen and directors.

Before the VD performs automatic shot selection, the system must process content analysis, gather information, and put the video into context for the VD. In content analysis, the first step to automatic photographic analysis is to extract salient objects. Goferman [30] analyzed high-contrast phenomena at the edges of areas within images; simulated human vision, applied to a static image, detected colors, faces, and other information. Fang [31] discussed spatiotemporal attention applied to images by considering human visual stimuli that simulate information. The authors of [32] attempted to detect the movements and gestures of a speaker and proposed a method of automatic image analysis to describe the behavior of the speaker, but they did not discuss any automatic shot selection between multiple cameras. The authors of [4, 29] studied the camera skills required for a photographer. By camera manipulation, they improved the appearance of some films regarding liveliness and smoothness, but they did not perform content analysis or learn shooting rules from real photographers

automatically. In the present research, to increase the attractiveness of the final video output, the proposed system must consider aesthetics and the rules of photography. An ideal system would apply a machine-learning algorithm [33, 34] to learn rules from professionals, such as fuzzy control [35], artificial neural network [36, 37], and deep learning [38-43]. Deep learning, especially convolutional neural networks (CNN)[44-50], have been applied to fields including image classification [51, 52], computer vision [53, 54], speech recognition [55, 56], natural language processing [57, 58], and autonomous driving [59-61], where they produced results comparable to and in some cases superior to human experts. However, the limitation of CNN is its local feature learning property where content analysis usually require global and temporal informations.

The automation of shot selection can be seen as a decision-making process that takes various shots and information as inputs and returns a single suitable shot as output. The learning process can involve learning shot selection skills from real directors. Hecht [62] introduced a counter propagation network (CPN) as a type of supervised machine-learning technique. If the training data was relevant to the input data, then the CPN applied classification to process the input data into output data quickly in the testing stage. However, the CPN yielded poor results when the input data were heterogeneous. Multiple kernel learning (MKL) [63, 64] is a machine-learning method that can combine predefined kernels for each individual data source by selecting an optimal kernel and parameters. MKL shows a better result in heterogeneous data classification than learning methods without kernel transforms, especially for data in different dimensions and different ranges. However, the MKL training process requires long periods of time for running a complex optimization algorithm. We utilize the fast convergence learning property of the CPN network to simplify the complex

optimization process of MKL and produce an approximate result. At the same time, the kernel transformation of MKL also improves the classification accuracy of the CPN.

Ideally, the VC and VD subsystems should exchange two-way communications. Therefore, the proposed VD subsystem should actively give visual instructions or advice to the VC at appropriate times instead of passively receiving video shots and signals. The VC that shoots the speaker should pass all relevant information, including the position and hand gestures of the speaker, to the VD subsystem. For VC subsystems taking the audience view and hall view, the shooting target may not be a specific individual. Therefore, audience view and hall view VCs may be focused on the detection of crowd motion. We follow [65] to establish a method of creating entropy models to quantify crowd motion and locate any abnormal behavior in crowd scenes.

Because of the restrictions on shot selection in real time, it is a challenge to design a VD that automatically evaluates the quality of views and immediately makes a consistent decision that results in a steady video. Therefore, an efficient and robust VD system must be implemented. Liu [66] proposed a finite state machine (FSM) that imitated a real director to select suitable shots. Liu's system operated by state transitions, and all states were defined carefully without exceptional inputs and undefined states. To avoid any deadlocks or empty shots in the proposed smart lecture recording (SLR) system, we also applied an FSM to our VD subsystem.

The proposed SLR system consists of three VC subsystems and a VD subsystem; it composes three different views from three different VC subsystems into one view in our interface. Not only can the VC automatically track speakers on stage; the VC can also perform posture recognition from depth images (or range images) of speakers. In addition, this study considers a number of possible scenarios and presents a system that automatically develops a set of reasonable rules for camera work. After a VC subsystem integrates all relevant speaker information, that VC system calculates the optimal shot

according to the rules of photographic opportunities, operates a pan, tilt, and zoom (PTZ) camera to take the optimal shot, and sends messages to inform the VD. After receiving the messages, the VD can choose the single best shot by the established rules and present that shot to the audience. In addition, speakers using laser pens or batons can also control the PTZ camera to shoot specific content, increasing the diversity of screen displays and interactions. Given multiple videos from the VCs, an advanced VD must automatically analyze all available information and choose the best shot by considering photographic aesthetics, optics, scenery continuity, and action continuity. In the shot selection stage, the CPN network executes the decision module for shot selection. The training data of the CPN network is composed of decisions from a real director. Therefore, the VD subsystem can learn shot selection rules from that director. However, the CPN network tends to yield poor results if the input data are heterogeneous. Therefore, we use MKL to transform all heterogeneous data from different content analysis methods into the same dimensions and ranges before those data are sent to the CPN network; this increases the accuracy of shot selection.

If the video signal is not broadcast as live video, then postproduction processes such as video editing, virtual–real synthesis, and subtitling can be executed. Related research on automatic video editing has been published in recent years. For example, Gleicher [65] proposed virtual videography, and Liu [66] tried to re-edit multiple videos into meaningful video clips. Numerous similar studies, such as those of Okuni [26] and Kumano [27], investigated similar topics. Even though studies have been able to find meaningful video clips and combine those clips into new videos, they are exercises in postproduction, not real-time editing.

Visual effects (VFX) involving virtual-real match-moving (VRMM) have been commonly applied in contemporary media, particularly in public entertainment, such as movies and commercials. To achieve perfect virtual-real synthesis, all relevant

parameters and the moving trajectory of the camera should be noted. A camera for capturing real objects in the real scene must be registered, and its parameters must exactly match those of the virtual camera in the virtual scene, to prevent spatial disorientation in the composite result, which shows a real object in a virtual scene (see Fig. 1.4). Conventionally, the match-moving [67] tedious operations for registering and representing real objects in virtual scenes are performed in postproduction. This work, especially the postproduction of stereo film, is highly manpower-intensive.

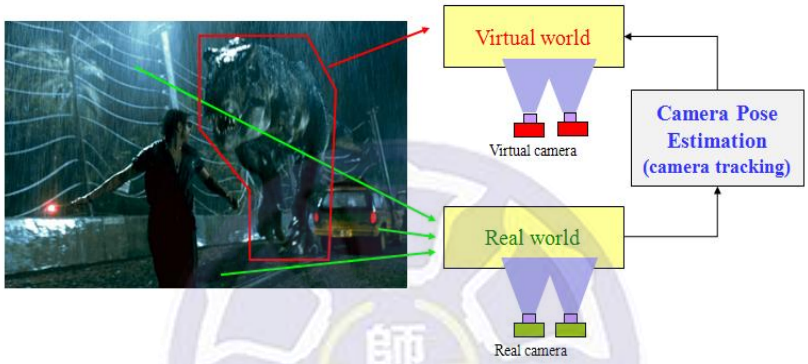


Figure 1.4. Camera tracking technology for VRMM.

Moreover, if the image information captured is insufficient for reconstructing the camera trajectory in a virtual scene, or if certain errors exist in the parameters registered on-site, the whole image recording of the real object must be done all over again. Clearly, the prolongation of production causes higher costs. An improved method or system of image composition, therefore, is required not only for reducing postproduction labor costs but also for preventing insufficient trajectory information and erroneous parameter registration at the early stage.

There are already numerous camera trajectory tracking and camera self-positioning techniques available on the market [68], such as structure-from-motion (SFM) [69] and simultaneous localization and mapping (SLAM) [70-72] for common cameras. KinectFusion from Microsoft is a self-positioning technique that combines a continuous

stream of 3D data from a depth camera. However, a scene to be filmed with the previously mentioned techniques must remain static; otherwise, the tracked trajectories are affected by the objects moving in the scene, and thus, the precision of the tracking is adversely affected. For VRMM in the present work, we modify the temporal depth fusion of KinectFusion, to apply it to dynamic environments. We develop a human skeleton detection method and a spatiotemporal attention (STA) analysis method to reduce the noise from moving objects and human characters in the scene.

1.3 Organization of this Thesis

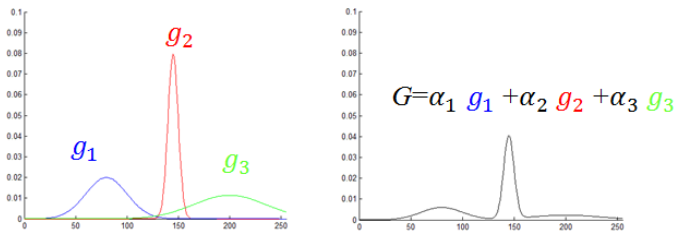
This study is organized as follows. The mathematical fundamentals are introduced in Chapter 2. Chapter 3 presents the system hardware architecture and software organization. Chapter 4 shows how to implement three VCs. Chapter 5 describes how the VD performs shot selection and visual instruction. Chapter 6 documents the details of VRMM. Chapter 7 presents the experimental results. The final chapter covers conclusions and future work.

Chapter 2
Mathematical Fundamentals

In this chapter, the mathematical fundamentals utilized in this thesis are addressed, including Gaussian mixture model is discussed in Section 2.1; finite automata theory is introduced in Section 2.2, the spatiotemporal attention neural network is detailed in Section 2.3, a multiple kernel learning method is introduced in Section 2.4, and the counter propagation neural network is presented in the last section.

2.1 Gaussian Mixture Model

A Gaussian mixture model (GMM) is a probabilistic model that assumes relevant data points can be formulated by a linear combination of multiple Gaussian probability density functions. The model can smoothly approximate the density distributions of arbitrary shapes. In this study, we use GMMs to describe postures of humans for the purpose of posture recognition. Figure 2.1(a) shows three individual Gaussian probability density functions; Figure 2.1(b) shows their mixture.



(a) (b)

Figure 2.1. Example of a GMM (a) Three individual Gaussian probability density functions, (b) The GMM of the three Gaussian probability density functions.

Suppose that we have a set of points $X = \{x_i\},\ i = 1, \dots, n$ in a d-dimensional space. We seek K Gaussian distributions $G_1, G_2, \dots, G_K$ that best represent x_i with

K corresponding contribution weights α_k where $\sum_{k=1}^{K} \alpha_k = 1$. The probability density function is defined by weighed sum of the G_k:

$$p(x_i) = \sum_{k=1}^{K} \alpha_k \; p(x_i|G_k). \tag{2.1}$$

The probability density function expressed in this way is called a Gaussian mixture model.

The probability density function of the distribution generated a point x_i:

$$p(x_i|G_k) = \frac{1}{\frac{d}{2\pi^2}|\Sigma_k|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x_i - \mu_k)^T {\Sigma_k}^{-1}(x_i - \mu_k)\right], \tag{2.2}$$

where μ_k is the mean of the density function, and Σ_k denotes the covariance matrix of the density function. These parameters determine the characteristics of this density function, such as the center, shape, width, and direction of the density function. Hence, the sum weighted contributions of all the G_k is defined as follows:

$$p(x_i) = \sum_{k=1}^{K} \frac{\alpha_k}{\frac{d}{2\pi^2}|\Sigma_k|^{\frac{1}{2}}} \exp\left[-\frac{1}{2}(x_i - \mu_k)^T {\Sigma_k}^{-1}(x_i - \mu_k)\right]. \tag{2.3}$$

Assume that $X = \{x_i\},\ i = 1, \dots, n$ are independent of each other. Therefore, the probability density of X is defined as follows:

$$p(X) = \prod_{i=1}^{n} p(x_i) \tag{2.4}$$

The problem is, given X, what are optimal values of $\alpha_k, \mu_k, \Sigma_k$? However, the processing steps of the optimization are nontrivial. A simpler alternative algorithm to estimate these parameters is called the expectation-maximization (EM). For a detailed exposition of EM, please refer to [73]. From derivation of [73], we can obtain the probability of the k^{th} Gaussian:

$$p_{ik} = \frac{\alpha_k p(x_i|G_k)}{\sum_{j=1}^{K} \alpha_j p(x_i|G_j)}. \tag{2.5}$$

And the new parameters are:

$$\alpha_k^{\text{new}} = \frac{1}{n} \quad \sum_{i=1}^{n} p_{ik}, \tag{2.6}$$

$$\mu_k^{new} = \frac{\sum_{i=1}^{n} p_{ik} x_i}{\sum_{i=1}^{n} p_{ik}}, \tag{2.7}$$

$$\Sigma_k^{new} = \frac{\sum_{i=1}^{n} p_{ik}(x_i - \mu_k^{new})(x_i - \mu_k^{new})^T}{\sum_{i=1}^{n} p_{ik}}. \quad (2.8)$$

By Equations 2.5–2.8, the EM iterative steps of the GMM procedure are listed as follows:

Step1. select the target number of Gaussians K.

Step2. initialize K Gaussians. Usually, we let $\alpha_k = \frac{1}{K}$, calculate the data cluster center by a K-means algorithm, and set μ_k and Σ_k as the initial values.

Step3. *expectation*: Calculate for each data point x_i the $p_{ik}$from the μ_k and Σ_k.

Step4. *maximization*: Update the Gaussian parameters: Equations 2.2-2.4

Step5. iterate from step3. Until convergence.

2.2 Finite Automata Theory

In this section, we address the basics of the finite automata theory which are utilized in this study. This will include how to convert a nondeterministic finite automaton (NFA) into a deterministic finite automaton (DFA), and how to simplify a DFA into a simplified finite automaton (SFA).

A. Finite state machine

The finite state machine (FSM), also called the finite state automata, is an efficient and simple mathematical model often used in logic circuits and computer programs. The FSM is defined by a finite number of states, input operations, and a transition function. For any defined triggering event, the current state transitions to the appropriate state.

B. Nondeterministic finite automaton

An NFA is expressed by a quintuple $M = (K, \Sigma, \Delta, s_0, F)$, where K represents a finite set of states, Σ is the input symbol collection, Δ is the transfer relation, s_0 is the initial state, and F is the set of final states. The NFA is one type of FSM. For each input symbol, an NFA transitions to a new state until all input symbols have been consumed. For some state and input symbol, the next state may be nothing, or one state,

or multiple possible states. The NFA consists of the following four transitions: empty transition, multiple input transition, ambiguous transition, and missing transition. The NFA is easier to construct, because NFAs can be constructed from any regular expression using Thompson's construction algorithm. Figure 2.2 is an example of NFA transition diagram.

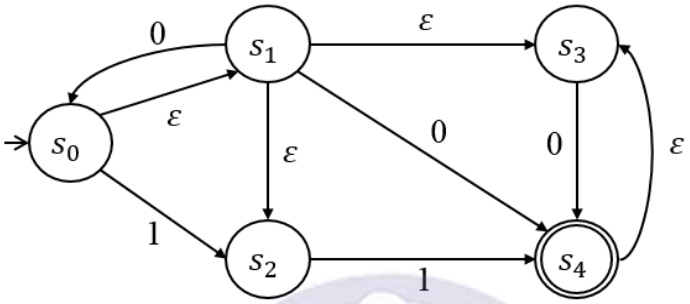


Figure 2.2. NFA transition diagram with empty transition, multiple input transition, ambiguous transition, and missing transition.

C. Deterministic finite automaton

The deterministic FSM can be referred to as a DFA. A DFA is described by a quintuple $M = (K, \Sigma, \delta, s_0, F)$, where K represents a finite set of states, Σ is the input symbol collection, δ is the transfer function, s_0 is the initial state, and F is the set of final states. The rules according to which the automaton M picks its next state are encoded into the transition function.

Every NFA has an equivalent DFA. The conversion is using the subset construction method, please refer to [74] for details. After the NFA has been converted to a DFA, the functionality of the new DFA is equivalent to that of the original NFA. The main purpose of the conversion is to eliminate the uncertainty of the NFA from ambiguous transitions. The system is easier to implement and debug if we design the system using a DFA.

D. Conversion from NFA to DFA

After the NFA is specified, it can be converted into an equivalent DFA. Using the subset construction algorithm, each NFA can be translated to an equivalent DFA. Given a transition diagram, the steps of the subset construction algorithm are as follows:

Step 1. Separate all multiple input transitions.

Step 2. Check whether each state has an empty transition that can transition without an input symbol.

Step 3. Check the input symbol and reachable state from a given state and store them in a transition table.

Step 4. Check whether any new state exists in the table. We try to find states that can be merged into a new state; if a state that can be merged is found, then repeat Step 3 with that state.

Step 5. Repeat Step 3 and Step 4 until all the possible states are merged.

Step 6. Rename the states.

Step 7. Mark initial state and final state.

After all Steps have been executed, we have converted the NFA into the DFA. Figure2.3 shows an example of DFA transition diagram.

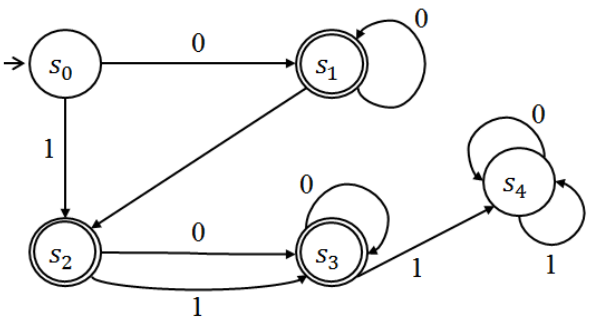


Figure 2.3. DFA transition diagram.

E. Conversion from DFA to SFA

Once we obtain the DFA, we must check whether DFA can be simplified. The simplification algorithm reduces the number of states, and improves the efficiency of

the system.

Step 1. Construct transition table.

Step 2. Partition states according to final and non-final states.

Step 3. Rename components.

Step 4. Find states that can be merged into a new state.

Step 5. For each component of the previous partition,

partition the component according to the next states.

Step 6. Repeat Step 3 to Step 5 until partition count is the same.

Step 7. Rename states, construct table, and draw diagram.

The practical steps for our system design are described in Chapter 3.

2.3 Spatiotemporal Attention Neural Network

The STA [31] neural network is configured as a two-layer network, with one layer for input and one layer for output. The extracted information serves as the input stimuli to a STA network embedded in the perceptual analyzer. The output layer is also referred to as the attention layer. Neurons in the attention layer are arranged into a two-dimensional (2D) array, in which they are interconnected. No direct links connect input neurons to each other, but each neuron is part of the two-layer network. Assume that a 2D Gaussian *G* (see Figure 2.4) is centered at an attention neuron. A weight value links an input neuron with the attention neuron at the center of the Gaussian *G*. If consistent stimuli repeatedly innervate the neural network, a focus of attention is established in the network.

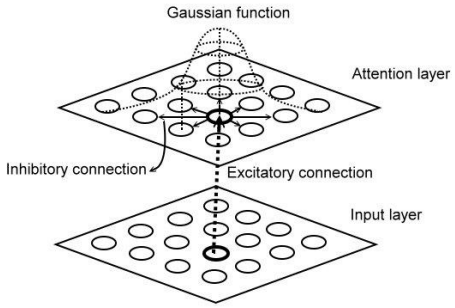


Figure 2.4. STA network.

Figure 2.5 shows the activation of an attention neuron in response to an input stimulus. If the input to the neuron is greater than that neuron's activation threshold within a time interval ΔT, the neuron requires ΔT_{rise} time to reach maximum activation and decays over a time of approximately ΔT_{decay}.

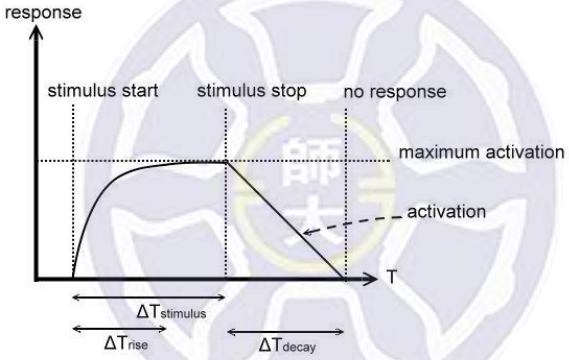


Figure 2.5. Activation of an attention neuron in response to stimuli.

The equation for this activation curve is formulated as follows:

$$STA(x,y,t) = \begin{cases} \min(\rho, STA(x,y,t_0) + \rho \cdot (1 - e^{-\sigma \cdot (t-t_0)})) & \text{, if } A(x,y,t) > T_a \\ STA(x,y,t-1) - \omega & \text{, otherwise} \end{cases} \quad (2.9)$$

where ρ is the maximum activation, σ controls the rate of rise, and ω controls the rate of decay. In addition, t_0 is the start time at which the STA neuron in position (x,y) receives an activation $A(x,y,t_0)$ larger than the threshold T_a.

To detect STA-salient objects in a video sequence, at first, a low-color image and a high-color image are extracted from the input video sequence. A high-color (low-color) image at time *t* preserves the maximum (minimum) color values of the input video sequence up to time *t*. A distinct spatial difference image is then computed for each

input in the STA neural module. Then, we calculate the temporal difference (derivative) image for each spatial difference image. The resulting temporal difference images then serve as inputs to the STA neural network. The process flowchart is shown in Figure 2.6.

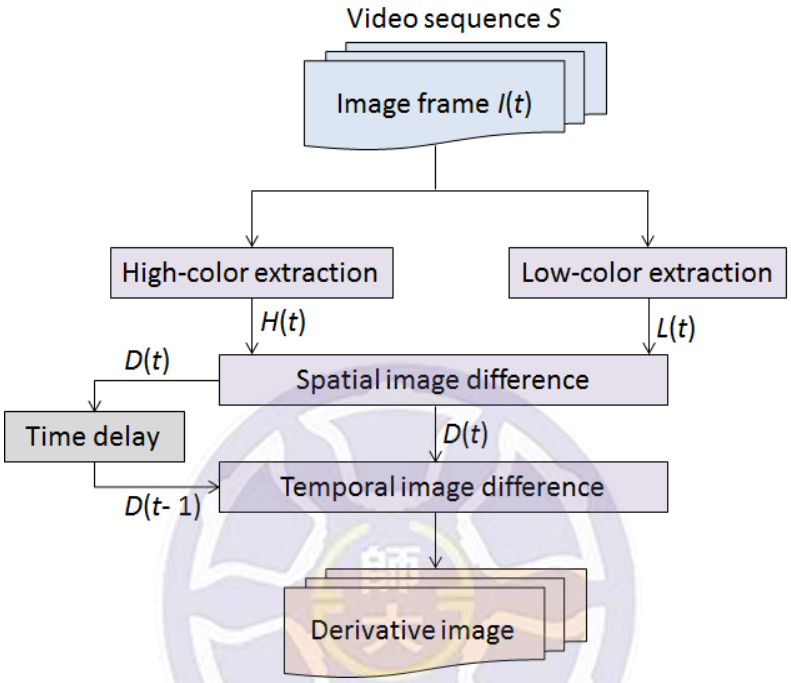


Figure 2.6. Flowchart of STA image acquisition.

2.4 Multiple Kernel Learning

MKL is a method that has been proven to produce excellent classification results when dealing with heterogeneous data from different sources of information with their own dimensions and ranges, especially in large sample spaces. MKL is used in our VD subsystem to improve the accuracy of shot selection.

If the problem is nonlinear, instead of trying to fit a nonlinear model to discriminate the data, we can map the problem to a new space by doing a nonlinear transformation using suitably chosen mapping function and then use a linear model in the new space (see Figure 2.7). Assume that we have the new dimensions calculated through the mapping functions $z = \emptyset(x)$ mapping from the x space to the z space.

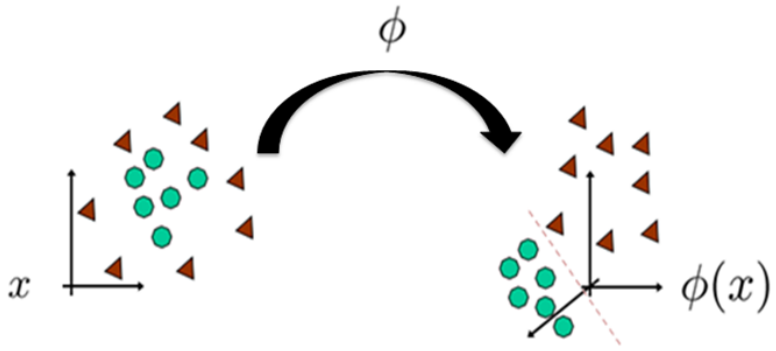


Figure 2.7. Mapping to a new space.

Given a sample $\mathrm{X} = \{(\mathrm{x}_i, \mathrm{y}_i)\}_{i=1}^{n}$. For a binary classification, the classifier can be trained by solving the following quadratic optimization problem:

$$\min_{\xi} \frac{1}{2}\|\mathrm{w}\|_2^2 + C\sum_{i=1}^{N}\xi_i \quad \text{s.t. } y_i(\mathrm{w}\cdot\emptyset(\mathrm{x}_i)+b) \geq 1-\xi_i \tag{2.10}$$

where C is a predefined positive trade-off parameter between model simplicity and classification error and ξ is the vector of slack variables. Instead of solving this optimization problem directly, we use the Lagrangian dual function to obtain the following dual formulation:

$$\max_{\alpha} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j \emptyset(\mathrm{x}_i)\cdot\emptyset(\mathrm{x}_j)$$

$$\text{s.t. } \sum_{i=1}^{N}\alpha_i y_i = 0\,, 0 \leq \alpha_i \leq \mathrm{C},\ \forall i \in \{1,2,\ldots,n\} \tag{2.11}$$

where α is the vector of dual variables corresponding to each separation constraint.

The idea in kernel machine is to replace the inner product of mapping functions, $\emptyset(x_i)\emptyset(x_j)$, by kernel function $K(x_i, x_j)$. Kernels are generally considered to be measures of similarity in the sense that $K(x_i, x_j)$ takes a larger value as x_i and x_j are more "similar" from the point of view of the application. The optimization process applies the following dual formulation:

$$\max_{\alpha} \sum_{i=1}^{N}\alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N}\alpha_i\alpha_j y_i y_j K(x_i, x_j) \quad \text{s.t. } \sum_{i=1}^{N}\alpha_i y_i = 0\,, 0 \leq \alpha_i \leq \mathrm{C}. \tag{2.12}$$

Moreover, the optimization process of a general kernel, which is represented in matrix form, is as follows:

$$\max_{\alpha,\alpha^{\mathrm{T}}y=0} \alpha^{\mathrm{T}}e - \frac{1}{2}\alpha^{\mathrm{T}}K\alpha \quad s.t.\ 0 \le \alpha_i \le C \tag{2.13}$$

where K is the kernel matrix. The kernel matrix K is the representation of the similarity between all pairs of data points. Because the elements of the kernel matrix are defined by the inner product from pairwise comparison, the kernel matrix is a symmetric positive definite matrix that forms a convex cone. Here we can find a property of the kernel matrix: any symmetric positive definite matrix specifies a kernel matrix, and every kernel matrix is a symmetric positive definite matrix.

It is possible to construct new kernels by combining multiple simpler kernels. Such that, we can fuse heterogeneous information from different sources. Each kernel measures similarity according to its domain. Assuming *M* different sources exist, and each source has its own base kernel matrix K_m, the kernel matrix *K* is defined as:

$$K = \sum_{\mathrm{m}=1}^{M} \beta_m K_m \text{ subject to } \beta_m \ge 0 \tag{2.14}$$

where *m*=1 to *M*. *K* is the representation of the linear combination of kernel matrices. This is called multiple kernel learning where we replace a single kernel with a weighed sum. By linear combination of *m* base kernel matrices with kernel coefficient β_m, where *m*=1 to *M*, we can synthesize kernel matrix *K*.

For training, the dual formulation of MKL with multiple kernels can be defined in matrix form as (rewritten by equation 2.13) :

$$\min_{\sum_m \beta_m K_m} \left(\max_{\alpha,\alpha^{\mathrm{T}}y=0} \alpha^{\mathrm{T}}e - \frac{1}{2}\alpha^{\mathrm{T}}\left(\sum_{\mathrm{m}=1}^{M} \beta_m K_m\right)\alpha \right) \quad s.t.\ 0 \le \alpha_i \le C \tag{2.15}$$

where α_i is the sample coefficient and β_m is the kernel weight. After training, β_m will take values depending on how the corresponding kernel is useful in discriminating. For input x, considering the classification with *N* training samples $\{x_i, y_i \in \pm 1\}_{i=1}^{N}$ and *M* base kernels $\{K_m\}_{m=1}^{M}$, the learned model is of the form:

$$f(x) = \sum_{i=1}^{N} \alpha_i y_i K(x_i, x) + b$$
$$= \sum_{i=1}^{N} \alpha_i y_i \sum_{m=1}^{M} \beta_m K_m(x_i, x) + b. \quad (2.16)$$

The sample coefficient is used for the relation between data and classes where α_i is the weight for the *i*th datum. The kernel coefficient is the representation of classes and features where β_m is the weight for base kernel matrix K_m.

The basic task of MKL is to find the sample coefficient α_i, and corresponding β_m. In other words, the task is to optimize both sample coefficient α_i and kernel coefficient β_m so that the error function can be minimized to achieve better data clustering results. In a conventional method, the kernel coefficient β_m could be obtained from machine-learning approaches, like Support vector machine (SVM). However, the optimization problem is too complex to solve directly; thus, an alternative approach is proposed here using an iterative method to obtain the optimized sample coefficient and kernel coefficient. More specifically, we just solve these two coefficients one at a time while the other is fixed. That is, we optimize α_i by fixing β_m and optimize β_m by fixing α_i. In any odd-numbered iteration, a nearly optimal α_i is obtained by solving a generalized eigenvalue problem; we obtain β_m by solving the relaxation of semidefinite programming (SDP). Next, in the even-numbered iteration, a nearly optimal β_m is obtained by solving a generalized eigenvalue problem; we obtain α_i by solving the relaxation of SDP. In each iteration, we get closer to the optimal solution and then use this solution as the input to the next loop until convergence.

2.5 Counter Propagation Neural Network

Our CPN network is used as a decision-making module for shot selection in our VD subsystem. The CPN is a supervised learning technique, and a real director provided the initial training input data. The most crucial feature of the CPN is its fast response time. The output of CPN is the single shot that has the highest score.

A. Counter propagation neural network introduction

Figure 2.8 shows the architecture of a fully connected CPN network. The network is constructed of five layers: two input layers, two output layers, and one hidden layer. The training data is X. There are n neurons in the X input layer, and the input data to neurons are denoted by x_i, where $i = 1, \dots, n$. Another input layer with m neurons takes the data Y, which is the labeled vector for X, denoted by y_k, where $k = 1, \dots, m$.

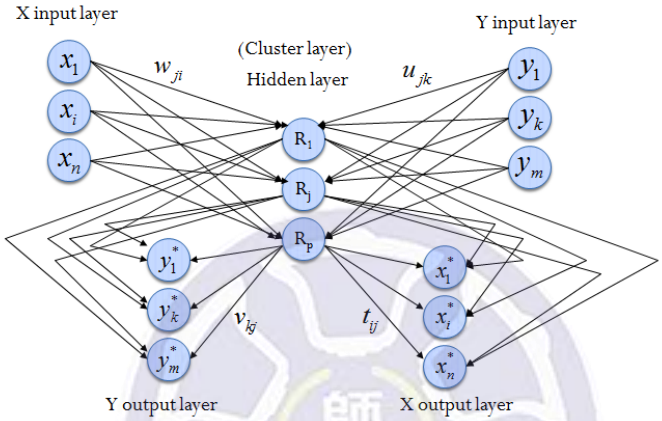


Figure 2.8. Architecture of a fully connected CPN.

In the architecture, the hidden layer contains p neurons; each neuron is denoted R_j, where $j = 1, \dots, p$. This hidden layer is also called the cluster layer. Each neuron in the hidden layer represents a class. The weight vector w_{ji} connected to input neurons x_i and hidden neurons R_j is used to classify i inputs to class j. In the same way, the weight vector u_{jk} connected to input neurons y_k and hidden neurons R_j is used to classify k inputs to class j.

After an input vector is classified, we can obtain the output result calculated from weight vectors v_{kj} and t_{ij}, which directly connect the hidden neurons and the output layer. If we input vector x_i to input layer X, the approximate output is $y_1^*, \dots, y_m^*$; if we input vector y_i to input layer Y, the approximate output is $x_1^*, \dots, x_n^*$.

B. Forward-mapping CPN

One of the features of a fully connected CPN is as follows: if the input to the network is an expected output result, one obtains an input vector corresponding to the result when there is a one-to-one mapping between input vector and output vector. However, there could be different situations in our VD subsystem. For example, assume the director selects the hall view; the cause might be that the speaker is interacting with the audience or that the director wants to use an establishing shot to avoid an emergency. Because both situations could motivate the director to choose the same shot, the mapping function should not be one-to-one. Thus, we simplified the CPN into a forward-only network, and the updated architecture, called a forward-mapping CPN, is shown in Figure 2.9.

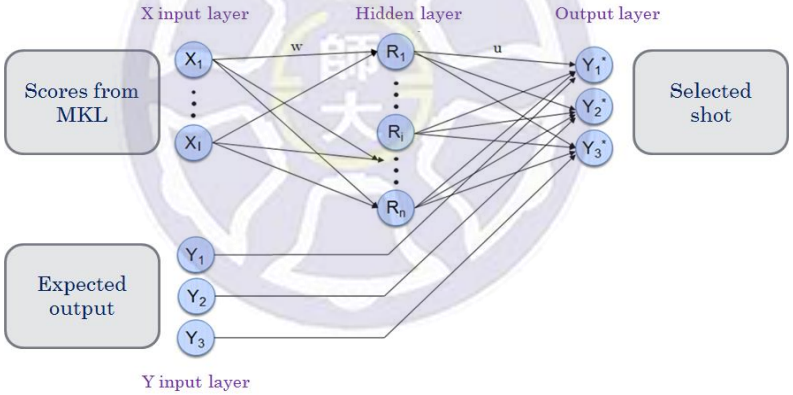


Figure 2.9. Architecture of forward-mapping CPN applied in VD subsystem.

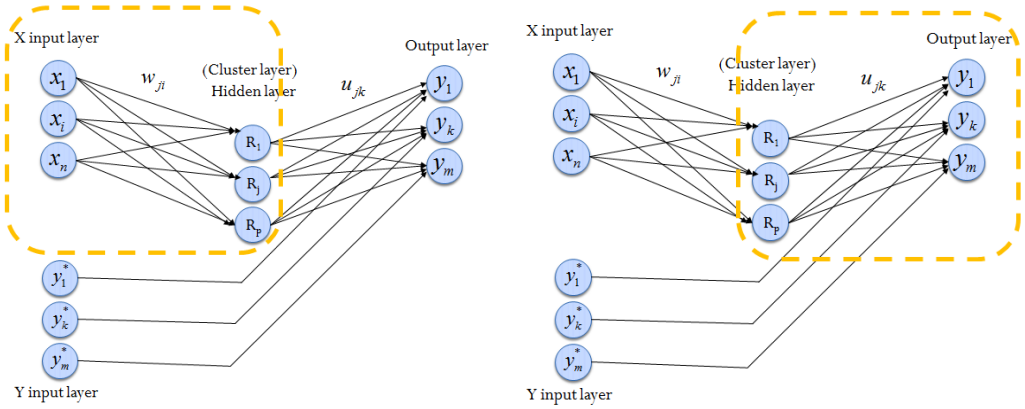


Figure 2.10. Kohonen layer (left) and Grossberg layer (right).

The forward-mapping CPN can be divided into two layers (Figure 2.10). The first layer, called the Kohonen layer, uses a winner-take-all learning algorithm to train the weight vector w_{ji}. The Kohonen layer executes an unsupervised learning algorithm. This layer is often used in classification. Each neuron in the hidden layer can represent a rule. Thus, the entire Kohonen layer can be seen as a rule library. The algorithm is as follows:

Step 1. Assume the score vector for training data is $x_1, \dots, x_n$, where n is the number of training data points.

Step 2. Calculate the likelihood between each class and corresponding weight by:

$$d_j = \sum_{i=1}^{n} \left| x_i(t) - w_{ji}(t) \right| \qquad (2.17)$$

where t is the tth training data point.

Step 3. Choose the minimum d_j, as a winner. Only the weight of the winner is updated.

$$d_{winner} = \min_{j=1,\dots,n} d_j. \qquad (2.18)$$

Step 4. To avoid an overly large difference between weight and input, a threshold Δ is used here. If $d_{winner} < \Delta$, the weight and input are similar; go to Step 5. If $d_{winner} > \Delta$, the weight and input are not similar and a new rule is required; go to Step 6.

Step 5. Update the weight connected to the winner; the update function is:

$$w_{winner,i}(t+1) = w_{winner,i}(t) + \alpha\left[x_{winner,i}(t) - w_{winner,i}(t)\right] \qquad (2.19)$$

where α is the learning rate, and its initial value is $\frac{1}{2}$, but α decreases through iteration to expedite convergence.

Step 6. If no class is found for an input, try to add a new neuron node to the hidden layer (see Figure 2.11). The initial weight for a new node is:

$$w_{p+1}(t+1) = x(t) \tag{2.20}$$

$$u_{p+1,k}(t+1) = y^*(t) \tag{2.21}$$

This uses this input value as the new weight $w_{p+1,i}$ and the expected (labeled) output value as the new weight $v_{p+1,k}$.

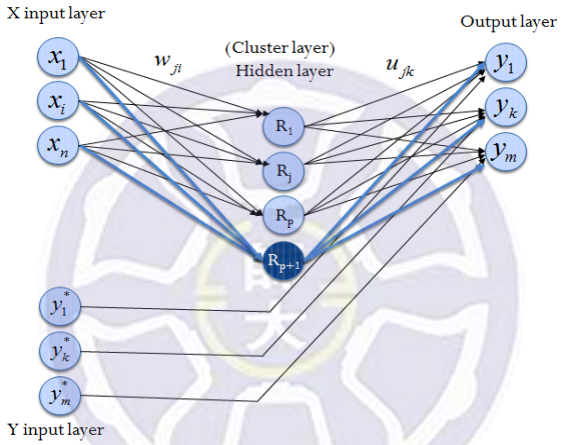


Figure 2.11. Adding a new node to the CPN.

The number of neurons in the hidden layer denotes the number of types that can be classified. The concept is the same as the shot selection of a real director. When real directors select a view, they always must determine the situation that it belongs to (e.g., an audience member is asking some questions, a speaker has a meaningful posture); directors classify the situation to which each selected shot belongs.

The second layer is the Grossberg layer. This layer uses the Grossberg supervised learning algorithm to train the weight vector u_{jk}. The training algorithm of the Grossberg layer resembles the training processes of the Kohonen layer. The Grossberg algorithm is as follows:

Step 1. Assume the score vector for training data is $x_1, \dots, x_n$, where n is the number of training data points, $y_1^*, \dots, y_m^*$ is the expected results, and m is the number of VCs (in this research $m = 3$).

Step 2. Calculate the likelihood between each class and the corresponding weight by:

$$d_j = \sum_{i=1}^{n} \left| x_i(t) - w_{ji}(t) \right| \tag{2.22}$$

where t is the tth training data point.

Step 3. Choose the minimum d_j as the winner. Only the weight of the winner is updated.

$$d_{winner} = \min_{j=1,\dots,n} d_j \tag{2.23}$$

Step 4. Update the weights that connect to the winner; the update function is:

$$w_{winner,i}(t+1) = w_{winner,i}(t) + \alpha \left[x_{winner,i}(t) - w_{winner,i}(t) \right] \tag{2.24}$$

$$u_{winner,k}(t+1) = u_{winner,k}(t) + \beta \left[y^*_{winner,k}(t) - u_{winner,k}(t) \right] \tag{2.25}$$

α is the learning rate of the Kohonen layer, and the value is a constant at this stage. α is the last value in the Kohonen layer after convergence. β is the learning rate of the Grossberg layer, and the initial value is also a constant.

Chapter 3
Smart Lecture Recording System

The overall organization of the smart lecture recording (SLR) system is shown in Figure 3.1. There are three principal components constituting the SLR system: virtual cameraman (VC), virtual director (VD), and manual control (MC). The VC component is further divided into three sub-components: speaker cameraman (SC), audience cameraman (AC), and hall cameraman (HC). This division is inspired by a professional lecture recording team that in general possesses at least three cameramen for performing the separate duties of shooting the speaker, the listeners, and the panoramic scene. In the ensuing sections, we discuss the kinds of lecture halls considered in this thesis in Section 3.1; the architecture of the SLR system is described in Section 3.2; the workflow of the system is finally addressed in Section 3.3.

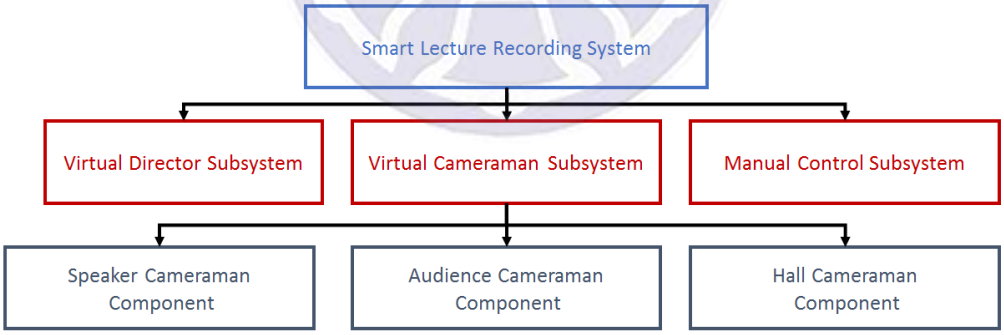


Figure 3.1. The organization of the SLR system.

3.1 Lecture halls under consideration

The lecture halls under our consideration range from ordinary classrooms, lecture theaters, to grand oration halls. In addition to extent, various kinds of lecture halls are characterized by distinct arrangements of auditoriums. Figure 3.2 shows several halls with different disposals of seats, such as (a) tiered, (b) level, and (c) ambient

auditoriums. Tiered auditoriums are often for presentations held by academic institutes and organizations. Level auditoriums are often for exhibitions of new products held by business firms or deliveries of new songs by music companies. As for ambient auditoriums, they are primarily for formal reports or speeches in congresses and parliaments. Our SLR system will be able to work in the sites mentioned above.

Figure 3.2. The kinds of lecture halls considered in this study (a) tiered (b) level (c) ambient auditoriums.

3.2 System Architecture

To illustrate the architecture of the SLR system, let us look at the layout depicted in Figure 3.3, which shows a deployment of hardware devices of the SLR system in a lecture room. There is a screen mounted on the front wall of the room for displaying lecture materials (e.g., power-points and videos) transmitted from a computer controlled by the speaker. The SC component of the SLR system consisting of a Kinect sensor and a PTZ camera sits in front of the speaker and points to the speaker. The Kinect sensor serves as a photographer and the PTZ camera plays as his/her imaging device. Once the Kinect sensor perceives an object, the Kinect sensor directs the PTZ camera toward the object for identification and tracking. The AC component comprising two PTZ cameras stood in the front of the room faces toward audience seats. Similarly, the top PTZ camera serves as a photographer and the bottom PTZ camera plays as his/her imaging device. Finally, the HC component containing only one PTZ camera is located at the rear of the lecture room. The major purpose of this component is to provide panoramic shots of the room as interesting episodes for weaving in the

lecture video.

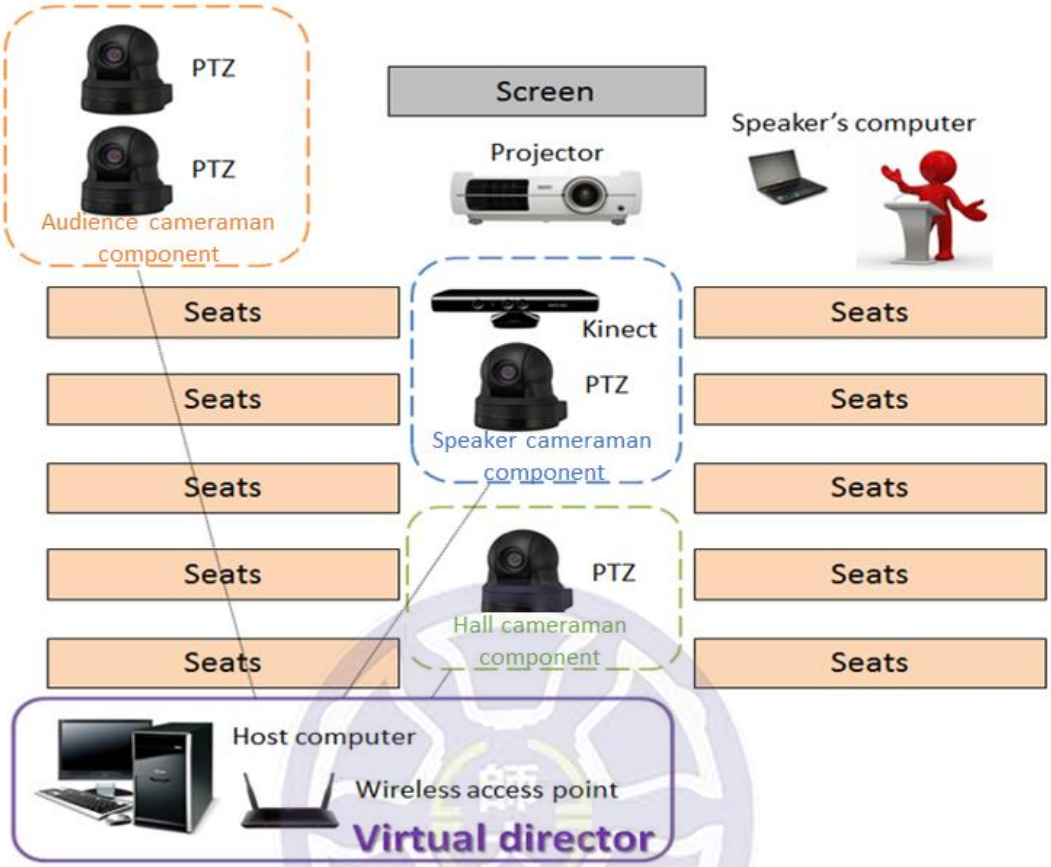


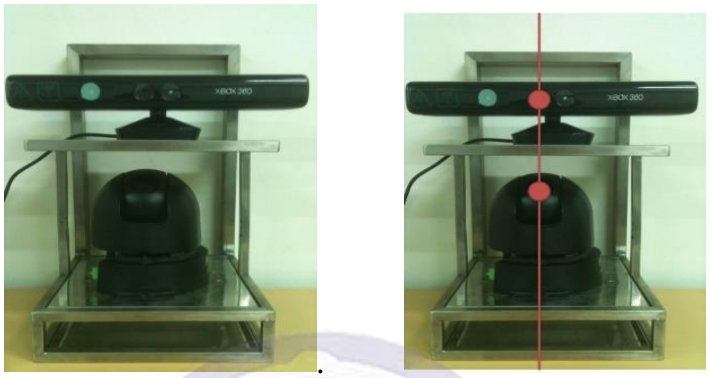
Figure 3.3. A deployment of hardware devices of the SLR system in a lecture room.

All the aforementioned three components of the SLR system and the speaker computer are connected through wire or wireless communications to a host computer situated anywhere in the room. Note that an SLR system may contain multiple SC, AC, and HC components for working in a large lecture hall. For such a system, the most important issue of concern may be the coordination of components during lecture recording. We leave this issue to the future work.

3.2.1 The SC component

The SC component consists of a Kinect sensor and a PTZ camera, whose specifications are to be addressed later. Figure 3.4(a) shows the configuration of the SC component, where the Kinect sensor is situated on the top of the PTZ camera. Initially, the PTZ camera is in the home position. It's lens is vertically aligned the lens of the

depth camera of the Kinect sensor (see Figure 3.4(b)). Once the Kinect sensor detects an object in the image plane of the depth camera, the Kinect sensor immediately computes the 3D orientation of the object and accordingly guides the PTZ camera toward the object to see whether it is the target of interest or not.



(a) (b)

Figure 3.4. The configuration of the SC component (a) The picture of SC component (b) The red points mark the lens positions of the depth camera and the PTZ camera, respectively.

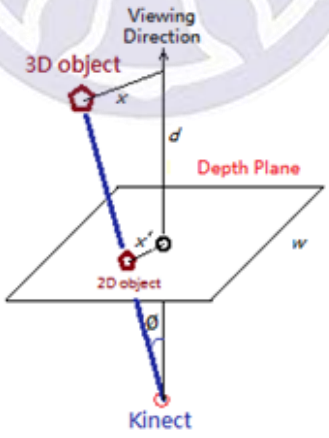


Figure 3.5. The Kinect perceives an object, i.e., an object is present in the image plane of the depth camera of the Kinect.

See Figure 3.5; let $\emptyset$ be the viewing angle of a 3D object. The angle is known by the Kinect sensor through the projection of the object onto the image plane of the depth camera of the Kinect sensor. Based on this angle, the horizontal pan angle $\emptyset_{hor}$ and

the vertical tilt angle $\emptyset_{vert}$ of the PTZ camera can be determined. Consider $\emptyset_{hor}$. Let f be the focal length of the PTZ camera and w' be the half width of the depth plane. Look at Figure 3.5; x' is the distance between the 2D object and the center of the depth plane, d is the distance from the Kinect sensor to the world plane, which is the plane parallel to the depth plane and passing through the 3D object. Refer to Figure 3.6; θ_{hor} is the horizontal viewing angle of the depth camera, w and x are the width of the world plane and the distance between the center of the world plane and the 3D object, respectively. Based on the similar triangle property $x = \frac{x'}{w'}w = \frac{x'}{w'}\left(d \tan\frac{\theta_{hor}}{2}\right)$. The horizontal pan angle $\emptyset_{hor}$ of the PTZ camera is calculated according to $\emptyset_{hor} = \tan^{-1}\left(\frac{x}{d}\right)$.

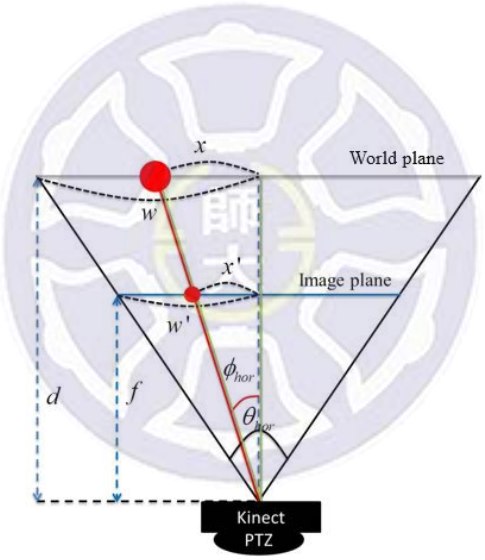


Figure 3.6. The horizontal pan angle $\emptyset_{hor}$ of the PTZ camera.

Consider the vertical tilt angle $\emptyset_{hor}$ of the PTZ camera. See Figure 3.7; suppose l is the displacement between the Kinect and the PTZ camera. h' and y' are the height of the image plane and the vertical distance between the target image and the center of the image plane, respectively. d is the distance from the camera to the world plane. In the world plane, h and y are the width of the world plane and the vertical distance between the center and the target, respectively. θ_{vert} is the vertical view angle

of the depth camera. $\emptyset_{vert}$ is the tilt angle for the PTZ camera. First, use similar triangle properties to calculate y : $y = \frac{y\prime}{h\prime}h = \frac{y\prime}{h\prime}\left(d\ tan\frac{\theta_{vert}}{2}\right)$. Then, calculate $\emptyset_{vert}$: $\emptyset_{vert} = \tan^{-1}\left(\frac{y+l}{d}\right)$. The aforementioned parameters are known except d, $\emptyset_{hor}$, and $\emptyset_{vert}$. Therefore, PTZ camera action requires only a simple computation process to generate control signals.

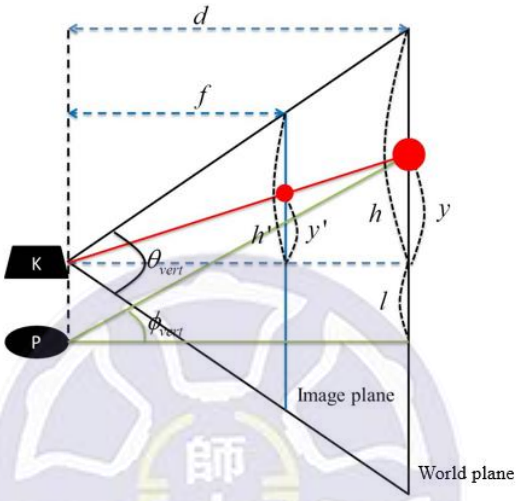


Figure 3.7. The vertical tilt angle $\emptyset_{hor}$ of the PTZ camera.

3.2.2 The AC component

The AC component consists of two PTZ cameras. Figure 3.8 shows the configuration of the AC component. Unlike the SC component in which the Kinect sensor serves as a photographer, a moving PTZ camera of the SC component is serving as a photographer instead. This is because audiences typically have much wider extent than the speaker even though the speaker can move around, the field of view of a moving PTZ camera will be able to cover the entire range of audiences. Similar to the PTZ camera of the SC component, the bottom PTZ camera of the AC component serves as the imaging device of the cameraman.

Figure 3.8. The configuration of the AC component.

Similar to the SC component, once the top PTZ camera detects an object, the 3D position of the object is determined by the SC component. The position of the object then guide the bottom PTZ camera toward the object.

3.2.3 The HC component

The HC component containing one PTZ camera is located at the rear of the lecture room. In addition to taking panoramic shots of the room, including the screen, podium, speaker, and audience, another important task for the HC component is to detect the interactions between the speaker and audience. Those interactions will provide interesting episodes for weaving in the lecture video.

3.2.4 Kinect sensor

A Kinect sensor (see Figure 3.9) is included in the SC component, which is an active depth sensor produced by Microsoft and typically used for games. With the Kinect sensor, users simply use gestures to direct a general operating system interface. The Kinect sensor that photographs a person also captures a virtual skeleton abstracted from visual information about that person (Figure 3.10). This feature allows users to play interactive, controller-free, Kinect-based games by moving their bodies.

Figure 3.9. Kinect (from: Google pictures)

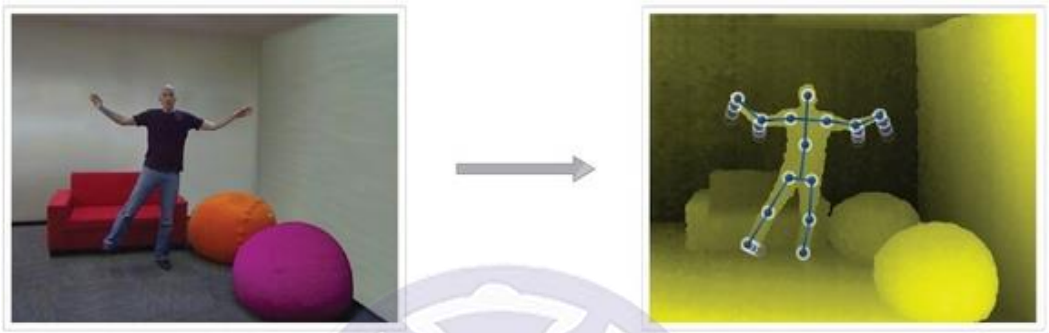


Figure 3.10. Kinect virtual skeleton (from: Primesense)

The Kinect sensor provides three pivotal types of information: color images, depth images (see Figure 3.11), and sound. Color images are obtained by the RGB camera in the middle of the Kinect, while depth images are produced by the infrared transmitter and infrared CMOS sensor at the left and right sides. The detailed specifications are as follows:

-Depth-sensing and skeleton detection preferred distance: 1.2 to 3.6 meters

-FOV: 57 degrees horizontal, vertical 43 degrees

-Motor rotation angle: Up and down 28 degrees

-Frames per second (FPS): 30 per second

-Depth resolution: QVGA (320 x 240)

-Color resolution: VGA (640 x 480)

(a) (b)

Figure 3.11. Information from a Kinect sensor (a) color image (b) depth image.

In addition, in 2011 Microsoft released the Kinect SDK for its own operating system software (including Windows 7). It allows users around the world to research and develop new Kinect applications (e.g., gestures can be used to control robots, operate slideshows, and select items). This study integrates the virtual skeletons recorded by Kinect sensors with a custom hand gesture library to identify speakers' hand gestures.

3.2.5 PTZ camera

All the three components of the SLR system contain PTZ cameras (see Figure 3.12). A PTZ camera is a camera that is capable of remote directional and zoom control. PTZ is an abbreviation for pan, tilt, and zoom, and PTZ cameras can execute all three of those motions. PTZ cameras are commonly used in applications such as surveillance, videoconferencing, live production, lecture capture, and distance learning.

Figure 3.12. PTZ cameras.

The specifications of the PTZ cameras in the present study are listed as follows：

-FOV: Horizontal 4.4 to 51.6 degrees

-Zoom: 12× optical and a multiple of 4 bits (a total of 48 times)

-Horizontal rotation angle: −170 to 170 degrees

-Vertical tilt angle: −90 to 90 degrees

-Image compression format: Motion JPEG, MPEG-4 Part 2 (ISO / IEC 14496-2)

-Resolution: 704x480 to 176x120

Unlike stationary cameras that can only capture a fixed area, PTZ cameras can freely rotate to shoot various areas. In addition, PTZ cameras can be connected directly to a wired or wireless network. A network connection can transmit captured images in real time. Some PTZ cameras conveniently include an embedded web server, thus obviating additional drivers and enabling direct control of the camera and direct access to images.

PTZ cameras are mostly used in surveillance systems, because they can capture images from different angles. In this study, speakers tend to move from time to time during speeches, and the videos may require views of the background or whiteboard as well. Therefore, the fact that the PTZ camera can be rotated freely makes it highly suitable for the present study.

3.3 System Workflow

Figure 3.13 shows a workflow of the SLR system. When we start the system, the two computers of the system: the speaker’s computer and the host computer, are powered on and initialized. Afterwards, the hardware devices of the system, including projectors, microphones, Kinect sensors, PTZ cameras, and Ethernet switches, are actuated and are connected to one another through either wire or wireless communications. The system is now ready to operate.

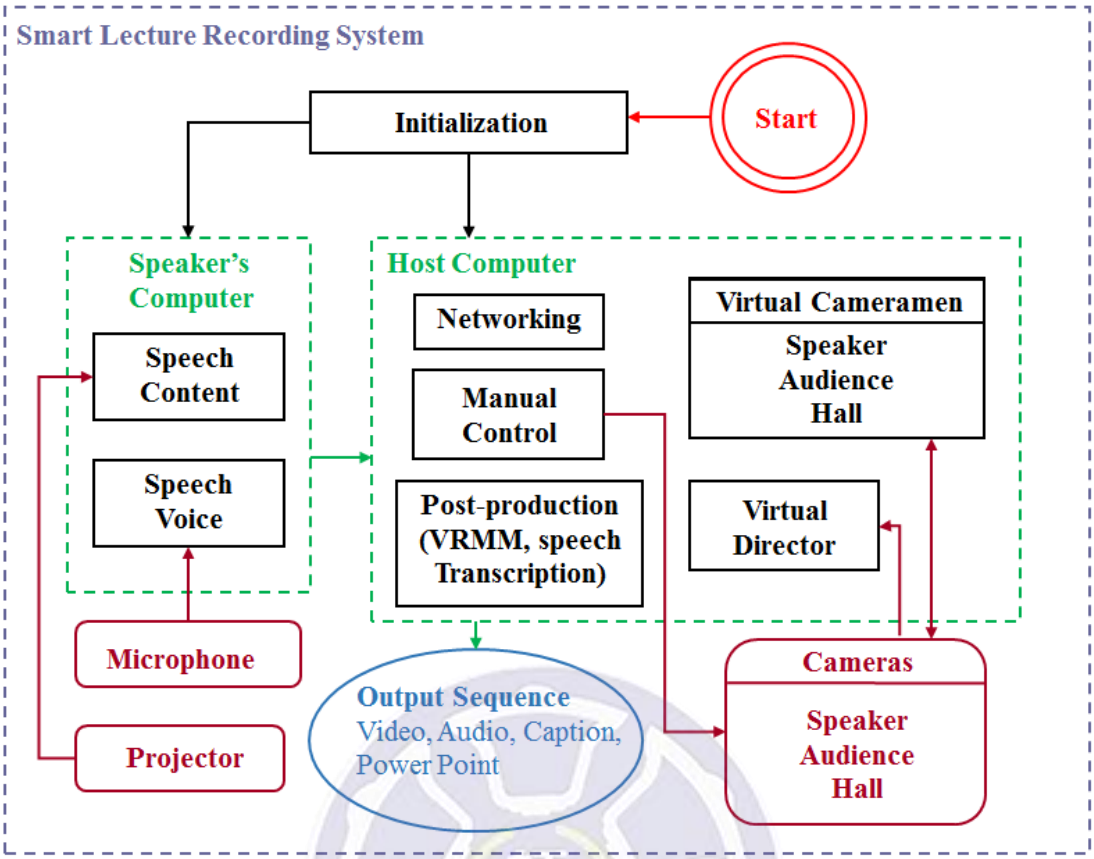


Figure 3.13. A workflow of the SLR system.

Before the speech begins, the user has to click the "start" button on the screen of a manual control interface (see Figure 3.13). The videos taken by the SC, AC, and HC components are then delivered to the host computer, in which videos are analyzed by the VC subsystem. The VC subsystem determine the subsequent actions of the PTZ cameras of the three components based on both the results of video analysis and an embedded set of photographical rules. Repeating the above process of taking videos, analyzing videos, and determining actions of cameras, the VC subsystem achieves the automatic operations of view finding, detection and tracking of targets. In the meantime, the VD subsystem embedded in the host computer chooses the optimal video from those delivered by the SC, AC, and HC components based on the results of video analysis accomplished by the VC subsystem as well as a pre-learned set of election rules. The selected video together with the voice, content and transcription of the speech

is recorded or broadcasted. More specifically, the output sequence will include videos, audios, captions, and power points.

There are three subsystems involved in the SLR system: virtual cameramen (VC), virtual director (VD), and manual control (MC). They are discussed in the ensuing sections, separately.

3.3.1 VC Subsystem

Recall that there are three components involved in the VC subsystem: the SC, AC, and HC components. Consider the SC component. There are two major tasks for the SC component: speaker posture detection and automatic control of the camera. First, the SC component locates the speaker using the AdaBoost algorithm [13], and then the speaker is tracked by the mean-shift algorithm. During tracking, the postures of the speaker are recognized from depth sensor data and a set of prebuilt gesture models. Our system uses a Kinect sensor to detect speakers' gestures for posture recognition. The SC component then determines the control signals of the PTZ camera according to a collection of predefined automatic camera-control rules. The output of the VC subsystem is an image sequence acquired by its PTZ camera.

The audience members that raise their hands during the speech may ask questions and interact with speakers, so the AC component shoots the audience to detect relevant events and to catches any audience member who might be asking a question. The AC component executes a crowd motion detection algorithm that applies optical-flow estimation and the STA model. Then the AC component controls its PTZ camera to shoot any event indicated by the detected results.

Because the hall shot is defined as a safe shot, the HC component is programmed to operate in passive mode, which means that the HC component only performs a camera action when the VD subsystem gives a visual instruction. Otherwise, the HC component simply takes a wide shot that includes both the speaker and the audience.

The three videos transmitted from the SC, AC, and HC components are analyzed by the VC subsystem. The analysis includes contextual information regarding the scene. Based on the results and the rules of photography, the VC subsystem generates control signals for the PTZ cameras and rearranges the shot composition. In comparison to the traditional scenario, in which cameramen only transfer the video signals to the director who judges them without assistance and selects the shots in the real world, one advantage of the proposed SLR system is that virtual cameramen provide additional helpful environmental information (e.g., posture of speaker, audience events) to the VD subsystem.

3.3.2 VD subsystem

The objective of the VD subsystem is to imitate a professional director. Figure 3.14 shows a flowchart for the VD subsystem. The VD subsystem collects clues from the videos taken by the SC, AC, and HC components of the VC subsystem. Figure 3.15 shows exemplar videos taken by the three components, respectively. The VD subsystem must determine the representative shot from three input videos. After receiving the video signals and environmental information, the VD subsystem performs the shot selection and visual instruction. In the shot selection stage, the VD subsystem decides which shot is the representative shot through content analysis. In visual instruction stage, the VD subsystem makes control decisions and generates control signals for the VC subsystem by content analysis results and event detection results. The output may combine the representative shot with titles, voiceovers, and slides, depending on the requirements of the live broadcast or permanent recording (see Figure 3.16).

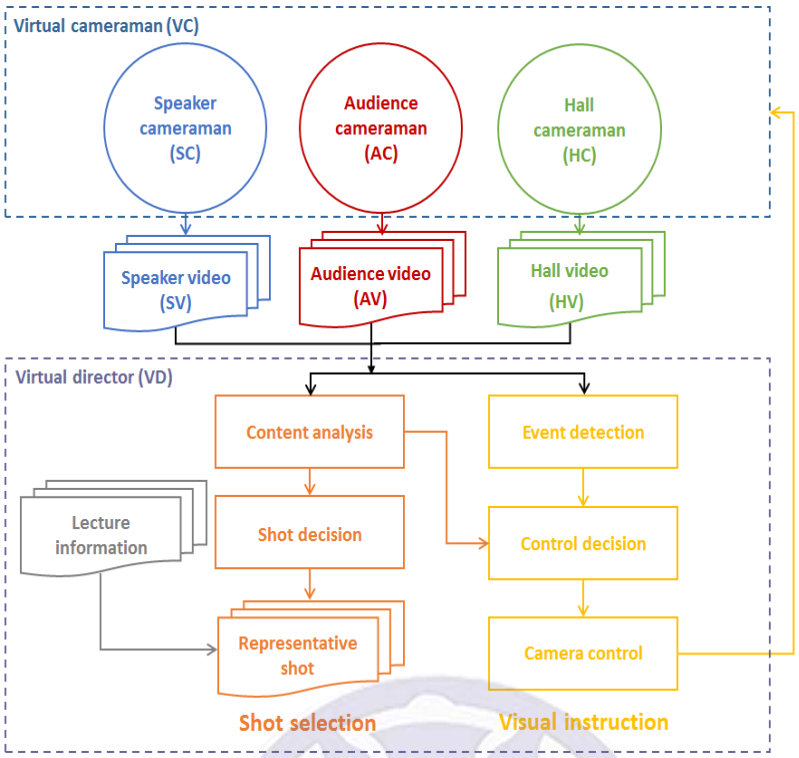


Figure 3.14. A flowchart of the VD subsystem.

(a) (b) (c)

Figure 3.15. Videos of the VC subsystem (a) SC (b) AC (c) HC.

Figure 3.16. An output of the VD subsystem.

3.3.3 Manual control subsystem

In our system, we need control buttons to enable a real director to override the automatic decisions of shot selection and camera control. Figure 3.17 shows a control panel of the user interface. In the first row of the panel, the left-most three images are transmitted from the three components of the VC subsystem, respectively. There is a small mark (a white arrow centered at a blue circle enclosed by a white square) in each of the second and the third images. The first image without mark means that it is currently selected by the VD subsystem for record or broadcast. An enlargement of this image is also displayed around the lower center of the panel.

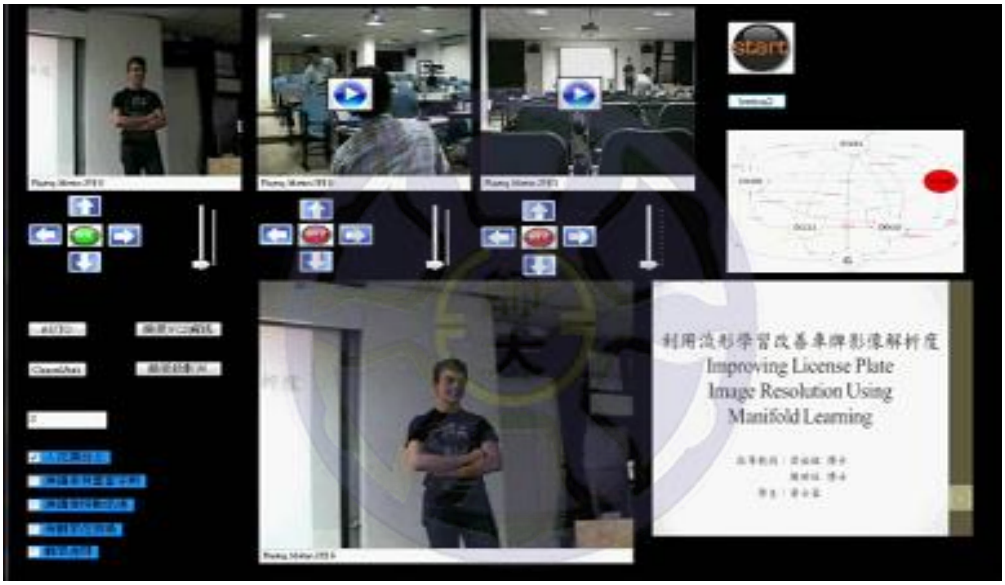


Figure 3.17. User interface of the manual control.

The operator can manually select a different image by clicking the mark on the image. The mark on the selected image disappears and meanwhile a mark appears in the previously selected image. Note also the graphs below the images. Each graph (see Figure 3.18) includes two parts. The left part is composed of four directional arrows with a central button. The button may be in the state of green “on” (or red “off”) if the corresponding image is selected (or not selected). The arrows around the button in the state of green “on” can be used to move the associated PTZ camera up, down, left, or right. The slide pole on the right part of the graph control zoom in/out of the PTZ camera.

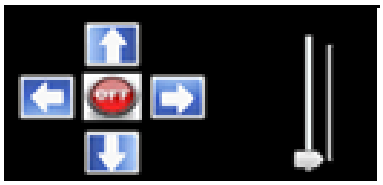


Figure 3.18. Manual control of a PTZ camera.

There is a “Start” button on the upper right corner of the panel, which can turn on/off the entire SLR system. The graph below the “Start” button shows in terms of FSA (finite state automaton) the internal state of the SLR system. The descriptions of the state of the system are displayed within the window around the lower left corner of the panel. The content of the speech is shown around the lower right corner of the panel.

3.3.4 System deadlock prevention

To avoid system deadlock and increase system efficiency, the design of the entire SLR system was analyzed in terms of finite automata theory. Lecture situations, such as the speaker pointing and the audience asking a question, were represented as finite state machines (FSMs). FSMs represented the transitions between system states and output camera shots. The answers to the following questions depend on the VC subsystems: 1. Is the speaker in the shot? 2. Is the speaker performing a special action? 3. Is the speaker moving fast? 4. Is the audience in the shot? 5. Are any audience members asking questions?

The SC and the AC were designed to return the aforementioned information as signal input to the VD. We created tables to describe these states, in which zero represented the nonoccurrence of an event. Example tables are shown as Table 3.1 and Table 3.2. For example, the symbol s100 means the speaker is in the shot, is not performing a special action, and is not moving fast.

Table 3.1 Speaker state table.

In the shot	Gesture	Moving	Symbol

0	0	0	s000
0	0	1	s001
0	1	0	s010
0	1	1	s011
1	0	0	s100
1	0	1	s101
1	1	0	s110
1	1	1	s111

Table 3.2. Audience state table.

In the shot	Ask	Symbol
0	0	a00
0	1	a01
1	0	a10
1	1	a11

Some symbols were omitted because they represented impossible situations (e.g., s010, the speaker is not in the shot but the SC detects the speaker performing a special action). We combined the symbols into conditions, as shown in Table 3.3.

Table 3.3. Condition-state table.

Condition	state code
c1	s000a00
c2	s000a10
c3	s000a11
c4	s100a00
c5	s100a10
c6	s100a11
c7	s101a00
c8	s101a10

Condition	state code
c9	s101a11
c10	s110a00
c11	s110a10
c12	s110a11
c13	s111a00
c14	s111a10
c15	s111a11

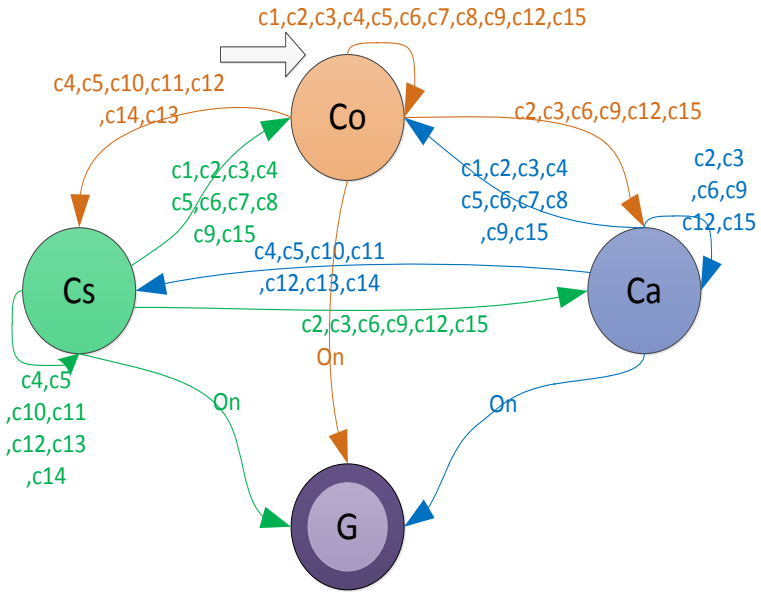


Figure 3.19. FSM of the VD subsystem.

The output of each camera shot was represented as a state and each input signal was represented as an input symbol of the FSM. The FSM represented the VD as shown in Fig 3.19, where Co is the initial state and G is the final state.

Table 3.4 presents the states in which camera shots should be selected.

Table 3.4. Output state table.

Hall	Co
Speaker	Cs
Audience	Ca

However, the FSM in Fig 3.20 is nondeterministic, thus we converted it into a deterministic finite automaton (DFA) using the method we described in Chapter 2 and simplified it to remove unnecessary states. The DFA representation is as follows:

$M=(\Sigma, S, q_0, \delta, F)$ where

$\Sigma=\{ c1, c2, c3, c4, c5, c6, c7, c8, c9, c10, c11, c12, c13, c14, c15, On \}$

$S=\{D100, D101, D110, D010, D111, G\}$

$q_0=\{D100\}$

$F=\{G\}$

Figure 3.21 shows the DFA. Table 3.5 shows which shots can be chosen in each state. For example, D101 means the director can select the hall camera or the audience camera. When the current state shows that multiple shots can be selected, the VD must perform a content analysis to find the representative shot.

Table 3.5. New state table after DFA.

Hall	Speaker	Audience	New state
0	1	0	D010
1	0	0	D100
1	0	1	D101
1	1	0	D110
1	1	1	D111

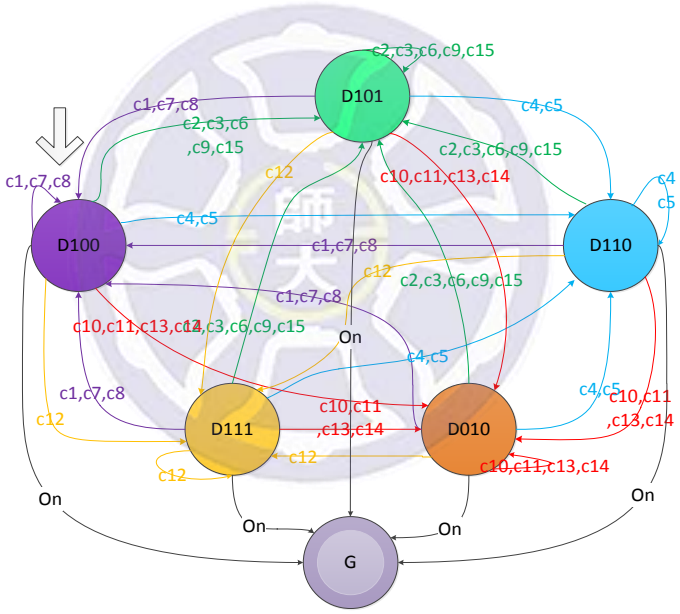


Figure 3.20. DFA of the VD subsystem.

Chapter 4 Virtual Cameramen

This chapter addresses the implementation details of the virtual cameraman (VC). There are three subsystems constituting the VC system: speaker cameraman (SC), audience cameraman (AC), and hall cameraman (HC). The SC subsystem is discussed in Section 4.1, the AC subsystem is detailed in Section 4.2, and the HC subsystem is addressed in Section 4.3. Finally, several rules regarding camera action are presented in Section 4.4.

4.1 Speaker Cameraman

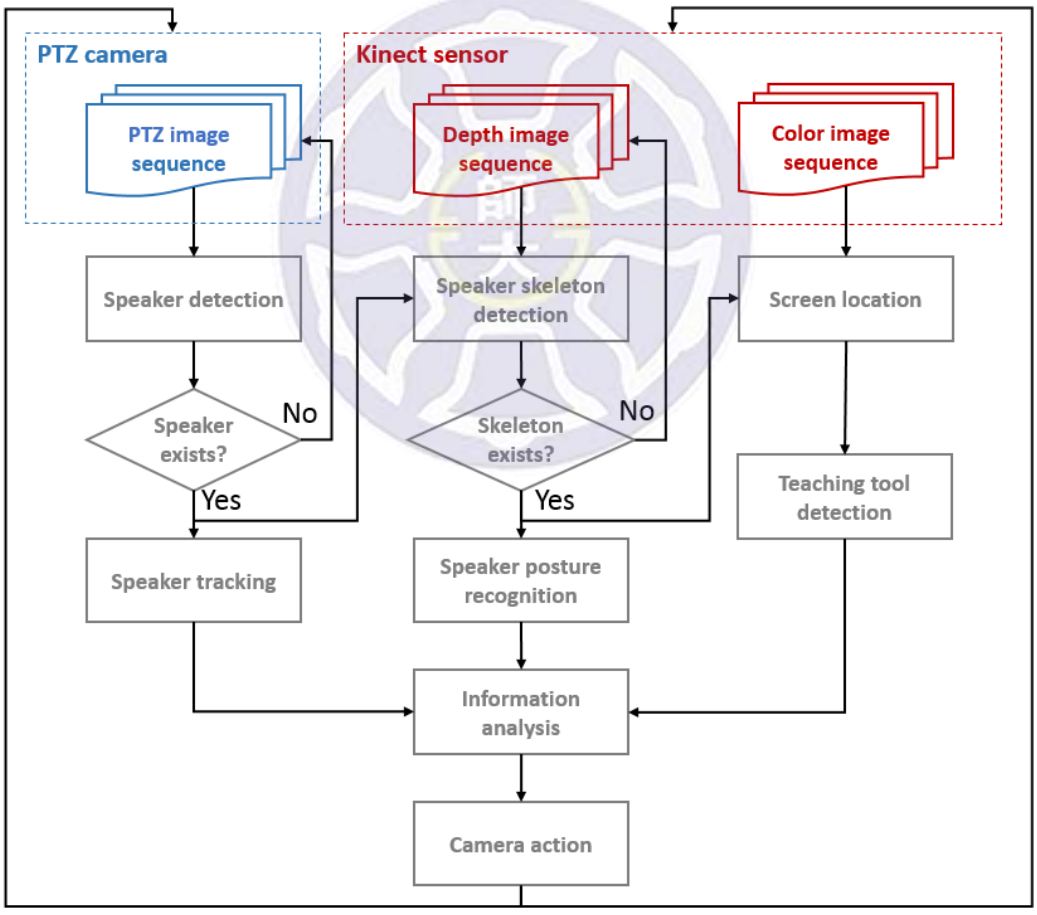


Figure 4.1. Flowchart of SC subsystem.

The flowchart of the SC subsystem is shown in Figure 4.1. To begin, the SC subsystem attempts to detect the speaker in the color image taken by its PTZ camera.

The detection continues until the speaker is found. The virtual skeleton of the speaker is next localized in the depth images given by the depth camera of the SC subsystem. Once the speaker virtual skeleton has been localized, the posture of the speaker can be recognized from the joints of the skeleton. When processing a color image to detect a speaker, the SC subsystem also looks for teaching tools, such as a projection screen and any visual aids used by the speaker. The system analyzes the detection results regarding the speaker, his or her posture and teaching tools in the context of a set of predefined rules of camera action; from that analysis, the system determines the subsequent actions of the color and depth cameras. In the following, we discuss speaker detection and tracking.

4.1.1 Speaker detection and tracking

This section consists of two parts: speaker detection and speaker tracking. Speaker detection involves detecting the presence of the speaker in the picture. Speaker tracking involves detecting that the speaker is moving, and quickly getting the new speaker position in the shot.

A Speaker detection

To find a speaker in an image from a PTZ camera, the system scans for one part of the speaker. The speaker's face is generally selected, because it is simple and uniform relative to the body, which is covered by clothes. Therefore, the AdaBoost algorithm proposed by Viola and Jones [13] is used in our system. We use Haar-like features to detect a speaker's face. A Haar-like feature is characterized by two or more adjacent black and white rectangles (see Figure 4.2). For the different parts of an image of a human face, different Haar-like features can be used with different effects (e.g., the sum of gray values for a nose may be larger than the sum of gray values for a cheek). Much time can be saved by using the integral image to accelerate the computation of Haar-

like features.

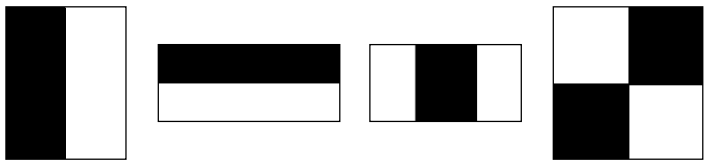


Figure 4.2. Haar-like features.

The learning procedure of the AdaBoost algorithm involves training a large number of weak classifiers and combining them into a strong classifier. This is called multilayer classification (see Figure 4.3).

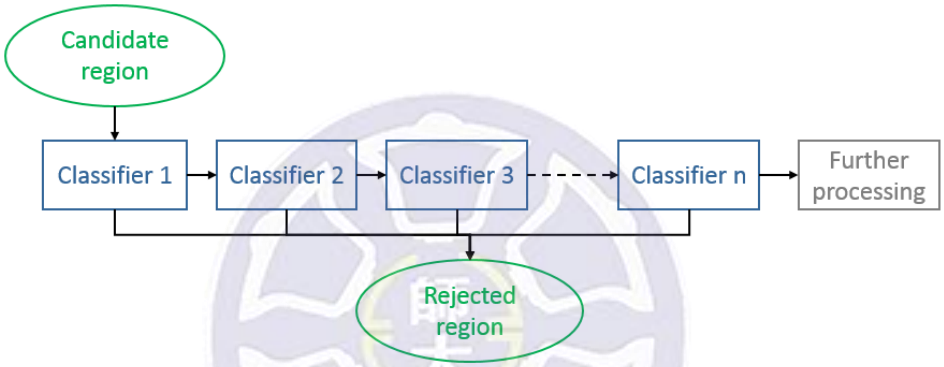


Figure 4.3. Multilayer classification. The blue rectangles represent the weak classifiers.

Given a sample set:

$S = \{(x_1, y_1), (x_2, y_2), \dots, (x_m, y_m), (x_{m+1}, y_{m+1}), (x_{m+2}, y_{m+2}), \dots, (x_{m+l}, y_{m+l})\}$,

where $x_i \in X$, $y_i \in \{1,0\}$（if $y = 1$ then a term has a positive label, if $y = 0$ then a term has a negative label）, m is the number of positive labels, l is the number of negative labels, and $n = m + l$ is the number of samples. The set of weak classifiers is defined as $H = \{h_1, \dots, h_k\}$, where k is the number of weak classifiers. The details of the AdaBoost algorithm are described as follows:

Step1. Initialize the weight of each sample: $D_1(i) = \left\{\frac{1}{2m}, \frac{1}{2m}, \dots, \frac{1}{2m}, \frac{1}{2l}, \frac{1}{2l}, \dots, \frac{1}{2l}\right\}$, and choose T as the target number of the weak classifier; note that T is also the iteration count of the algorithm.

Step2. For $t = 1, \ldots, T$:

(1) Normalize the weight:

$$D'_t(i) = \frac{D_t(i)}{\sum_{j=1}^{n} D_t(j)}. \quad (4.1)$$

(2) Find a weak classifier h_t that minimizes the error ε_t:

$$\varepsilon_t = \sum_{i=1}^{n} D'_t(i)|h_i(x_i) - y_i|, \text{ where } h_t \in H. \quad (4.2)$$

(3) Update the weight:

$$D'_{t+1}(i) = D'_t(i) \times \begin{cases} \alpha_t & , \text{if } h_t(x_i) = y_i \\ 1 & , \text{if } h_t(x_i) \neq y_i \end{cases}, \text{ where } \alpha_t = \frac{\varepsilon_t}{1-\varepsilon_t} \quad (4.3)$$

Step3. Combine T weak classifiers into one strong classifier:

$$H(x) = \begin{cases} 1 & , \text{if } \sum_{t=1}^{T} \beta_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \beta_t \\ 0 & , \text{otherwise} \end{cases}, \text{ where } \beta_t = \log \frac{1}{\alpha_t}. \quad (4.4)$$

In the input image, a thorough search of different locations and subset sizes is performed to find the ROI that will be classified. After that search procedure, those ROI images are entered as inputs to the AdaBoost strong classifier that can determine whether a human face is visible in the image. Finally, the locations of faces are marked on the input image. Figure 4.4(a) and Figure 4.4(b) show the outdoor and indoor face results, respectively.

(a) (b)

Figure 4.4. Face detection results (a)outdoor (b)indoor.

B. Speaker tracking

During a speech, the speaker moves from time to time, so the location of speaker constantly changes. Therefore, after finding the area of the speaker's face, the SC

subsystem must track the current position of speaker such that the PTZ camera can follow and shoot the speaker nonstop. The mean-shift tracking algorithm [14] is used here because of its fast performance. The virtual skeleton information from the depth camera and the hue histogram (see Figure 4.5) of the image from the PTZ camera are selected to represent the template and candidate images.

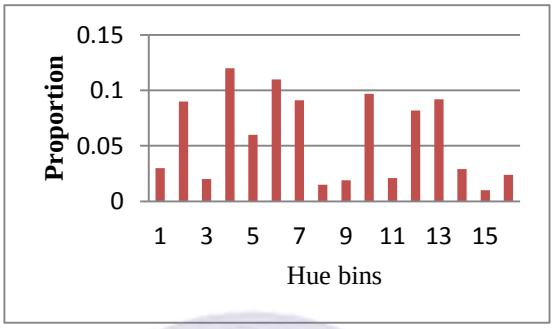


Figure 4.5. Distribution of the hue histogram.

The mean-shift tracking algorithm aims uses the similarity between past tracking results and candidate locations to find the current location. The similarity is defined by the Bhattacharyya coefficient:

$\rho(y) \equiv \rho[p(y), q] = \sum_{u=1,\ldots,m} \sqrt{p_u(y)q}$ where q is hue distribution density function of the template image, $p(y)$ is hue distribution density function of the candidate image, u is number of bins, $\rho(y) = [0,1]$ and $\rho(y) \in R$. If $\rho(y)$ is close to 1, it indicates that $p(y)$ is similar to q; otherwise, if $\rho(y)$ is close to 0, it indicates that $p(y)$ is not similar to q.

First, assume $\{x_i^*\}_{i=1,\ldots,n}$ represents each pixel in a template image, and assume $b: R^2 \rightarrow \{1, \ldots, m\}$ where m is the number index of the hue and $b(x_i^*)$ is the hue index of pixel x_i^*. The hue distribution density function of the template image is formulated as:

$$q = \{q_u\}_{u=1,\ldots,m} \tag{4.5}$$

where $q_u = \frac{1}{C}\sum_{i=1}^{n} k(\|x_i^*\|^2)\delta[b(x_i^*) - u]$, C is a normalization constant (let

$\sum_{u=1}^{m} q_u = 1$), k is an Epanechnikov kernel function, and δ is a Kronecker delta function.

Then, assume $\{x_i^*\}_{i=1,\dots,n}$ represents the pixel set of the candidate image, $y \in \{x_i\}_{i=1,..,n}$ represents the center of the candidate image, and then the candidate hue distribution density function can be formulated as:

$$p(y) = \{p_u(y)\}_{u=1,..,m} \tag{4.6}$$

where $p_u(y) = \frac{1}{C}\sum_{i=1}^{n} k\left(\left\|\frac{x_i - y}{h}\right\|^2\right)\delta[b(x_i) - u]$, C is a normalization constant (let $\sum_{u=1}^{m} p_u(y) = 1$), and h is the radius of the kernel function.

The steps of the mean-shift tracking algorithm based on [15] are as follows:

Step 1. Construct the hue distribution density function q of the template image.

Step 2. Initialize the center y_0:

For the first iteration, y_0 is set to the center of the face area (Figure 4.6(a)) detected by the AdaBoost algorithm. In later iterations, y_0 is set to the center of the tracked area (Figure 4.6(b)) provided by the mean-shift tracking algorithm.

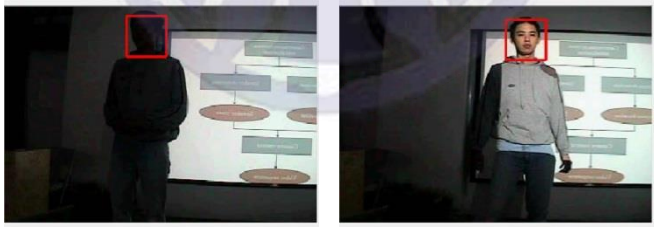


Figure 4.6. Detected face areas (a) the area detected by AdaBoost (b) the area detected by the mean-shift tracking algorithm.

Step 3. Get the next image as a candidate image, and find the corresponding hue distribution density function (y_0).

Step 4. Calculate the Bhattacharyya coefficient between the template and candidate images:

$$\rho[p(y_0), q] = \sum_{u=1}^{m} \sqrt{p_u(y_0) q_u} \tag{4.7}$$

Step 5. Compute the weight $\{w_i\}_{i=1,\dots,n_h}$:

$$w_i = \sum_{u=1}^{m} \delta[b(x_i) - u] \sqrt{\frac{q_u}{p_u(y_0)}} \quad (4.8)$$

Step 6. Compute the new center y_1:

$$y_1 = \frac{\sum_{i=1}^{n} x_i k\left(\left\|\frac{x_i - y_0}{h}\right\|^2\right) w_i}{\sum_{i=1}^{n} k\left(\left\|\frac{x_i - y_0}{h}\right\|^2\right) w_i} \quad (4.9)$$

Step 7. Calculate the Bhattacharyya coefficient between the new template image formed by the new center y_1 and the candidate images:

$$\rho[p(y_1), q] = \sum_{u=1}^{m} \sqrt{p_u(y_1) q_u} \quad (4.10)$$

Step 8. Compare the similarity:

If $\rho[p(y_1), q] < \rho[p(y_0), q]$, then let $y_1 = \frac{1}{2}(y_0 + y_1)$. (4.11)

Step 9. Compare $\|y_1 - y_0\|$ to a threshold ε:

If $\|y_1 - y_0\| < \varepsilon$, then stop the iteration; otherwise, let $y_0 = y_1$ and go to Step 2.

By using the mean-shift tracking algorithm, the current speaker in the position of the previous image, and the continuous tracking process, we can see that the speaker positions in the two images differ. These calculations also yield additional information regarding the direction and speed of the speaker's motion.

4.1.2 Speaker posture recognition

In general, speakers use numerous postures during speeches, and some of them have noteworthy or particular meanings. Because the postures of a speaker can exercise considerable influence on an audience's attention, the virtual joints extracted by KINECT from the images of the speaker's hands are used for two procedures: posture database construction and posture recognition.

A. Posture database construction

Before any speaker's posture can be recognized, a posture database must be constructed; it must include various types of hand postures. The flowchart of our hand

posture data set construction is shown in Figure 4.7. Starting from hand postures displayed by training skeletons, extract the joint coordinates of each hand and then normalize these joint coordinates based on the corresponding shoulder coordinates and the corresponding lengths of arms. Afterward, give each joint a GMM and construct several GMMs for each hand posture.

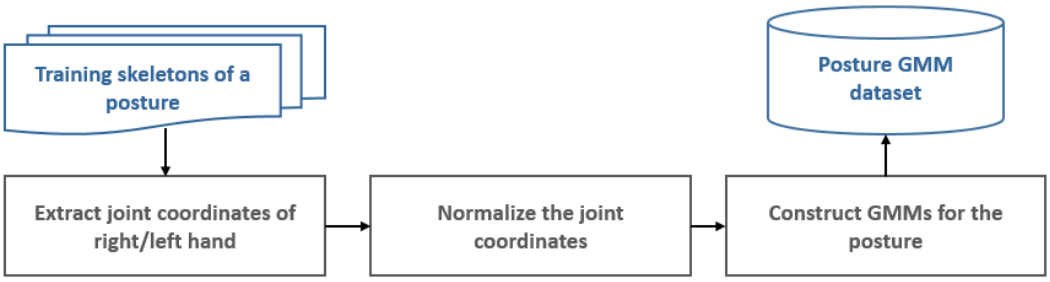


Figure 4.7. Flowchart of posture database construction.

We chose two types of postures from the categories of nonverbal behavior proposed by [16] called **illustrating** and **pointing** when we constructed our posture data set. The definitions of illustrating and pointing are as follows:

Illustrating: Illustrating involves making movements that are directly tied to the speech, serving to illustrate what is being said verbally.

Pointing: Pointing occurs when some part of the body, usually the fingers or a hand, points to a person, a part of the body, an object, or a place.

Furthermore, posture can represent the speaker's emotion at the moment. For example, crossed arms may indicate anxiety, which is either driven by a lack of trust in another person or an internal discomfort and sense of vulnerability (that may, for example, be rooted in childhood trauma). Raising an arm communicates the idea of lifting something up. Done rapidly, it symbolizes throwing things into the air. With both arms, it exaggerates the message further. A typical two-arm-raising gesture shows frustration, as if every feeling of confusion that is weighing the person down must be thrown up into the air. Coupled with a shrug, it indicates confusion.

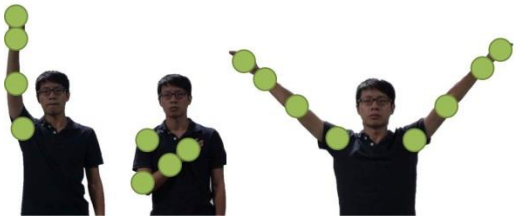


Figure 4.8. Three types of illustrating posture. The rightmost type involves two pointing hands.

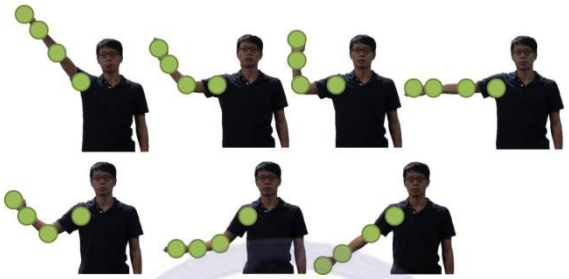


Figure 4.9. Seven types of pointing postures.

According to the aforementioned definitions of illustrating and pointing, ten types of postures related to illustrating and pointing were defined in our posture data set. Figure 4.8 shows the three types of the illustrating posture; Figure 4.9 shows seven types of pointing postures.

Although a depth camera could be used to extract joint position and coordinate information, our system recognizes hand gestures as key speaker actions; only hand joint information is used. The system divides the depth information into several bins, because the precision of KINECT is at the micrometer level, which is too sensitive for posture recognition. The depth data in any one bin has the precision of all data in that bin.

Because each person has arms of some unique length, each person's joints are separated by different distances. It is even possible to ascribe different coordinates to the same person because of a change in viewing angle (Figure 4.10). Thus, it is necessary to normalize the arm joint coordinates to increase the matching rate between

the database and incoming data.

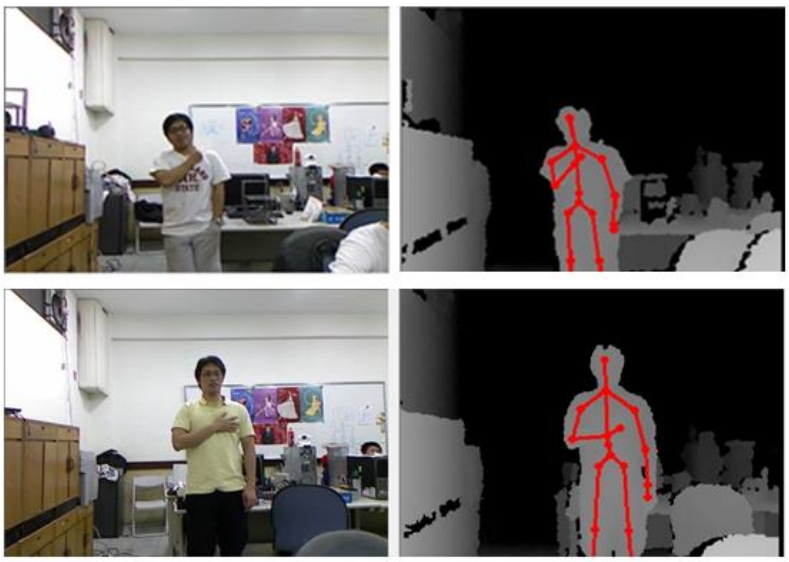


Figure 4.10. Different virtual skeletons of different users.

The arm joint coordinate normalization process is as follows:

1. Define the coordinates of the shoulder to be the origin of the body. Recalculate the coordinates of all joints in terms of the new coordinate system. Suppose that there are n joints for one hand of a user. O_i is the original coordinate of the ith joint for each of the n joints, O_1 is the coordinate of the shoulder, and N is the new coordinate after the shift:

   $$N_i = O_i - O_1, \forall i = 1, \dots, n \quad (4.12)$$

2. Let the length of the upper arm be the standard unit of distance. Normalize and rescale all joints with the new unit in the new coordinate system. Let N_i be the arm joint and let J_i be the coordinate after rescaling:

   $$J_i = \frac{N_i}{|N_2 - N_1|}, \forall i = 1, \dots, n \quad (4.13)$$

After arm joint coordinate normalization has been performed on the data from all users, the system can build a database of postures by applying GMMs to data from users. A GMM (described in Chapter 2) can solve the problem of data variance after

normalization. For each joint, we fit a GMM, which means each joint is represented by several Gaussian probability density functions. In Figure 4.11, the overlapping green circles are different coordinate estimates of the coordinates of same joint. The average (or "mean") location of a joint is marked as a red point.

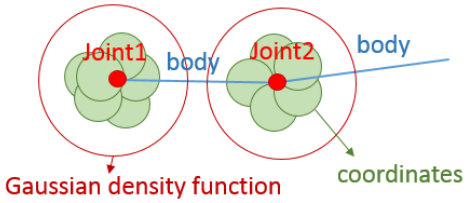


Figure 4.11. Gaussian probability density functions for two joints.

The covariance matrix represents the distribution of the estimated coordinates of a joint. When the standard deviation is small, the variance of the mean is also small, which means only small differences exist between samples at this joint. The weight is the importance of each Gaussian probability density function. Joint coordinates with higher weights occur more frequently. All estimates are assigned equal weight in the present study.

To optimize joint coordinates, the Gaussian probability density function of each joint updates with the joint coordinates. Three parameters must be updated constantly: means, standard deviations, and weights. When updating means and standard deviations, we must consider the probability of the new coordinate in the Gaussian probability density function. A high confidence value for the new coordinate indicates that the new coordinate can adjust the means of the model. If the confidence value is larger than a threshold, the update is performed; if the confidence value is lower than that threshold, the original means and standard deviations are used without an update.

B. Posture recognition

After the hand posture data set construction, it is possible to recognize the

speaker's hand postures. Our hand posture recognition flowchart is shown in Figure 4.12.

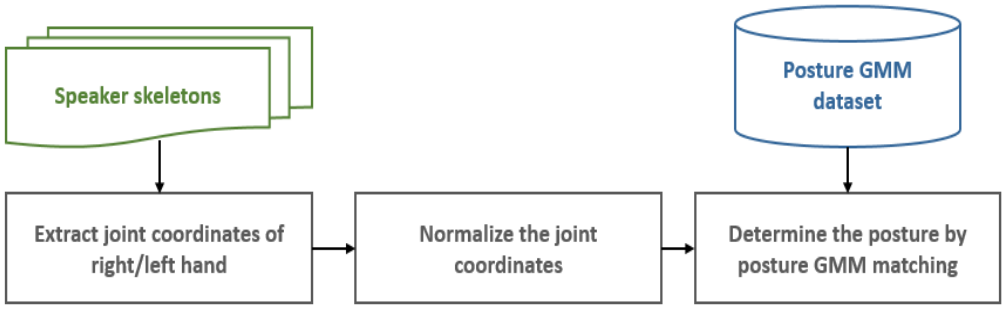


Figure 4.12. Flowchart of posture recognition.

This involves the same steps as those performed in the posture data set construction: extract all joint coordinates of the hands of the speaker and normalize them. Afterward, determine the speaker's hand posture by scanning all GMMs in the posture database to find the most similar posture. After matching the observed posture with a set of several known postures, we interpret the observed posture as an example of the known posture with the largest probability. If the probability of the GMM is greater than or equal to threshold $T_{posture}$, then the testing data can be classified as an example of the known gesture (see Figure 4.13). If the gesture is classified as undefined in the database, it means "relaxing."

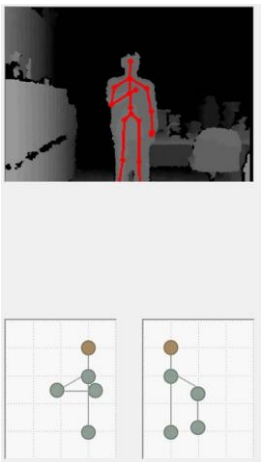


Figure 4.13. Posture detected by GMM.

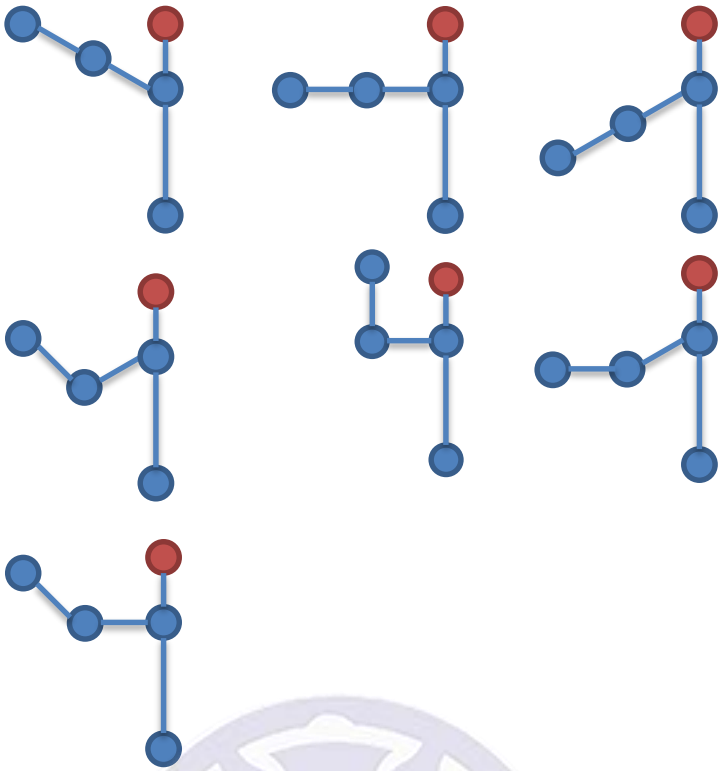


Figure 4.14. Postures of pointing.

Figure 4.15. Postures of illustrating.

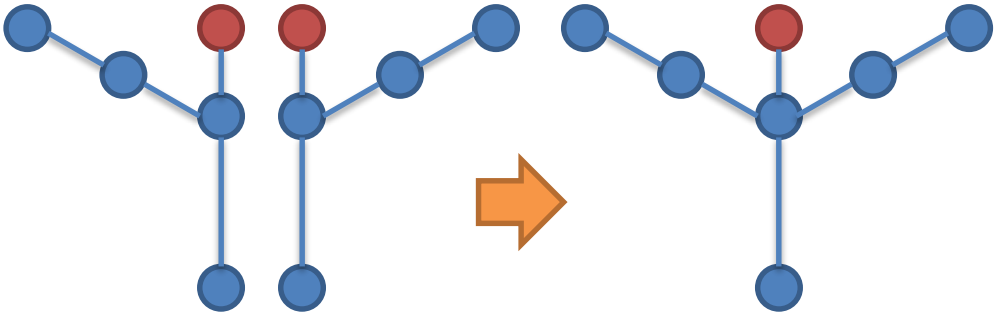


Figure 4.16. Combination of illustrating postures.

The definitions of several hand gestures depend on the postures of illustrating (for emphasis) and pointing; see Figure 4.14 shows seven pointing postures and Figure 4.15 shows two illustrating postures. For example, for pointing, the joints of the elbow and

arm form a straight line, and the joints of the arm are above or below the shoulder. The set of emphasizing hand gestures illustrated in Figure 4.15 contains two distinct hand gestures: raising the hands and folding the arms on the chest. Note that when the speaker is in the posture of pointing, he or she must use only one hand. Thus, when two hands are both detected as being in pointing postures, the speaker's behavior is classified as illustrating instead of pointing (see Figure 4.16).

The system determines suitable PTZ camera actions by analyzing observed images according to the definitions of illustrating and pointing (see Figure 4.17). When our SC subsystem detects illustrating behavior, it is programmed to focus on the moving area of the speaker's hands, so it controls the PTZ camera to continue tracking and shooting the speaker (see Figure 4.18) but simultaneously sends a message to notify the VD that the speaker is illustrating at the current time. For pointing, the focus target is the area pointed to by the speaker's hand, so the SC subsystem instructs the PTZ camera to move and shoot that area (see Figure 4.19) and simultaneously sends a message to the VD.

Figure 4.17. Result of hand posture recognition.

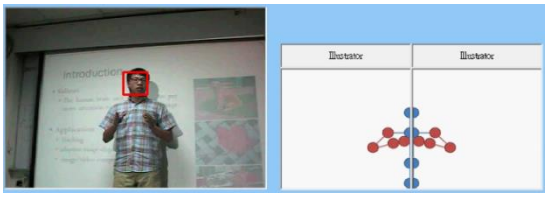


Figure 4.18. Illustrating recognition. The PTZ camera continues shooting the speaker.

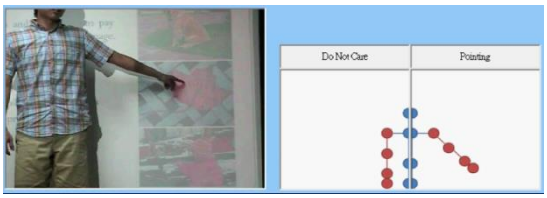


Figure 4.19. Pointing recognition. The PTZ camera moves and shoots the area in which the speaker is pointing.

4.1.3. Camera action

Because the Kinect sensor is on top of the PTZ camera, the lenses of these two cameras may be located along the same vertical line. To determine the pan and vertical tilt angle of the PTZ camera, one must calculate $\emptyset_{hor}$ and $\emptyset_{vert}$ (detailed in Chap. 3), respectively, by finding the target image in the image plane and calculating the angles to the center of the optical axis of the Kinect sensor. The aforementioned parameters are known except d, $\emptyset_{hor}$, and $\emptyset_{vert}$. Therefore, PTZ camera action requires only a simple computation process to generate control signals. The shooting rules are described in Section 4.4.

4.2 Audience Cameraman

The purpose of the AC subsystem is to simulate the camera-control behaviors of a professional cameraman to capture the audience shot. The proposed system must decide which region to shoot before the camera can be aimed at the correct angle. The system flowchart of the AC is shown in Figure 4.20. The AC system is divided into two parts. Two PTZ cameras are mounted together to make a set; one is the global-view camera and the other is the local-view camera. The global-view camera can be regarded as analogous to a photographer's eyes. It can monitor the whole audience and help with ROI detection. The local-view camera can be regarded as analogous to a photographer's camera. When the system has detected the ROI, the local-view camera can receive camera action control signals to shoot the ROI. The AC system performs face detection

in local-view images, so that PTZ can shoot specific audience members with camera action rules. If an audience member's face is detected, the AC controls the local-view camera to zoom and focus on that specific audience member; otherwise the AC performs ROI detection again. The output shot provided by the local-view camera is transferred to the VD; the VD performs shot selection.

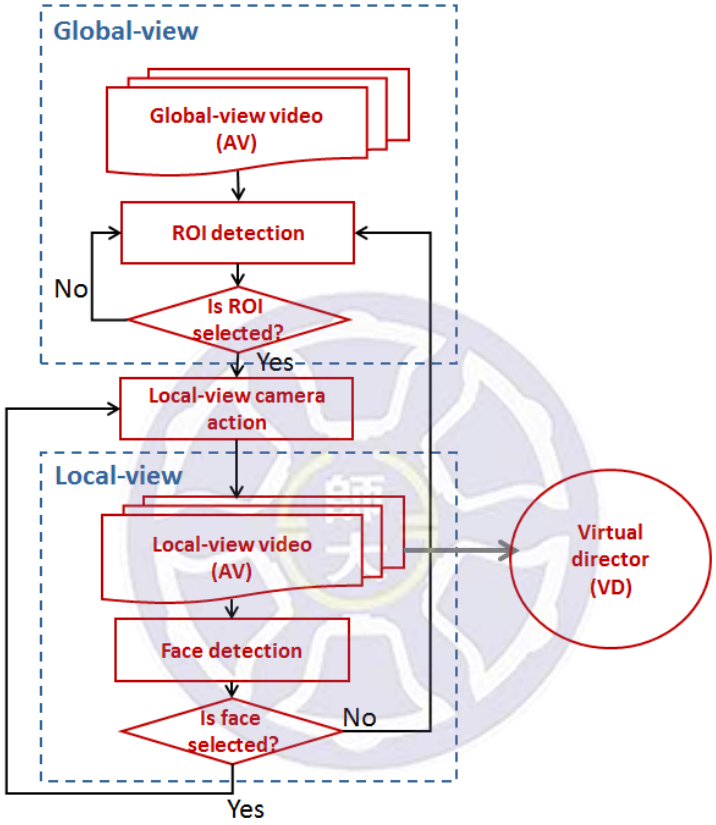
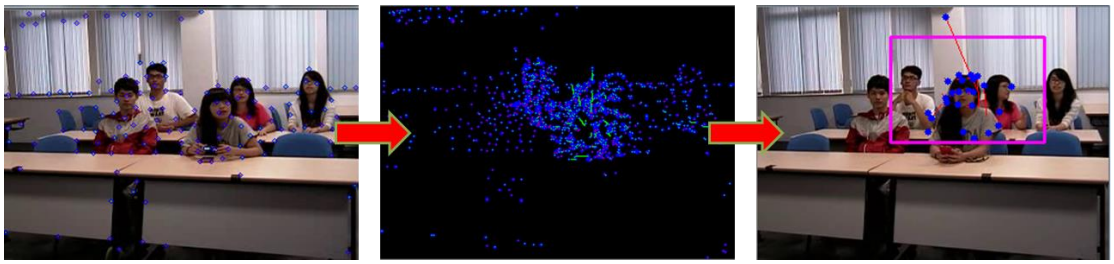


Figure 4.20. Flowchart of audience cameraman.

A. ROI detection with the global-view camera

First, the AC obtains input videos from the global-view camera; then the AC scans for audience motion features to locate ROI candidates, which are the regions in which audience or crowd events have the highest probabilities to appear. The system uses "Features from Accelerated Segment Test" (FAST) [75] corner detection to find the feature points; the system uses optical-flow estimation to estimate the motion vector of each feature point. Pixels are measured in terms of motion feature density to identify

ROI candidates. An ROI detection result is shown in Figure 4.21.



(a) (b) (c)

Figure 4.21. ROI detection (a) motion feature map (b) motion feature density map (c) ROI candidate.

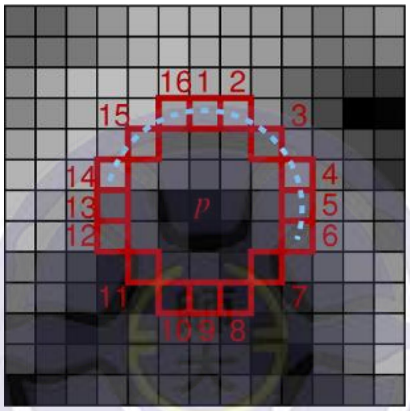


Figure 4.22. FAST corner detection [75].

A.1 FAST corner detection

FAST corner detection is an approach used to reduce the computational time of optical-flow estimation significantly. As shown in Figure 4.22, let point p be the center of the image, and make an approximate circle with a circumference that consists of 16 pixels. Then observe the 16 pixels of the circumference, and mark the pixels with the following state:

$$\text{state}_i = \begin{cases} dark & , I_i \leq I_p - t \\ bright & , I_i \geq I_p + t \end{cases} \tag{4.14}$$

where state_i is the state of pixel i, $i = 1, \ldots, 16$. I_i is the intensity value of pixel i, I_p is the intensity value of point p, and t is a threshold. If m pixels of the 16 pixels have the same state, the point p is a FAST feature point; in this study, the m is 9.

Figure 4.23 shows a FAST corner detection result.

Figure 4.23. FAST corner detection result.

A.2 Optical-flow estimation

After FAST feature detection, the optical-flow method calculates the motion vector between successive images. The motion vector extraction applies Lucas-Kanade optical-flow [76] estimation. The proposed optical-flow method is based on a pyramid architecture involving multiple layered images with different resolutions. The system performs coarse tracking on a low-resolution image; it performs fine tracking on a high-resolution image.

The fundamental theory of the optical-flow method assumes the intensity invariance of some pixel between successive images of a moving object taken within a very short time. The velocity vector is the motion direction of the pixel. $E(x, y, t)$ is the intensity value of pixel (x, y) at time t. After Δt time, Δx and Δy express the displacement; the equation is as follows:

$$E(x, y, t) = E(x + \Delta x, y + \Delta y, t + \Delta t) \tag{4.15}$$

The Taylor expansion of the aforementioned Equation 4.15 is as follows:

$$E(x + \Delta x, y + \Delta y, t + \Delta t) = E(x, y, t) + E_x \Delta x + E_y \Delta y + E_t \Delta t \tag{4.16}$$

Then, by substituting (4.16) into (4.15) and dividing by Δt at the same time, and performing simplification, we can obtain:

$$E_x \frac{\Delta x}{\Delta t} + E_y \frac{\Delta y}{\Delta t} + E_t = 0 \tag{4.17}$$

When Δt is lower than our threshold of acceptance, we can rewrite (4.17) as:

$$E_x \frac{dx}{dt} + E_y \frac{dy}{dt} + E_t = 0 \quad (4.18)$$

Let $u = \frac{dx}{dt}$ and $v = \frac{dy}{dt}$ be the motion velocities of the horizontal vector and vertical vector, respectively. Therefore, (u, v) is the proposed optical-flow estimation result.

The optical-flow method can track FAST feature points as follows. Considering an image sequence $\mathrm{I} = \{I_1, I_2, \ldots, I_T\}$, let $[P_x, P_y]^t$ be the position of a feature point P, where $t \in \{1, \ldots, T\}$. $I_t(P)$ is the intensity value of P in the image. τ is the time interval. The motion between successive frames can be represented as follows:

$$I_t(x, y) = I_{t+\tau}(x + \Delta x, y + \Delta y) \quad (4.19)$$

where $\vec{v} = (v_x, v_y)$ is the velocity for P during the time interval τ.

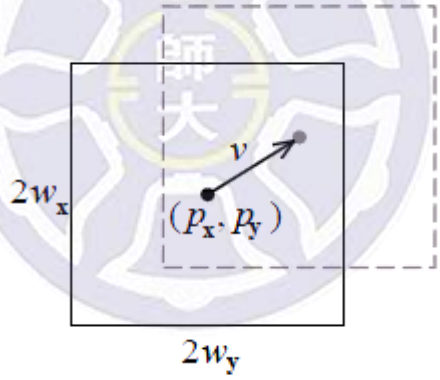


Figure 4.24. Search window of proposed optical-flow method.

The purpose of the optical-flow method is to find the optimal $\vec{v}$ and minimize the matching error ε. Let w_x and w_y be the half-width and half-height of the search window (see Figure 4.24). The definition of ε is as follows:

$$\varepsilon(\vec{v}) = \varepsilon(v_x, v_y)$$

$$= \sum_{x=p_x-w_x}^{p_x+w_x} \sum_{y=p_y-w_y}^{p_y+w_y} (I_{t+\tau}(x + \Delta x, y + \Delta y) - I_t(x + v_x, y + v_y))^2 \quad (4.20)$$

To obtain the optimized values v_{xopt} and v_{yopt}, we process the partial

derivatives of the function ε with respect to v_x and let v_y be 0. We can obtain:

$$\left.\frac{\partial \varepsilon(v_x)}{\partial v_x}\right|_{v_x=v_{xopt}} = 0 \tag{4.21}$$

$$\left.\frac{\partial \varepsilon(v_y)}{\partial v_y}\right|_{v_y=v_{yopt}} = 0 \tag{4.22}$$

The Taylor expansions of Equations (4.21) and (4.22) are as follows:

$$\frac{\partial \varepsilon(\vec{v})}{\partial v_x} = -2\sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\left[I_{t+\tau}(x+\Delta x, y+\Delta y) - I_t(x,y) - \frac{\partial I_t}{\partial x}v_x - \frac{\partial I_t}{\partial y}v_y\right]\cdot \frac{\partial I_t}{\partial x} \tag{4.23}$$

$$\frac{\partial \varepsilon(\vec{v})}{\partial v_y} = -2\sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\left[I_{t+\tau}(x+\Delta x, y+\Delta y) - I_t(x,y) - \frac{\partial I_t}{\partial x}v_x - \frac{\partial I_t}{\partial y}v_y\right]\cdot \frac{\partial I_t}{\partial y} \tag{4.24}$$

where $I_{t+\tau}(x+\Delta x, y+\Delta y) - I_t(x,y)$can be explained as the differential of image with respect to time.

$$\forall(x,y) \in ([p_x - w_x, p_x + w_x], [p_y - w_y, p_y + w_y])$$

$$I_t = \frac{\partial I}{\partial t} = I_{t+\tau}(x+\Delta x, y+\Delta y) - I_t(x,y) \tag{4.25}$$

Because images are digital signals, the pixel values are not continuous values; they are discrete integers. We can perform finite difference calculations of the partial derivative values. Let the image width be *W* and height be *H*; then one may calculate:

$$I_x = \frac{\partial I_t}{\partial x} = \begin{cases} I_t(x+1,y) - I_t(x,y) & \text{, if } x = 1 \\ \frac{I_t(x+1,y) - I_t(x,y)}{2} & \text{, if } 1 < x < W \\ I_t(x,y) - I_t(x-1,y) & \text{, if } x = W \end{cases} \tag{4.26}$$

$$I_y = \frac{\partial I_t}{\partial y} = \begin{cases} I_t(x,y+1) - I_t(x,y) & \text{, if } y = 1 \\ \frac{I_t(x,y+1) - I_t(x,y)}{2} & \text{, if } 1 < y < H \\ I_t(x,y) - I_t(x,y-1) & \text{, if } y = H \end{cases} \tag{4.27}$$

Now substitute (4.26) and (4.27) into (4.23) and (4.24), respectively:

$$\frac{\partial \varepsilon(\vec{v})}{\partial v_x} = -2\sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\left[I_t I_x + v_x I_x^2 + v_y I_x I_y\right] \tag{4.28}$$

$$\frac{\partial \varepsilon(\vec{v})}{\partial v_y} = -2\sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y}\left[I_t I_x + v_y I_y^2 + v_x I_x I_y\right] \tag{4.29}$$

Then transform them to matrix form:

$$\begin{bmatrix} \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y} I_x^2 & \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y} I_x I_y \\ \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y} I_x I_y & \sum_{x=p_x-w_x}^{p_x+w_x}\sum_{y=p_y-w_y}^{p_y+w_y} I_y^2 \end{bmatrix} \begin{bmatrix} v_x \\ v_y \end{bmatrix} = \begin{bmatrix} I_x I_t \\ I_y I_t \end{bmatrix} \quad (4.30)$$

Rewrite (4.30) to $G\vec{v} = -b$, and $\vec{v} = -G^{-1}b$ is the proposed vector.

If the motion is fast, we can increase the search window size, but the computing time will increase too. In the present research, the search window size was 10×10. If the displacement of the feature point is excessively large, it can be challenging to track the feature point.

The Lucas-Kanade optical-flow method provides a solution to overcome this problem by building an image pyramid. First, we define I^L to be the L th layer of the image pyramid where $L = \{0,1,2,\dots,m\}$ and I^0 is the original image. By using bilinear interpolation, we can measure the intensity of pixels on the next layer. The following is the bilinear interpolation function:

$$\begin{aligned} I^L(x,y) = {} & \frac{1}{4} I^{L-1}(2x,2y) \\ & + \frac{1}{8}[I^{L-1}(2x-1,2y) + I^{L-1}(2x+1,2y) + I^{L-1}(2x,2y+1)] \\ & + \frac{1}{16}[I^{L-1}(2x+1,2y) + I^{L-1}(2x-1,2y+1) + I^{L-1}(2x+1,2y+1)] \quad (4.31) \end{aligned}$$

Therefore, we can perform tracking on the image pyramid. First, the displacement of the m^{th} layer is computed. Sequentially, the $m-1^{th}$ layer must be computed until $m = 0$. The tracking process is to track from the low-resolution layer to the high-resolution layer. In other words, we find rough features at low resolution and update the location layer by layer. By using the technique, we can improve the stability of the optical-flow estimation. The image pyramid in the present research was a four-layer architecture, therefore $m = 3$.

A.3 ROI selection

Next, the motion information from optical-flow estimation must be converted into motion density maps. Each motion density map is assembled from the statistics of optical-flow estimation through a motion point and its moving distance. As high and low densities give different shades and intensities, a low-density area means smaller motion, whereas a high-density region means larger motion. A region with relatively high density and with relatively numerous and concentrated motion vectors can be selected as a candidate ROI.

However, the aforementioned method may choose multiple ROI candidates with the highest motion density. And some noise in the image might also be chosen as ROI candidate. To avoid those situations, this study applies the STA network model (described in Chapter 2) to record a captured ROI in an attention map. From the information in the attention map the system must determine whether the candidate ROI can be considered a feasible ROI. The ROI candidates are then entered as inputs into the STA (see Figure 4.25). The STA model can record and provide additional information to help the system to determine the ROI that is most suitable as a photographic subject. Further, the system computes the relative distance between the location of the ROI center and the image center and outputs motion-control signals for the local-view camera based on camera action rules.

(a) (b) (c)

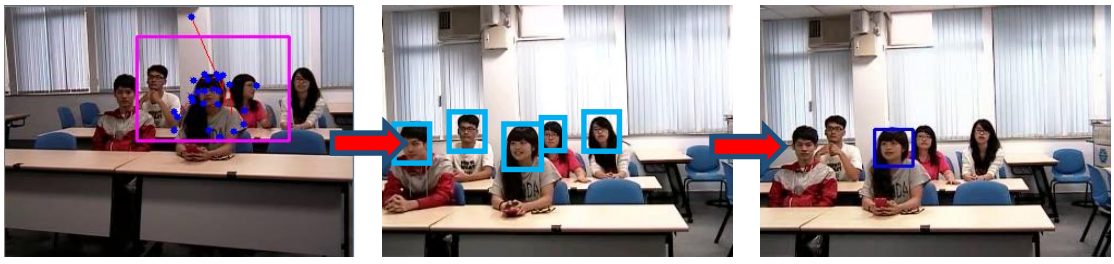
Figure 4.25. ROI selection (a) ROI detection result (b) the shot after camera steering (c) the attention map for selection.

B. Face detection with the local-view camera

Because the AC subsystem is focused on the audience, numerous faces might appear in the AC shot simultaneously; therefore all face information is vital. At the same time, faces are the most easily recognizable parts of the audience. The local-view camera captures video information from the ROI by considering aesthetics and by analyzing optical characteristics. The AC system simulates professional photography shooting skills through the aforementioned process.

B.1 Face detection method

To find the audience in the shot, we must determine face locations. The AC uses the same face detection method as the SC uses. The subject's face is generally selected, because it is simple and uniform relative to the body, which is covered by clothes. The AdaBoost algorithm described in Section 4.1 is also applied in the AC subsystem. After the completion of face detection, the AC subsystem records the center coordinates, height, and width of each individual face. Each individual face is a principal candidate of a salient object. Finally, the recorded face coordinates nearest to the center of a recent ROI are selected as a target (see Figure 4.26).



(a) (b) (c)

Figure 4.26. Face detection process (a) ROI detected result (b) the face results after camera steering (c) the face of salient object.

B.2 Camera action and shot composition

Faces are not only salient objects for the cameraman; the center of the screen can

also be a salient object. For example, when the salient object is the audience, to highlight specific audience members, the proportion of the selected audience in the picture must not be too small; one-third of the size of the image is best. The details of the shooting and camera action rules are described in Section 4.4.

4.3 Hall Cameraman

Timely changes to hall view shots allow viewers to watch with greater focus. According to an earlier study [1], after viewers have watched the same shot for too long, they can easily become distracted, thus it is necessary to change the shot occasionally. However, when the speaker is explaining some critical information, it is not suitable for the VD to change the current shot to an audience shot. The hall view is very useful at this time, because we do not discard all speaker information but we can also capture audience information. We call this type of shot a “safe shot” or “establishing shot.”

The software that implements the HC assumes that the HC should only perform a camera action when the VD gives a visual instruction. Otherwise, the HC simply takes a wide shot including both the speaker and the audience. The HC defaults to passive mode, in which it only transmits the hall view and waits for visual instruction. For example, if the VD detects that the speaker is waving his hand and simultaneously detects a big motion in the hall view shot, the VD gives a pan instruction to the HC. Once the HC receives the instruction, the HC performs a pan camera action so that both the speaker and the area with motion can be in the shot.

The details of VC camera control and shot composition are addressed in next section.

4.4. Rules of Camera Action

Cameramen share a common language of shots and camera moves. Numerous

cameramen plan their shots in advance, seeking emotional resonance and story-telling power through compositional choices. Several types of shots are needed in a lecture recording: close-up shots, medium shots, and establishing shots. A close-up positions a person or object as the paramount element in the frame. A head-and-shoulders framing of one person is often used in documentary interviews. Close-ups of objects allow the audience to see the objects clearly without other visual distractions. A medium shot shows the larger setting or situates people in their environment to create a sense of location and context. An establishing shot provides an overall view of the lecture room. People, if they are included in an establishing shot, appear very small.

(a) (b)

Figure 4.27. Pointing detection (a) Pointing detected (b) change to a close-up shot.

When a speaker points a hand at some item within an area, the shot should include the region in which the speaker is pointing. The hand and pointing region must be focused on the screen for a close-up shot. Therefore, in this study, we set the PTZ camera to perform pan and tilt operations and zoom-in operations to the pointing region. In other words, if the system determines that the speaker is making a pointing gesture, the SC gives the region a significant close-up shot (see Figure 4.27) and then sends messages to the VD to inform it that the speaker is pointing.

However, when the speaker is using an illustrating posture, the camera should focus on the speaker. Therefore, the SC performs pan and tilt operations to position the speaker near the center of the screen and then sends messages to the VD to inform it that the speaker is illustrating now.

In addition to the obvious pointing and illustrating postures, another hand gesture that we have not yet defined, called “relaxing,” exists. The system simply ignores this undefined posture (see Figure 4.28). In this situation, the PTZ camera continues tracking the speaker (see Figure 4.29), while the system continues to check whether the speaker is using any pointing or illustrating gesture.

Figure 4.28. Posture of relaxing.

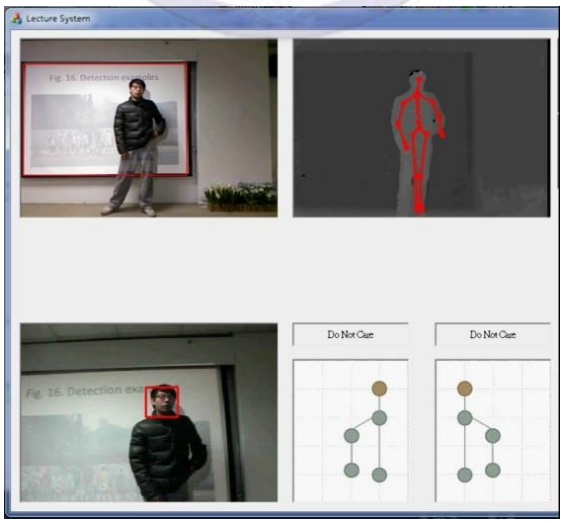


Figure 4.29. Result of relaxing.

The relevant speaker information is obtained, including the speaker's position and posture (pointing, illustrating, or relaxing) and whether the speaker is using any

teaching tools (laser pointer or baton). After this, all VCs automatically control their PTZ cameras based on the rules listed in Appendix (see Table A.1) to shoot their ideal shots, from which the VD performs shot selection. Aside from video signals, the VCs also send event messages to the VD to inform it of current events.

Chapter 5 Virtual Director

The workflows of a real director are simulated by the system's automatic shot selection and visual instruction workflows. This work is done by the "virtual director" (VD). Figure 5.1 shows the two major tasks of the VD: shot selection and visual instruction. Before those tasks begin, the VD must perform content analysis and evaluate the scores of shots. Hence, the VD considers the photographic aesthetics, optical analysis, action analysis, and continuity analysis for evaluating the quality of each shot received from the VCs.

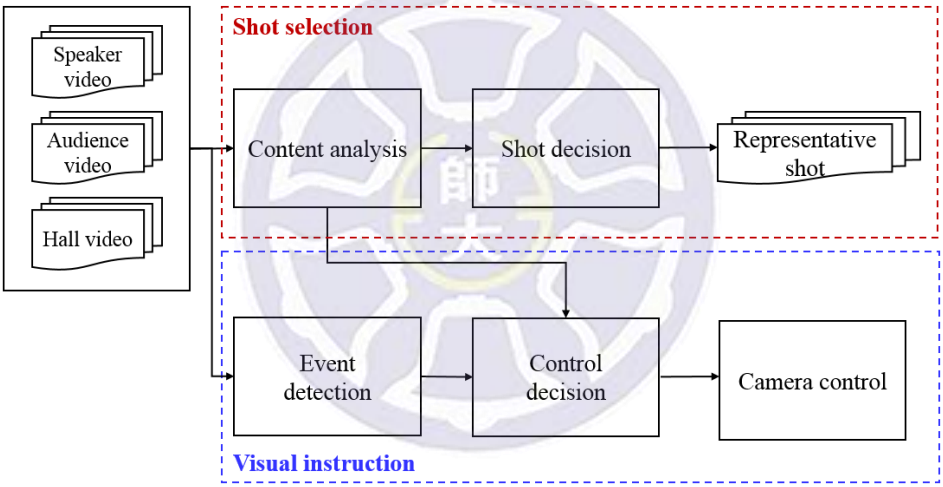


Figure 5.1. Procedures for VD shot selection and visual instruction.

Content analysis is introduced in Section 5.1. The details of shot selection are addressed in Section 5.2. Section 5.3 covers visual instruction.

5.1. Content Analysis

Content analysis is a technique to quantize the quality of image according to a series of evaluation rules. The evaluation rules include optical analysis, aesthetic analysis, continuity analysis, and action analysis, as shown in Table 3.1. The optical

analysis includes exposure, region of focus (ROF or picture sharpness), and saturation information. The aesthetic analysis considers the shot's visual merits in terms of the rule of thirds, visual balance, and size of saliency. The continuity analysis includes illuminance continuity, color continuity, scene continuity, and position continuity; continuity analysis is intended to prevent the confusion of human vision and intuition on the screen as much as possible; a video with favorable continuity appears smooth and continuous during shot changes. Finally, action analysis determines the movements of PTZ cameras of the VC subsystem that just be made, because the pictures shot after those camera movements may involve the content in which viewers might be interested.

Table 5.1. Evaluation rules of content analysis

Aesthetic analysis	Optical analysis	Continuity analysis	Action analysis
1.Rule of third 2.Visual balance 3.Size of subject	1.Exposure 2.Regions of focus 3.Saturation	1.Illuminance 2.Color 3.Scenery 4.Position	Camera motion -Hold and move -After zoom-in -After zoom-out

The flowchart of content analysis is shown in Figure 5.2. To evaluate the scores of video shots during the content analysis stage, our system must extract several feature points. Aesthetic analysis focuses on photographic composition (i.e., where the subject is located in the image, which affects human visual reactions). Therefore, the location of the subject should be determined first, and then the saliency map can be constructed to determine the positions of salient objects. Action analysis assigns action scores by detecting camera motion. Optical-flow estimation is a suitable way to determine this. In addition, to avoid choosing a blurred shot, the system produces a region of focus (ROF) map to detect the regions on which the camera focuses. The saturation and exposure are also considered, so each frame is analyzed in the HSV color space. These

criteria are illustrated in this section.

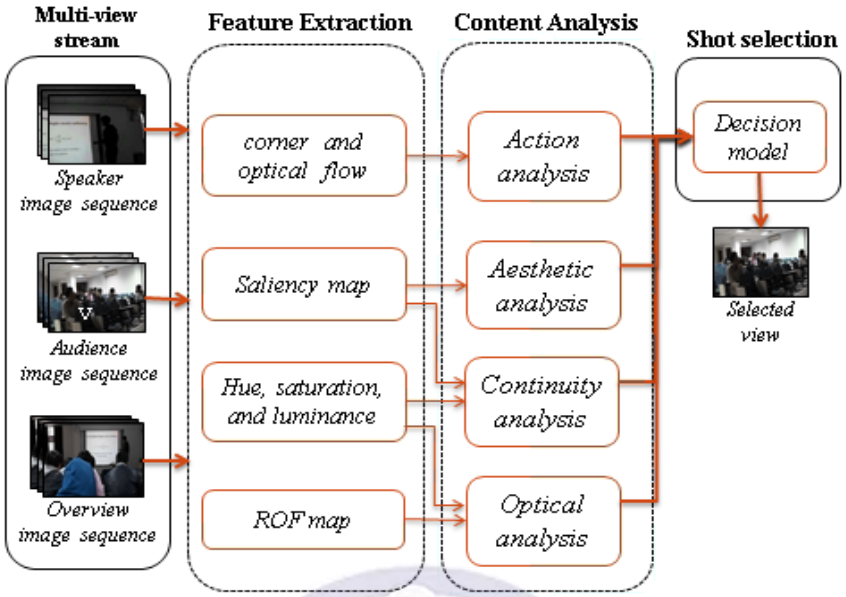


Figure 5.2. Flowchart of content analysis.

5.1.1 Aesthetic analysis

The first object that viewers notice when watching a video is called the salient object. In general, the location of the salient object in the picture and how much space it occupies affect human aesthetic perceptions of the image. The photographer often tries to convey a smooth visual effect by placing the salient object in an appropriate position. According to [30], we can summarize three types of visual rules for taking an artistic photo: rule of thirds, visual balance, and size of salient object.

A. Rule of thirds

The rule of thirds is a "rule of thumb" or guideline that applies to the process of composing visual images, such as designs, films, paintings, and photographs. The guideline proposes that an image should be imagined as divided into nine equal parts by two equally spaced horizontal lines and two equally spaced vertical lines and that pivotal compositional elements should be placed along these lines or at their

intersections. Proponents of the technique claim that aligning a salient object with these points creates more tension, energy, and interest in the composition than simply centering the subject (see Figure 5.3). The intersection of two lines is sometimes called a power point or a crash point. Points of interest in the photo are not required to actually touch one of these lines to take advantage of the rule of thirds. For example, in a photograph of a sunset, the brightest part of the sky near the horizon where the sun has recently set does not fall directly on one of the lines but does fall near the intersection of two of the lines, close enough to take advantage of the rule.

Figure 5.3. Rule of thirds.

B. Visual balance

The visual balance guideline emphasizes the balance of the image. In visual balance, each area of a painting suggests a certain visual weight, a certain degree of lightness or heaviness. Visually, transparent areas seem to weigh less than opaque areas. When a clear salient object or an obvious line structure appears in an image, this object produces relatively heavy visual weight, and traditional guidelines stress that the weight ratio of deployment must be balanced. For example, when a salient object is located in the upper right of the image, while the other bits are in the bottom-left portion of the main image, the image is visually balanced (Figure 5.4).

Figure 5.4. Visual balance.

Figure 5.5. Comparison between different salient object sizes.

C. Size of salient object

The size of the salient object is essential to highlight the theme of an image. By controlling the size of the salient object in an image (see Figure 5.5), we can convey different themes. Depending on particular needs, the size of the salient object might have different restrictions. For example, when the target is the speaker, to highlight the speaker, the proportions of the speaker in the picture cannot be too small (one-third of the image size is optimal). In addition, to reserve some screen space for projection screens and lecture materials, we must maintain some blank space in front of the speaker's face orientation; this practice is called "blank-leaving." The blank-leaving also allows us to maintain the visual aesthetics of the screen shot.

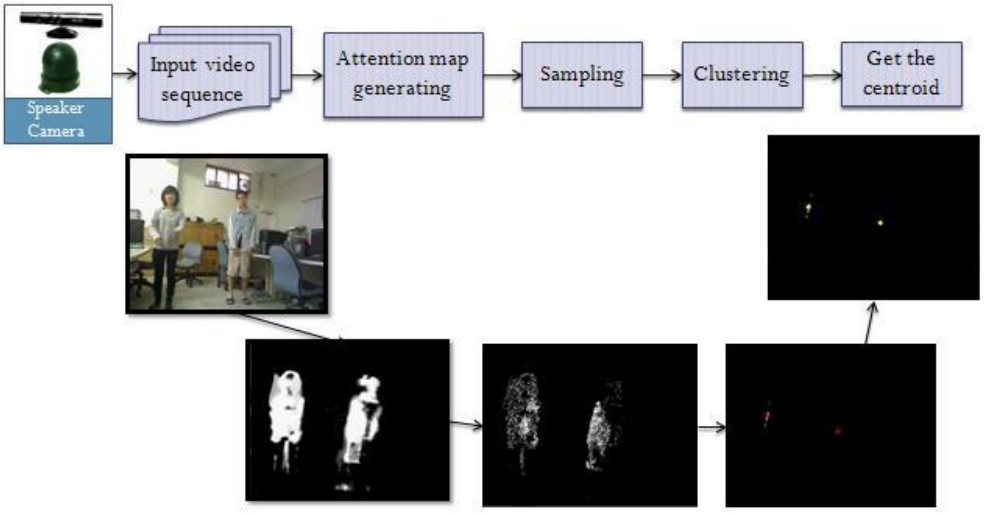


Figure 5.6. Flowchart of salient object detection.

D. Salient object detection

Salient object analyzers consider salient object features. It is also called "salient object detection," "visual attention," or "saliency mapping." The salient object detection flowchart is shown in Figure 5.6. Our analysis component actually simulates human vision systems. First, the system generates an attention map, which includes two types of features: static and dynamic salient features. To find the centroids of the salient objects, we measure the density of the attention map by a sampling method and cluster the sampling data by a mean-shift algorithm. Thus, the clustered centroids represent the locations of the salient objects.

We propose a hybrid approach that combines static and dynamic saliency information. The static salient features are for single-image cases, and the dynamic salient features are specifically for sequential images. Dynamic salient features are normally detected by motion, whereas static salient features are normally detected by high contrast. The details of static and dynamic salient object analysis are described in the following sections.

D.1. Static salient object analysis

Figure 5.7 shows an example in which the colors of figure 5.7(a) differ from those

of figure 5.7(b), but a typical human can easily discern that block shapes are shown in both images. Even though figure 5.7(c) and figure 5.7 (d) use the same colors and shapes, the difference is still apparent. Finally, figure 5.7(e) and figure 5.7(f) use the same color and a similar arrangement, but the shapes are different; the pattern profile can be discerned. These figures show that high contrast is a common feature in these figures and their background patterns; however, color and shape are not the most critical factors in attracting human attention. According to research [23] to identify the most essential features of human perception, high-contrast graphics attract attention, thus a high-contrast region is a possible location of a salient object.

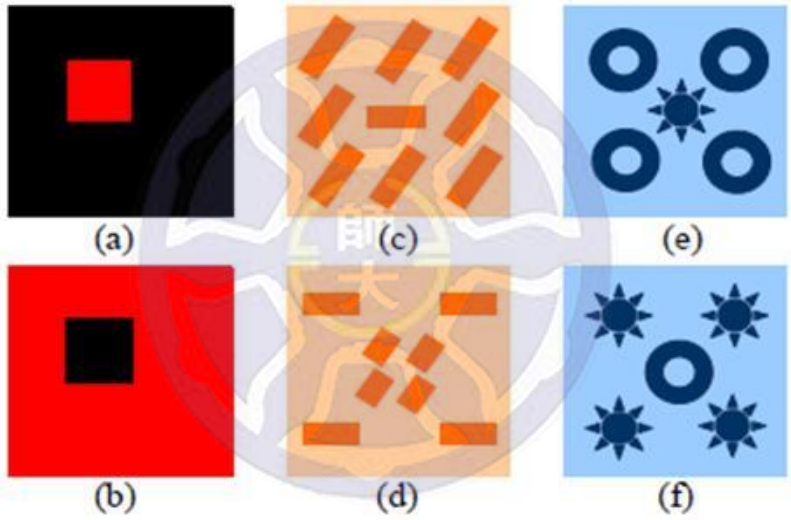


Figure 5.7. Examples of images that attract human attention [23].

In static salient object analysis, we use the method of [77] to calculate the contrast at each image scale by constructing a scale-invariant saliency map through a multiple-scale analysis. The underlying idea of this multiple-scale method is to calculate the image contrast at an image scale matching the feature scale. The large-scale features are highlighted at a coarse scale, and the small-scale features are highlighted at a fine scale (see Figure 5.8).

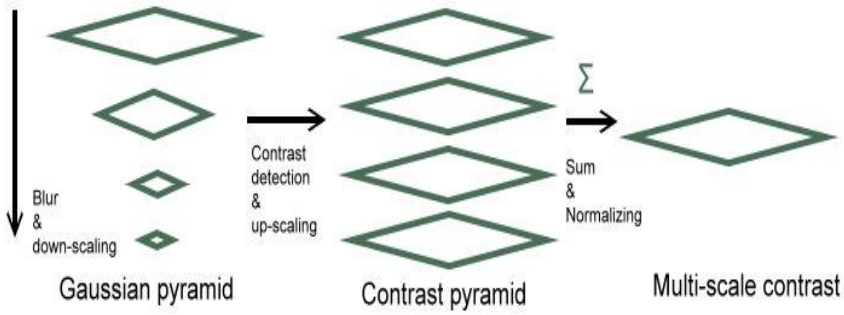
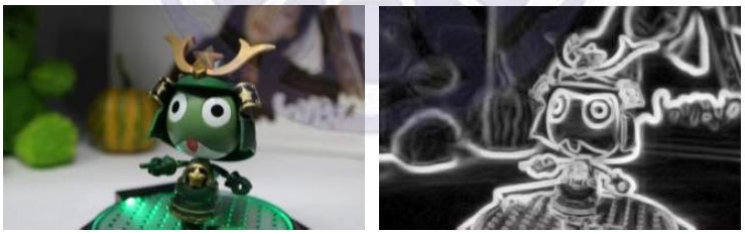


Figure 5.8. Flowchart of multiple-scale contrast detection.

First, we transform the image into a perceptually uniform LUV color space. Then, we build a Gaussian image pyramid from the image. Building a Gaussian pyramid involves creating a series of images that are weighted using a Gaussian average (Gaussian blur) and scaled down. When this technique is used multiple times, it creates a stack of successively smaller images, with each pixel containing a local average that corresponds to a pixel neighborhood on a lower level of the pyramid.



(a) (b)

Figure 5.9. Results of static salient object analysis (a) the original image with a blurred object (background) and a clear object (main object) (b) the static saliency map of (a).

Next, we build the contrast pyramid by calculating the contrast map at each scale. The multiple-scale contrast feature is a linear combination of contrasts in a Gaussian image pyramid. The final step is reconstructing the saliency map from the contrast pyramid by summing up the contrast map at all the scales. The multiple-scale contrast highlights the high-contrast boundaries by giving low scores to the homogenous regions

inside the salient object, because we always obtain stronger results on the edges or depth discontinuity regions. This static salient object analysis is useful for distinguishing a blurred object (out of focus) from a clear one (see Figure 5.9).

D.2. Dynamic salient object analysis

We apply our STA neural network as our dynamic salient object detection method. As described in Chapter 2, STA salient object analysis is useful for distinguishing objects with single moves or repeated moves in a frame, as shown in Figure 5.10. More details can be found in [31].

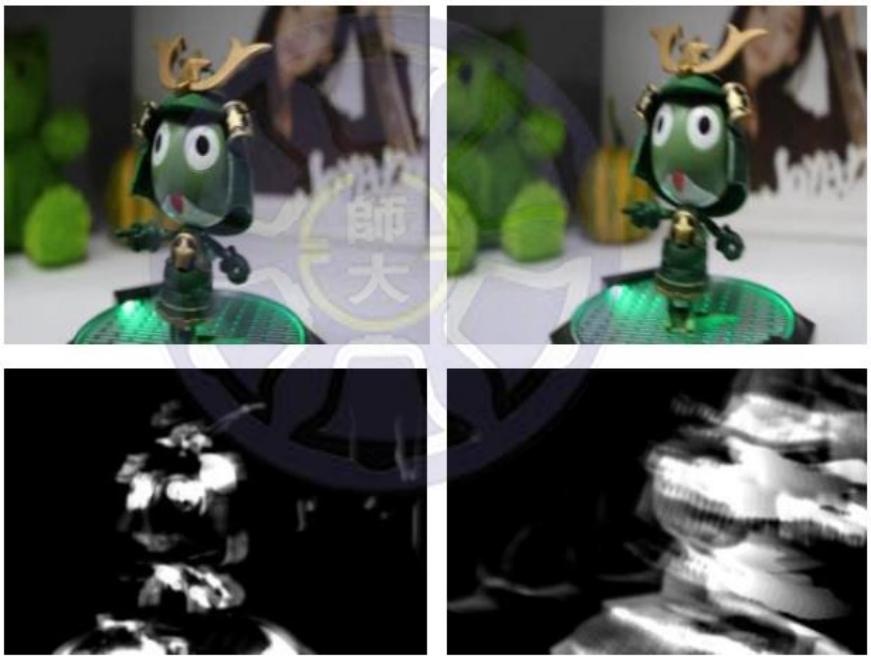


Figure 5.10. Dynamic salient object analysis of a camera that shakes once (left column) and a camera that shakes repetitively (right column).

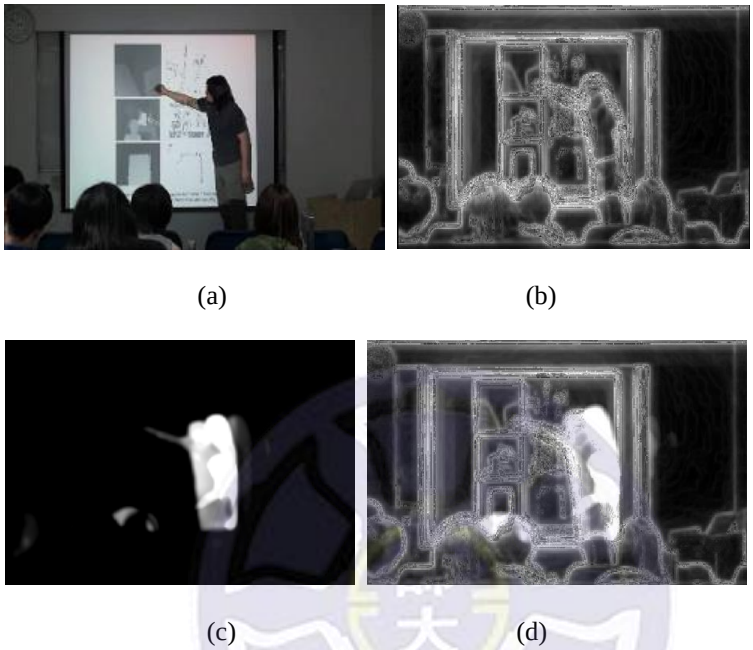
D.3. Saliency map segmentation

When we have generated static and dynamic saliency maps, these two maps can be unified into a single saliency map, called an attention map, using linear combination.

$$A(x,y,t) = \lambda_1 \cdot S_s(x,y,t) + \lambda_2 \cdot S_d(x,y,t) \quad (5.1)$$

where $A(x,y,t)$ is the unified saliency map in frame t, $S_s(x,y,t)$ is the static

saliency map, $S_d(x, y, t)$ is the dynamic attention map, and λ_1, λ_2 are predefined constants. A single frame result of a saliency map is shown in Figure 5.11. Figure 5.12 shows the salient object analysis result of sequential frames from a recording of a real lecture.



(a) (b)

(c) (d)

Figure 5.11. The saliency map (a) input image (b) static saliency map (c) dynamic saliency map (d) combined saliency map.

Because the focus of attention might be on more than one point in the image, it is necessary to extract the attention region by an image segmentation method called saliency map segmentation. The segmented image is called a pattern, and a pattern can have only one focus of attention.

The proposed image segmentation method is the mean-shift algorithm [23]. The mean-shift algorithm can be applied in numerous research areas; the proposed system uses it for clustering. Because the system cannot predict how many salient objects can be found in an image, we use the "T is equal to S" type of mean-shift clustering algorithm: let each point in T be the initial point, and perform the mean-shift algorithm. The points could be seen as being part of the same group if they converge to the same

point cluster centroid. Because the mean-shift algorithm is an iterative algorithm, we must go through numerous iterations until it converges.

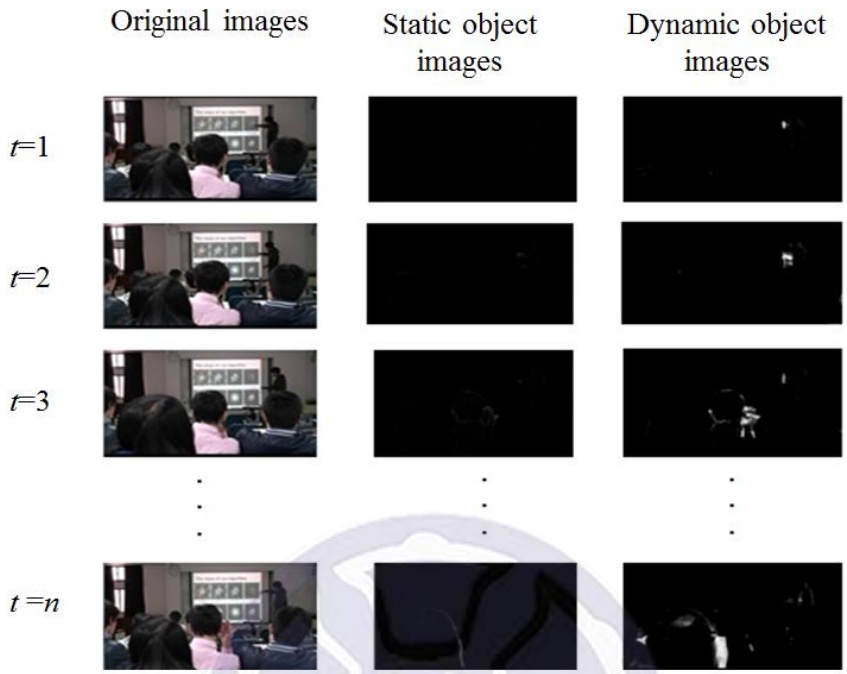
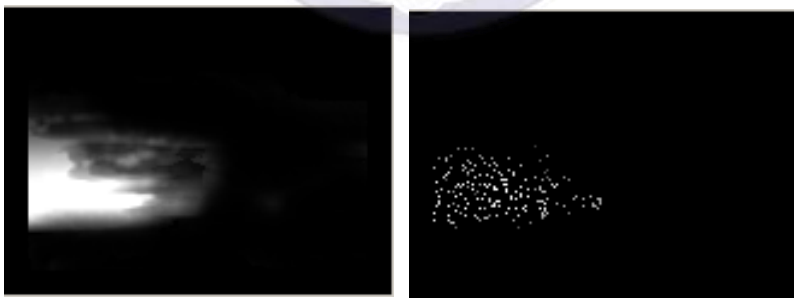


Figure 5.12. Salient object analysis in real lecture image sequences. The original image sequences are in the first column (from top to bottom). The static salient object analysis results are in the second column. The dynamic object analysis results are in the third column.



(a) (b)

Figure 5.13. Attention map sampling (a) attention map (b) sample points

The performance is extremely low if the algorithm uses all the attention points to process the attention image (see Figure 5.13(a)). Therefore, a sampling method is proposed here to reduce the number of attention points to be processed. These attention points from sampling methods are referred to as attention samples; they constitute a collection called an attention sample set (see Figure 5.13(b)). Because the number of

attention sample points is low, it is possible to speed up the convergence of the mean-shift algorithm. Although the sampling method reduces the number of attention points, the distribution of these sample points is maintained, and it shows contours similar to those of the original attention map.

The brighter the saliency map is, the more likely it is that a salient object appears. Therefore, the sampling rate is higher for a high brightness area. An area with lower brightness is still sampled, but with fewer sampling points. Sample point addition is used to yield S_{n+1}^{+} by sampling $I_{n+1} - I_n$; the sampling algorithm is shown below:

For $x = 0$ to image length

For $x = 0$ to image width

If $i(x, y) > T$

$m = \alpha \cdot \frac{i(x,y)}{255}$, random ($\gamma$).

If $m > \gamma$

Insert point (x, y) to sample set S.

where T and α are the brightness threshold and constant, respectively. $i(x, y)$ is the intensity value at pixel (x, y) of the attention map. γ is a random value in the range [0,1].

Each brightness point value $i(x, y)$ on the saliency map can be used to calculate a value of m, and if the value is more than a randomly generated large value, the system will pick up this pixel. Because the larger the value $i(x, y)$ is, the greater the value of m is, that pixel's probability of being selected is relatively high. Furthermore, the larger the value $i(x, y)$ is, the more points are in line with the conditions mentioned previously, and thus the more pixels are sampled because of their high luminance. In contrast, the smaller the value of $i(x, y)$ is, the lower the value of m is, and the probability of being chosen is relatively low for low values of $i(x, y)$. However, although the pixel value may be larger than the value of m, it is still possible that the pixel might be selected. The steps are as follows:

Step 1. If the brightness of a pixel is the highest value in the distributions of clustering, this pixel is likely to be the focus of attention points, and it has a high probability of belonging to the cluster.

Step 2. If the brightness of a pixel is not the highest value in the distributions of clustering, this pixel is not likely to be the focus of attention points, and it has a low probability of belonging to the cluster. Therefore, the brightness value of the pixel is set to the lowest value of the distribution map.

Through the aforementioned steps for the classification of the highlights that might attract attention in an image, one can split a single focus of attention while maintaining its integrity and the properties of the gradient.

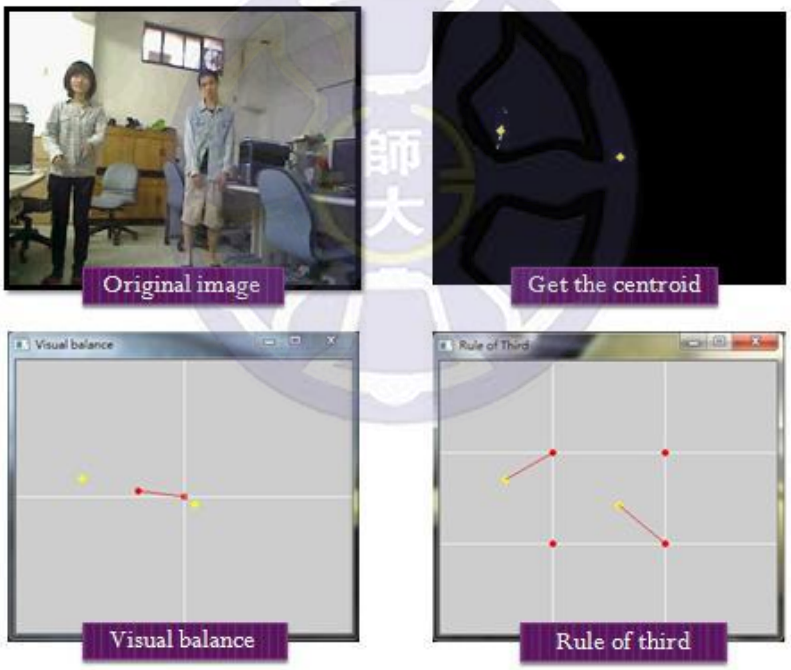


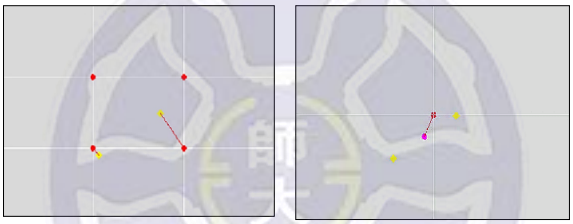
Figure 5.14. Attention points and aesthetic rules.

In our system, three VC shoot the speaker, audience, and hall. They find different salient objects from different angles. The system finds the salient objects in the attention map and then changes different features to extract them depending on the different aesthetic rules. After clustering, the system obtains the centroid of the salient object and then follows the aforementioned three rules to evaluate the aesthetic quality (see Figure

5.14).

E. Aesthetic score evaluation

The present research evaluates scores for three types of photographic composition rules: rule of thirds, visual balance, and size of salient object. These rules all require information on the salient object's position. In Figure 5.15, the yellow points are the centroids of the subject, and the red points are the "power points," which are crossed by two horizontal lines and two vertical lines in Figure 5.15(a). In addition, the red points in Figure 5.15(b) are at the center of the image. In Figure 5.15(a), the red line between the red point and the yellow point denotes the distance. With a shorter distance, the frame can obtain a higher rule of thirds score.



(a) (b)

Figure 5.15. The principle of aesthetic scoring (a) schematic of rule of third (b) schematic of visual balance.

The principle of scoring by the rule of thirds is: the shorter the distance is between the salient object and the power point, the higher the score is, but the converse does not hold. The representation of the score function is as follows:

$$S_{RT} = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{{D_M}^2(P_S - P_G)}{2\sigma^2}}, \tag{5.2}$$

where S_{RT} is the output score given by the rule of thirds. P_S is the centroid of the salient object. P_G is the nearest power point. $D_M(x)$ is the Manhattan distance.

Calculation of visual balance facilitates balancing the sense of weight from the image; we can use the following expression to calculate a visual balance score, S_{VB} :

$$S_{VB} = D_M(C, \frac{1}{n}\sum_{i=1}^{n} P_{S_i}), \tag{5.3}$$

where P_{S_i} is the position of the i th salient object. n is the number of salient objects. As the average positions of all items move closer to the center of the screen, the score becomes higher.

When shooting a presentation, the speaker occupies one-third of the screen for the most appropriate picture. The score of the size of the salient object is:

$$S_{Size} = \exp(-1.25(p - 0.3)^2), \quad (5.4)$$

where p is the ratio of the salient object in the saliency map.

5.1.2. Action analysis

A static picture selected by the director must be pleasant to look at, but it is even more essential that moving pictures be attractive to viewers. The movements of the cameraman can provide vital clues for analysis.

A. Camera movement and priority shot

Based on [78], we summarize three types of motions that are essential as a reference for shot selection of the VD system. The zoom-in camera movement usually means that a special event is happening. These types of shots are called "information shots." At this time, the higher priority of shot selection will be given to this shot. Conversely, the zoom-out camera movement means that the special event has concluded or there are multiple salient objects in a scene. These types of shots, called decorative shots, contain little information and are just used to fill (or "stuff") the empty space of the video when editing.

When the camera movement suddenly stops, it indicates some event might happen. Abdollahian [78] calls this type of shot "move-and-hold." At this time, the shot should be selected by giving it higher priority.

B. Optical-flow estimation

In this section, we describe how to use optical-flow estimation to detect camera motion and classify the results. In contrast to the discussion in Chapter 4, the optical-flow method is used here to represent camera motion rather than to detect ROI. The action analysis stage is focused on camera motion, because a moving camera can sometimes present an interesting shot. We use FAST corner detection to find the feature points and optical-flow estimation to estimate the general motion vector from every feature point. Therefore, the motion of a camera can be obtained.

To obtain the camera motion, after we use FAST to extract feature points (as shown in Chapter 4), the Lucas-Kanade optical-flow method (or hierarchical optical-flow estimation) is applied to estimate the motion vector of each feature point. Assume the object motion displacement between two frames is very small and does not change in the local area. Let p be a feature point on an image. Given a window of fixed size centered at p, the motion vector in the window can be maintained. In other words, the motion vector (V_x, V_y) satisfies the following：

$$f_x(q_i)V_x + f_y(q_i)V_y = -f_t(q_i), i = 1,2, \dots, n \tag{5.5}$$

where q_i is the point in the window from 1 to n. $f_x(q_i)$ and $f_y(q_i)$ are the partial derivatives at q_i in the horizontal and vertical directions. The aforementioned equation can be rewritten in a matrix form as:

$$Av = b, \tag{5.6}$$

where $A = \begin{bmatrix} f_x(q_1) & f_y(q_1) \\ f_x(q_2) & f_y(q_2) \\ \vdots & \vdots \\ f_x(q_n) & f_y(q_n) \end{bmatrix}$, $v = \begin{bmatrix} V_x \\ V_y \end{bmatrix}$, and $\mathrm{b} = \begin{bmatrix} -f_t(q_1) \\ -f_t(q_2) \\ \vdots \\ -f_t(q_n) \end{bmatrix}$.

This is calculated according to the minimum square error method $A^TAv = A^Tb$ or $v = (A^TA)^{-1}A^Tb$ to solve the equation $Av = b$. The results are given by substitution into the original formula:

$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum_i f_x(q_i)^2 & \sum_i f_x(q_i) f_y(q_i) \\ \sum_i f_x(q_i) f_y(q_i) & \sum_i f_y(q_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum_i f_x(q_i) f_t(q_i) \\ -\sum_i f_y(q_i) f_t(q_i) \end{bmatrix}. \quad (5.7)$$

(V_x, V_y) is the proposed optical-flow matrix.

C. Score evaluation of action analysis

After estimating the flow vectors, we assign direction labels to each feature point by the following formula:

$$\text{Label}_i = \begin{cases} pan & , \|v_x\| \neq 0, \ \|v_y\| = 0 \\ tilt & , \|v_x\| = 0, \ \|v_y\| \neq 0 \\ zoom & , \|v_x\| \neq 0, \|v_y\| \neq 0 \\ stable & , \|v_x\| = 0, \|v_y\| = 0 \end{cases}, \quad (5.8)$$

where V is the optical-flow vector. We calculate the norms of V with respect to the *x* component and the *y* component. The type of label most commonly found in one single view could present the type of camera motion of each frame. The VD is programmed to seek the frame after a zoom and a pan.

Next, count the numbers of the four labels. The most common numbers of direction labels in this frame can be used to represent the camera motion. Figure 5.16 and Figure 5.17 are examples of zoom-in and move-and-hold shots. The input sequences are listed at the tops of the figures. The normalized ratios of the direction labels are listed at the bottoms of the figures. When camera motions follow zoom-in, zoom-out, or move-and-hold actions, the VD system gives those camera motions higher priority. However, when a camera is moving, the VD should not select that camera's shot. Examples of the scores of camera motions can be observed in Figure 5.16. At the 35th frame, the ratio of radius labels is decreasing, whereas the ratio of stable labels is increasing. This phenomenon indicates that just after a zooming action is the best time to switch the shots, and it gives a score of 1 to the 35th frame. The score of a camera motion can be represented by:

$$S_{camera}(t) = \begin{cases} 1 & ,L(t) \neq L(t-1) \\ 0 & ,L(t) = L(t-1) \end{cases}, \tag{5.9}$$

where $S_{camera}(t)$ is the score of the t th frame. $L(t)$ and $L(t-1)$ are the direction labels of the t th frame and t-1 th frame, respectively. In other words, the points at which the camera suddenly changes the direction of its motion are given higher scores.

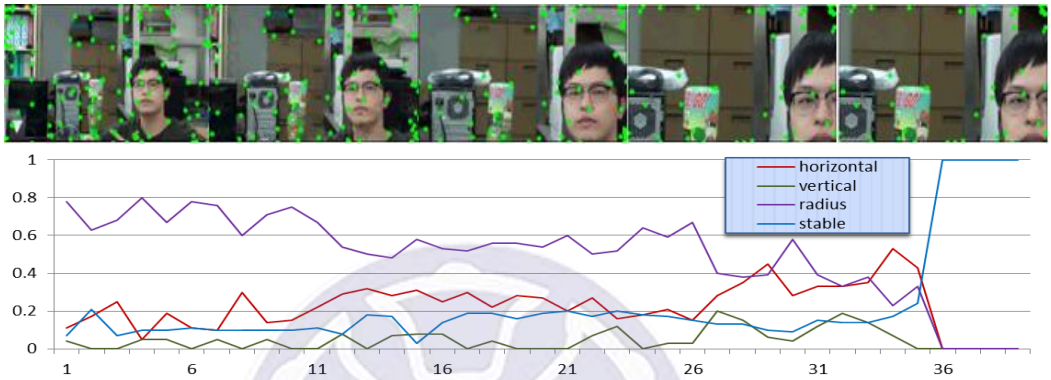


Figure 5.16. Zoom-in example. The input sequence is shown at the top. The normalized ratios of the direction labels of the four directions are graphed on the bottom.

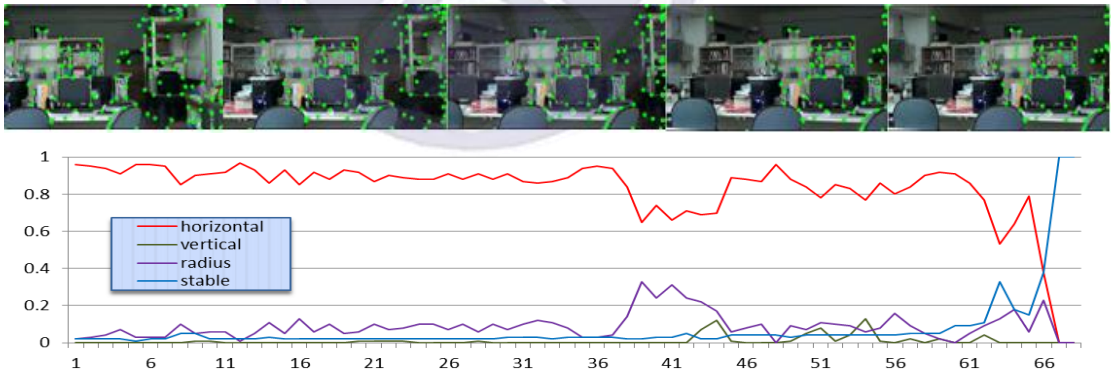


Figure 5.17. Move and hold example. The input sequence is shown at the top. The normalized ratios of the direction labels of the four directions are graphed on the bottom.

5.1.3. Continuity analysis

Aside from shot selection, a director must prevent viewer discomfort by presenting smooth and coherent shot changes. Four continuities should be considered: illuminance continuity, color continuity, scenery continuity, and position continuity.

A. Illuminance and color continuity

The goal of continuity analysis is to choose the frame connected with the broadcast frame for the smoothest possible appearance. We consider illuminance continuity and color continuity. The color continuity evaluation is:

$$S_{CC} = \frac{1}{k}\sum_{j=1}^{k} \frac{[H_h(S_{i,j}(t)) - H_h(S_j(t-1))]^2}{H_h(S_{i,j}(t)) + H_h(S_j(t-1))} \tag{5.10}$$

where k is the bin number of the hue histogram. i is number of candidate shots. $H_h(S_{i,j}(t))$ is current value of hue histogram at bin j of candidate shots. $H_h(S_j(t-1))$ is value of hue histogram at bin j of chosen shot $S(t-1)$ at time $t-1$. Just similar to color continuity, the illuminance continuity evaluation is:

$$S_{CI} = \frac{1}{k}\sum_{j=1}^{k} \frac{[H_v(S_{i,j}(t)) - H_v(S_j(t-1))]^2}{H_v(S_{i,j}(t)) + H_v(S_j(t-1))}. \tag{5.11}$$

Only difference is using value histogram instead of hue histogram.

We construct the intensity histogram and hue histogram of each frame in the HSV color space and then estimate the chi-squared distance between the histogram of the broadcast view and the histogram of the candidate view. The greater the distance of the candidate view is, the lower its score is.

B. Scenery continuity

Scenery refers to the distance between the camera and the object being shot. As Figure 5.18 shows, a shot with long scenery (known as a "long shot"), which contains considerable information, is usually used to describe the temporal relationship between the salient object and the background. A shot with shorter scenery (known as a "close shot"), which contains more detail on the local features of the salient object, is usually used to emphasize dramatic tension or a theme. Because different scenery lengths have

their own uses and characteristics, appropriate scenery shifting can add considerable depth of detail to a story.

Figure 5.18. Scenery shots with different lengths.

However, excessively abrupt shifts of scenery often cause viewer discomfort. Usually, longer scenes have more information, and shorter scenes have less information. Thus, we calculate the color variance in the color space frame by frame. When the color variance is higher, it means the scenery is longer; when the variance is lower, it means the scenery is shorter. Scenery continuity can be defined by the difference between the variances of the frame:

$$S_{CS} = Var\big(S_i(t)\big) - Var\big(S(t-1)\big) \tag{5.12}$$

where $S(t-1)$ is the shot selected by the VD subsystem at time $t-1$. $Var(S_i)$ is the variance of candidate shot S_i at time t. Hence, we use the spatial-color distribution to estimate the scenery continuity. The formulas are:

$$Var(S) = \frac{1}{2C}\sum_c (Var_h(S_c) + Var_v(S_c)), \tag{5.13}$$

$$Var_h(S_c) = \frac{1}{N}\sum_P (P_x - M_h(S_c))^2 \times S_c(P), \tag{5.14}$$

$$Var_v(S_c) = \frac{1}{N}\sum_P (P_y - M_v(S_c))^2 \times S_c(P), \tag{5.15}$$

$$M_h(S_c) = \frac{1}{N}\sum_P P_x \times S_c(P), \tag{5.16}$$

$$M_v(S_c) = \frac{1}{N}\sum_P P_y \times S_c(P), \tag{5.17}$$

where $Var(S)$ is the spatial-color variance, which is combined with the horizontal variance and the vertical variance of shot S. Equations 5.14 and 5.15 estimate the

scattered degree of each color in a single image; i.e., Equations 5.14 and 5.3 calculate the variance of color C in shot S. P_x and P_y are the intensity values of pixel P in the x and y coordinates. M_h and M_v are the means of the spatial-color distribution in the horizontal and vertical directions. If the $Var(S)$ rises, the scene is possibly in close-up view. If the $Var(S)$ declines, the scene is possibly a hall view, because the shot contains more information.

C. Position continuity

To avoid long-distance movements of viewers' eyes, the distances between different salient objects during a shot change should not be large. After a long period of video viewing, viewers are prone to fatigue. For position continuity, the score function is as follows:

$$S_{CP} = d\left(P\left(S_i(t)\right), P\left(S(t-1)\right)\right), \tag{5.18}$$

where i is number of candidate shots. $P(S_i(t))$ is the salient object position of candidate shot S_i at time t. $P(S(t-1))$ is the salient object position of chosen shot $S(t-1)$ at time $t-1$. $d(x)$ is the Euclidean distance function.

5.1.3. Optical analysis

Today, a growing number of built-in subsystems in video cameras automatically optimize features such as automatic exposure compensation, automatic white balance, color adjustment, auto focus, and so on. While these automatic optimization functions sometimes are needed because of changes in the environment, they may also be adjusted improperly and cause discomfort for the viewer. For example, it the speaker suddenly moves, the camera may automatically adjust the focus, and the sudden movement may cause a brightness readjustment. Alternatively, intensity changes from

a slideshow change may cause the camera to perform auto exposure and auto white balance adjustment of hue. These features are automatically optimized, but they can easily produce audience discomfort. We therefore present a VD system that can avoid choosing those shots in the aforementioned situations.

The common evaluation methods, such as exposure, saturation, and sharpness, are also essential, whether a salient object is in the image or not. Images saturated with different colors can suggest different moods; for example, relatively low saturation can show nostalgia. If you want a more fun and vibrant image of the performance, you can use high color saturation. Therefore, we define three types of optical analysis methods: saturation, exposure, and sharpness.

A. Saturation

For the proposed lecture recording system, we want to show a neutral and formal impression to viewers. Thus, both shots that lack tone color and oversaturated shots are unsuitable. A shot with moderate saturation is most suitable (see Figure 5.19).

Figure 5.19. Saturation of an image.

For an input image, we build a saturation histogram and intensity histogram in the HSV color space. The saturation score function is defined as:

$$S_{SA} = \frac{1}{m}\sum_{i=1}^{m} \frac{s_i}{\sigma\sqrt{2\pi}} e^{-\frac{(s_i-\bar{s})^2}{2\sigma^2}}, \tag{5.20}$$

where S_{SA} is the score for saturation. m is the bin number of the saturation histogram. s_i is the saturation value at bin i. In addition, $\bar{s}$ is the average saturation of the entire image. By utilizing the Gaussian function, the VD can give higher weight to an image with a normal saturation value and give lower weights to images with undersaturation and oversaturation.

B. Exposure

Similarly, neither underexposure nor overexposure should be chosen. The VD is programmed to select shots with moderate exposure (see Figure 5.20). By giving a higher score to the moderate exposure shot, the score function is written as:

$$S_{EX} = \frac{1}{k}\sum_{i=1}^{k} \frac{v_i}{\sigma\sqrt{2\pi}} e^{-\frac{(v_i-\bar{v})^2}{2\sigma^2}}, \tag{5.21}$$

where S_{EX} is the exposure score. k is the bin number of the intensity histogram. v_i is the intensity at bin i. $\bar{v}$ is the average intensity of the entire image.

Figure 5.20. Exposure example of an image.

C. Sharpness

As shown in Figure 5.21, when the camera is moving, it inevitably results in

blurring or a lack of focus. These can cause the salient object to appear vague, whether the background is clear or not. The system should try to avoid selecting shots with blurred salient objects. The ROF map is proposed here to detect regions with higher sharpness values.

Figure 5.21. Three examples of sharpness in images.

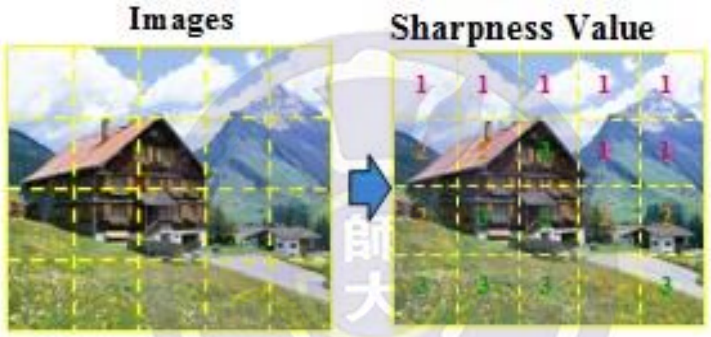


Figure 5.22. Detection of sharpness.

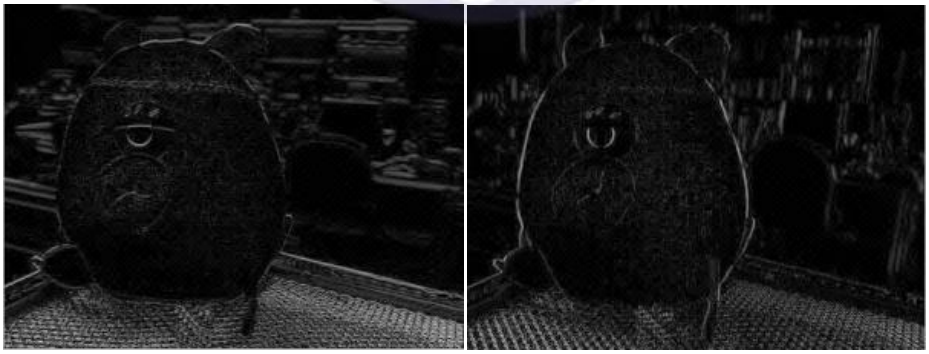


Figure 5.23. Gradient map (left) horizontal direction (right) vertical direction.

Generally, in sharpness detection, to identify the clear region of an image, the clear region shows numerous details, whereas the obscure region has fewer details (Figure 5.22). First, convert the input image to a gray-scale image, and then perform Gaussian

blur filtering to get a blurred gray-scale image. Make horizontal and vertical gradient maps from the gray-scale image and blurred gray-scale image, respectively (Figure 5.23). Subtract the two gradient images in the individual horizontal and vertical directions (Figure 5.24) to get the horizontal and vertical detail maps. Finally, combine the horizontal and vertical detail maps to get the ROF map.

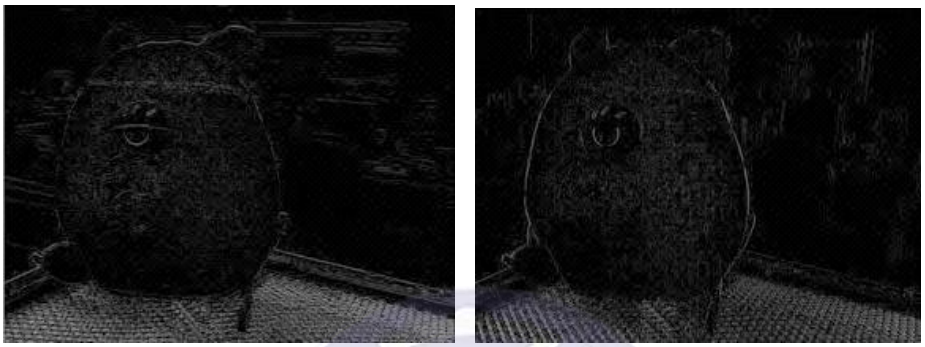
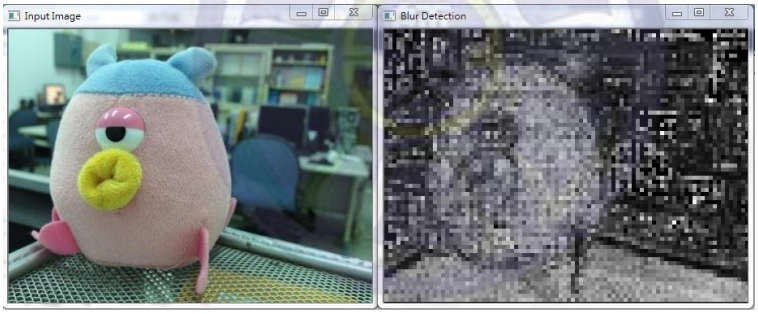
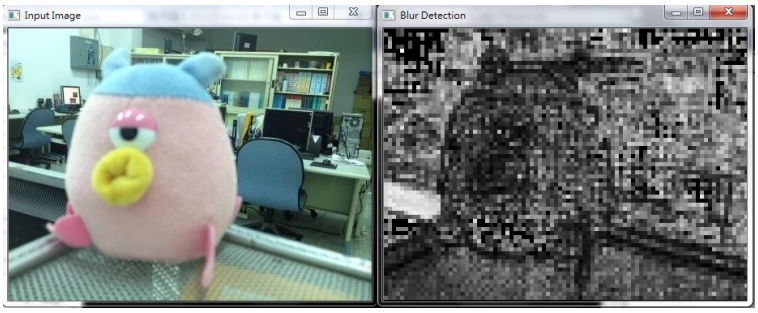


Figure 5.24. Detail map (left) horizontal direction (right) vertical direction.



(a)



(b)

Figure 5.25. ROF map (a) blurred background (b) clear background.

Usually, the regions on which the camera focuses have more detail, so we produce an ROF map to find the portions of the image that have a clear subject. We evaluate the

optical score by combining saliency map *S* and ROF map *F*,

$$S_{SH} = \frac{1}{n}\sum_{i=1}^{n}(F_i \times S_i), \tag{5.22}$$

where F_i is the intensity value at pixel i on the ROF map. S_i is the intensity value at pixel i of the saliency map. Figure 5.25 shows the ROF map between different focus positions. Figure 5.26 shows an example that gets a high score of optical analysis from a comparison of the saliency map and the ROF map.

Figure 5.26. Saliency map and ROF map.

5.2 Shot Selection

Shot selection is a type of decision-making problem. Thus, after assigning evaluation scores to each view, the system uses the trained counter propagation network (CPN) [62] neural network to decide which shot to broadcast. When the shot selection stage was in training procedure, the CPN network learned the skill of shot selection from real-world directors. The input to the CPN network consisted of the scores estimated in the content analyzing stage; the range of those scores was [0, 1]. We invited a professional who had experience as a program director to provide the expected output for the training data.

However, input data may have different score ranges and different definitions of desirable and undesirable. To avoid this heterogeneous data interference in the convergence of the result of the neural network, we introduce the concept of multiple

kernel learning (MKL) [63, 64] to transform the data space to a homogeneous representation. The details of CPN and MKL can be found in Chapter 2.

The VD system performs shot selection from multiple shots by CPN machine learning. In this section, we discuss details of how to use CPN to construct a shot selection model and how to apply the model to a VD system. In the first section, we compare the shot selection of a real director and the VD. Next, we discuss the obstacle of heterogeneous input data and how MKL can be applied to integrate the heterogeneous data, which helps to improve shot selection performance. Finally, we describe how to apply CPN to shot selection.

5.2.1. Shot selection of real director and virtual director

When recording a live speech or a program, the role of the director is vital. However, it is extremely difficult to hire a highly experienced and professional director. In general, those who want to become directors must learn photography and video editing. A good director must have the ability to react immediately, lead others, and be perceptive. To acquire these capabilities, a learner must start from the bottom. After accumulating some experience, the next step is to follow an experienced director to learn how to command cameramen and perform shot selection. Initially, learners start with imitation learning, sharpen skills to proficiency, and then slowly add their own styles; those who succeed become outstanding directors. The shot selection of a director is not based on definite, cut-and-dried criteria, and the learning process is slow, so a director only forms his own selection rules after repeated operations, gradual exercises, and self-teaching. The rules known to an individual director evolve into a "rule library."

The CPN network was first mentioned by [62]. This type of network is used to approximate some functions or memory problems (i.e., when the relationship between input and expected output cannot be represented by a simple linear function). At the

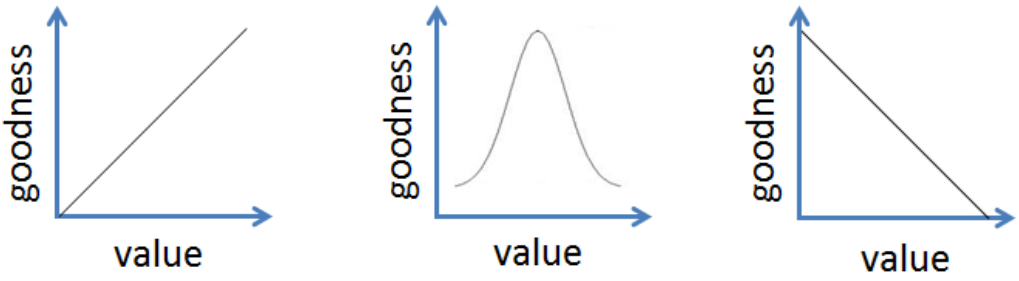
training stage, the CPN adjusts its structure to “memorize” the relationship between the input and labeled answer. The altered structural relationships can be seen as a new rule. These rules are accumulated into a “rule library” one by one. After the library is built, when new input comes, the network automatically chooses the most similar rule and provides output based on the rule.

Both the CPN and a real director use existing information to form their own rule libraries, and both types of rule libraries are accumulated, one rule at a time. Moreover, the training strategies both belong to supervised learning. At testing time, both of them use pretrained rules to perform shot selection. The proposed shot selection method implemented by the CPN matches the operational model of a real-world director exactly. That is why the CPN is our shot selection method.

5.2.2. Heterogeneous data and multiple kernel learning

The VD system uses content analysis to assess image quality at the shot selection stage. Previous sections of this test summarize nine score evaluation guidelines: (1) rule of thirds, (2) visual balance, (3) camera motion, (4) sharpness, (5) exposure, (6) saturation, (7) illuminance continuity, (8) color continuity, and (9) scenery continuity. Moreover, two scores (size of salient object and position continuity) are not part of the input, because both of them can be covered by some other assessment (e.g., size of salient could be included in scenery continuity). However, the outputs of the aforementioned nine score assessments are all heterogeneous data with different ranges. Depending on the goodness definition, three types of data distributions may be relevant (see Figure 5.27). (1) The larger the score is, the higher the candidacy probability is for the shot selection stage (e.g., rule of thirds, visual balance, camera motion, and sharpness). (2) A score range in the middle is favorable (like Gaussian distribution); scores that are too high or too low are undesirable (e.g., exposure and saturation). (3)

The lower the score is, the better it is (e.g., illuminance continuity, color continuity, scenery continuity, for all of which, a lower score means continuity is higher).



(a) (b) (c)

Figure 5.27. Three definitions of score goodness

(a) first type (b) second type (c) third type.

During the shot selection stage, the scores of three shots (from SC, AC, and HC) are provided by the score assessment function, and the VD chooses one of them by using a machine learning-based classifier. As mentioned previously, each of the nine scores has a different meaning and range. It is necessary to control the accuracy of the classifier that handles such heterogeneous data. Obviously, the traditional normalization method is not sufficient, because the likelihood measurement of winner-take-all stage in the CPN network is easily dominated by the features of the data ranges, so traditional normalization would lower the accuracy of the shot selection.

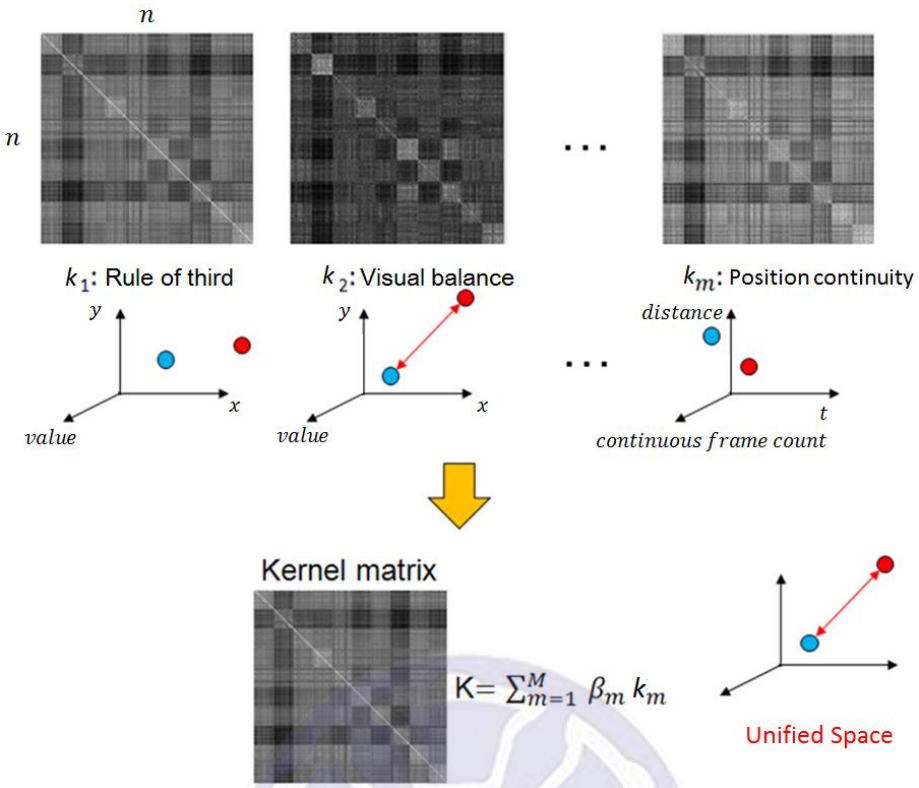


Figure 5.28. MKL implementation.

Figure 5.28 shows the implementation of MKL in the proposed VD system. Assume the system has n data points of labeled shot selection data; assume each data point contains nine scores from content analysis and the shot selection result is labeled by a real director.

In the conventional method, the kernel coefficient could be obtained from machine-learning approaches, such as PCA or SVM. However, considering the fact that there are nine scores and the system must be scalable for multiple cameras, methods such as PCA and SVM would introduce too many complexities and difficulties for a real-time classifier. Therefore, we utilize MKL and find kernel coefficient β_i by a neural network method instead of solving a time-consuming optimization problem. Explicitly, we use the CPN to solve the optimization problem and obtain the weight from the training sample. In this way, the system can converge quickly and easily.

Our approach has nine assessment scores, so it is necessary to design a different

kernel function for each score to obtain favorable mapping results. After kernel transformation, the nine scores from the individual analysis method have the same definition in the likelihood assessment.

First, we build a kernel function for every score:

- Rule of thirds: $K_{RT}(S_{RT}) = x \times 255$ (5.23)
- Visual balance: $K_{VB}(S_{VB}) = x \times 255$ (5.24)
- Camera motion: $K_{CA}(S_{CA}) = x \times 255$ (5.25)
- Sharpness: $K_{SH}(S_{SH}) = x \times 255$ (5.26)

The aforementioned four methods can be classified as class 1, which means a shot with a larger score would have a higher probability of being chosen as the best shot. The kernel function uses multiplication to transform the scores to the appropriate range.

- Exposure: $K_{EX}(S_{EX}; \mu, \lambda) = \left[\frac{\lambda}{2\pi x^3}\right]^{1/2} \exp\frac{-\lambda(x-\mu)^2}{2\mu^2}$, for $x > 0$ (5.27)
- Saturation: $K_{SA}(S_{SA}; \mu, \lambda) = \left[\frac{\lambda}{2\pi x^3}\right]^{1/2} \exp\frac{-\lambda(x-\mu)^2}{2\mu^2}$, for $x > 0$ (5.28)

The aforementioned two methods can be classified as class 2, which means the shots with medium scores have relatively high probabilities of being chosen as the best shot. The kernel function uses the inverse of the Gaussian distribution function to transform the scores to the appropriate range.

- Illuminance continuity: $K_{CI}(S_{CI}) = (1-x) \times 255$ (5.29)
- Color continuity: $K_{CC}(S_{CC}) = (2-x) \times 255$ (5.30)
- Scenery continuity: $K_{CS}(S_{CS}) = (1-x) \times 255$ (5.31)

The aforementioned three analysis methods can be classified as class 3, which means the shots with lower scores would have a higher probability of being chosen as the best shot. The kernel function subtracts scores from a constant value and multiplies the results to transform the scores to the appropriate range.

After kernel transformation, the range of scores is mapped to [0, 255], where 255

means good and 0 means bad (see Figure 5.29).

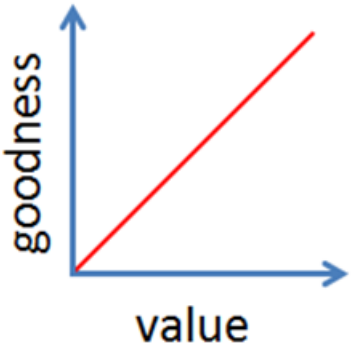


Figure 5.29. Score definition after kernel transformation.

As shown in Figures 5.30 to 5.34, the image result is the visualized kernel matrix of 155 training data points sent from the SV. Each data point contains nine different scores. Each element of the kernel matrix is the inner product of a pair of data points. If the element value is 255, it means the two data points of that pair are very similar. In other words, the score of a shot might be similar to that of another shot; if the element value is 0, it means the two data points are not similar. In other words, the score of one shot might be not at all similar to the score of another shot.

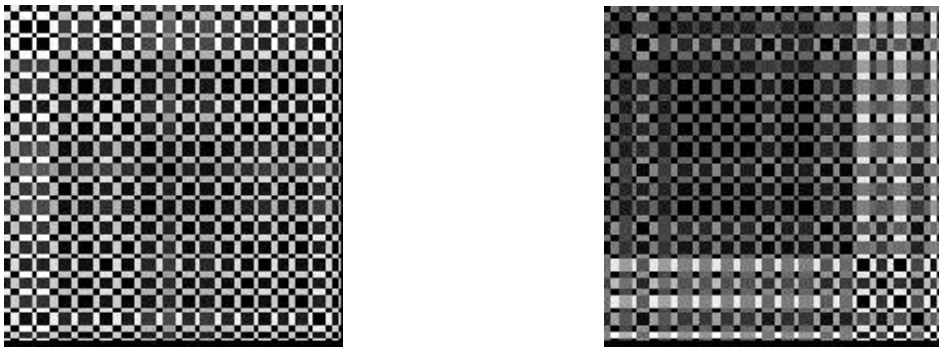


Figure 5.30. Kernel matrix (left) rule of third (right) visual balance.

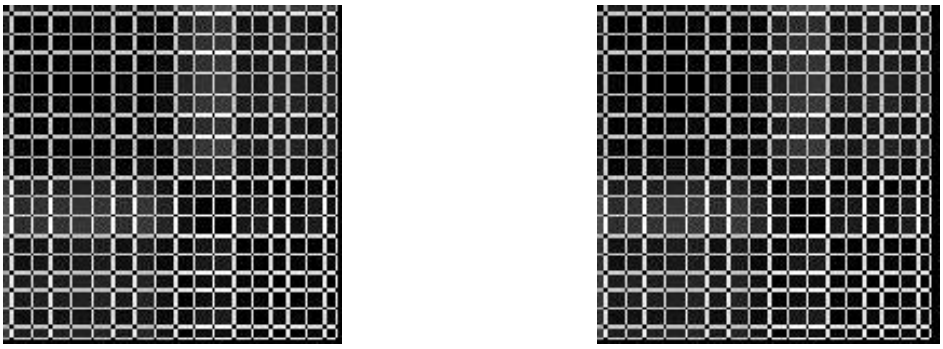


Figure 5.31. Kernel matrix (left) sharpness (right) exposure.

Figure 5.32. Kernel matrix (left) saturation (right) illuminance continuity.

Figure 5.33. Kernel matrix (left) color continuity (right) Scenery continuity.

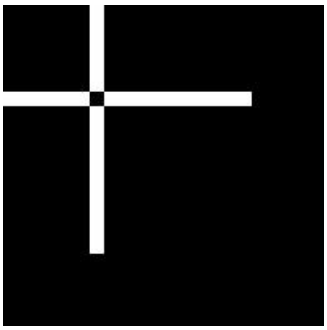


Figure 5.34. Kernel matrix of camera motion.

5.2.3. Apply counter propagation neural network to VD

As described in Chapter 2, the CPN network is suitable for shot selection. Because of the hidden layer, a neuron can be seen as a class or as a rule. The connections between the hidden layer and the input layer and output layer can be regarded as a complete description of the rules; that is, the network operation can be described as follows: "If x is w, then y is u."

The forward-mapping CPN network makes decisions for the VD. The inputs of the system are the scores from the content analysis; each shot has nine scores. Thus, a total of 27 input data points arrive at the VD system for each shot that is selected. The corresponding output value $y_1^*, y_2^*, \ldots, y_m^*$ indicates the selected shot. Only one of the output values is 1, whereas the others are 0. The neurons at the hidden layer can be mapped to specific conditions at a lecture (e.g., one of the audience members raises a hand, the speaker is pointing somewhere, etc.). Figure 5.35 shows the training model of the CPN network. After training the model, the scores obtained from the content analysis stage enter the CPN testing model. Then the final chosen view from the trained model can be used. Figure 5.35 illustrates the network architecture of the MK-CPN Network.

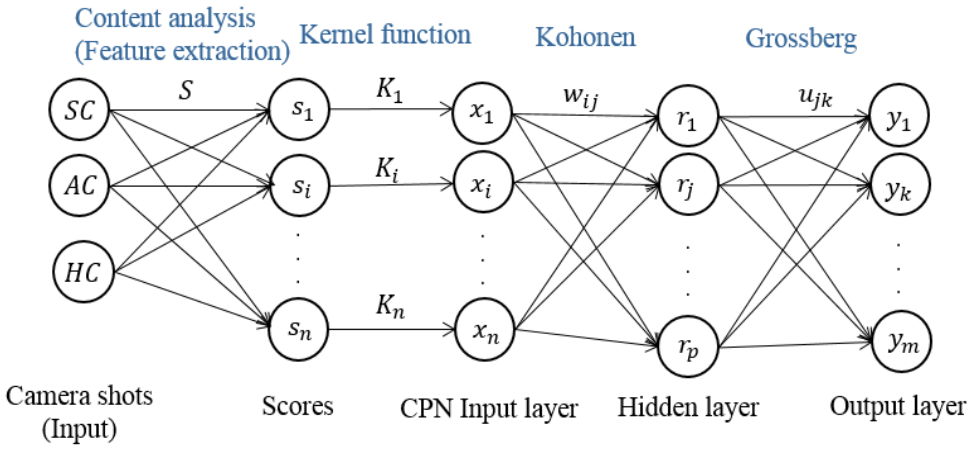


Figure 5.35. Training model of CPN.

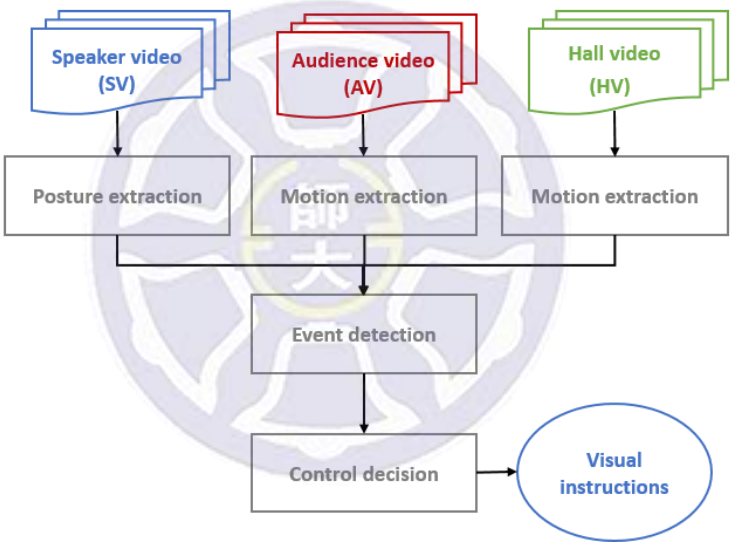


Figure 5.36. Flowchart of visual instruction.

5.3 Visual Instruction

In this section, we introduce the visual instruction from the VD to the VCs, and discuss the points in time at which the VD must give visual instructions. See Figure 5.36, the VD simultaneously receives videos from three cameramen: SV, AV, and HV. The VD gives visual instruction through event definition at a reasonable time. The

events shot by the SV are defined by speaker gestures (see Chapter 3). The events of the AV and HV are defined by the scales of movements in their scenes.

The movement area detection is based on the STA image described in Section 4.1. Unlike the method in Section 4.1, we want to detect the scale of movement in a scene instead of the center of the salient object. Entropy is applied here to measure the scale.

Entropy is calculated block by block for the attention map. By combining the entropy values of the adjusting blocks, we can obtain the scale of movement. The block size we used is 10 pixels × 10 pixels (see Figure 5.37).

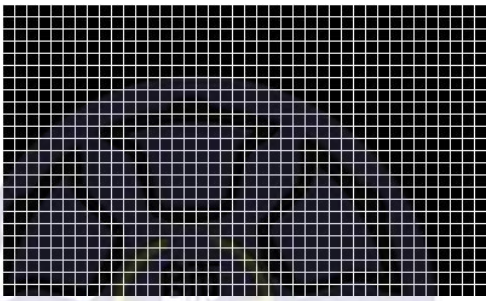
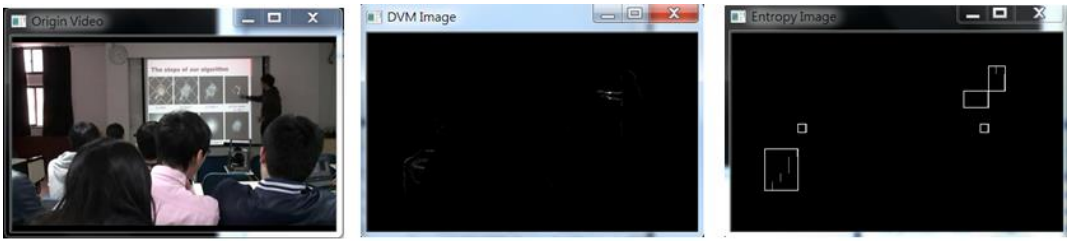


Figure 5.37. Blocks of 10 pixels × 10 pixels.

A histogram that depends on the number of gray levels is built for each block. Then normalization is performed to convert the summation of the probability to 1. In addition, the entropy value is recalculated by following function:

$$E = -\sum_{i=0}^{255}(p_i \log p_i) \tag{5.32}$$

where p_i is the probability value at pixel i. The entropy E is a measure of the unpredictability of information content. The larger the value of entropy is, the larger the degree of clutter in the block is, and vice versa. After that, we merge the blocks with the similar entropy values. We call the new area the movement area; an example is shown in Figure 5.38.



(a) (b) (c)

Figure 5.38. Movement area detection (a) original image (b) attention map (c) blocks after merging.

As shown in Table 5.2, the scale of motion is defined by the largest entropy value E_{max}. If E_{max} is larger than 1, a large motion is detected; otherwise, it is a small motion. The range of movement is defined by the diagonal of the bounding box $R_{diagonal}$ of the blocks after merging. If the diagonal is larger than 70 pixels, a big range is detected; otherwise, it is a small range.

Table 5.2. Definition of the movement.

	$E_{max} \geqq 1$	$E_{max} < 1$
$R_{diagonal} \geqq 70$	(Big motion, Big range)	(Small motion, Big range)
$R_{diagonal} < 70$	(Big motion, Small range)	(Small motion, Small range)

According to Chapter 3, speaker events defined by postures are divided into three types: pointing, illustrating, and relaxing. Audience events defined by movement can be divided into four types: (Big motion, Big range), (Big motion, Small range), (Small motion, Big range), and (Small motion, Small range). Like the audience event, the event of the hall is also defined by movement and divided into four types: (Big motion, Big range), (Big motion, Small range), (Small motion, Big range), and (Small motion, Small range). Therefore, the system can recognize a total of 3×4×4=48 types of events.

Once an event is detected, the VD gives a visual instruction that depends on the event's type. The corresponding visual instructions are listed in Appendix (see Table A.2, Table A.3, and Table A.4). For example, when there is an event of type 16, the speaker might be pointing to something in the slideshow, and the audience might be focused on the lecture. The VD commands the SC to zoom in on the pointing area and commands the HC to pan repeatedly and slowly. Consider another example: When the event is of type 40, the speaker might be waiting for something, and the audience might experience a disturbance. The VD commands the AC to zoom out for more information about the audience.

Because the three VCs have their own missions and targets, only the VD receives all pictures and integrates all VC information from a broad perspective. The communication between the VC and the VD improves the quality of the lecture recording. The SC reports the posture of the speaker to the VD. The VD judges the events of the AC and the HC through the AV and the HV. Most VDs cannot give visual instructions as human directors do in the real world. Our VD system performs shot selection in a manner superior to other VDs, because our VD can also give visual instruction. The VCs have their own shooting rules, but once a cameraman receives a visual instruction from the VD, there may be a conflict between the instruction and its own shooting rules. In this case, the cameraman is programmed to obey unconditionally; it considers the instruction from the VD to take precedence over its own rules, just like the authority of a real director takes precedence over the notions of real cameramen.

Chapter 6
Virtual-Real Match Moving

In this chapter, a virtual-real synthesis image composition method, called match moving, is presented. This method is primarily designed for graphic image composition and stereoscopic image composition. The input data to this method is assumed to be provided by the equipment consisting of a color camera and a depth camera. Figure 6.1 shows the configuration of this equipment, where the depth camera is mounted on the color camera. Note that these two cameras need not be facing toward the same direction for filming the same object. We refer to the system including both software and hardware components as the virtual-real match moving (VRMM) system.

Figure 6.1. The depth camera is mounted on a color camera.

The rest of this chapter is organized as follows. Section 6.1 discusses the workflow of the VRMM system. There are three major processes: temporal depth fusion, camera tracking, and virtual-real synthesis preview, involved in the VRMM system. These three processes are detailed in Sections 6.2, 6.3 and 6.4, respectively.

6.1 Workflow of the VRMM System

Figure 6.2 shows a flowchart of the VRMM system. To begin, the depth images acquired by the depth camera are fused through time to lead to a 3D construction of the

scene. Based on the information of the 3D scene, the pose of the depth camera is easily determined. We assume that the spatial relationship between the depth camera and the color camera is known a priori. The pose of the color camera can hence be figured out from that of the depth camera. Continuing the above process, a sequence of poses of the color cameras and in turn its trajectory can be obtained. This completes the camera-tracking process. The attained trajectory of the color camera will direct a virtual camera to generate synthetic images of a given 3D object model. The generated images are next superimposed upon the real images acquired by the color camera. We call this process the virtual-real synthesis image composition. The resultant images are called preview images.

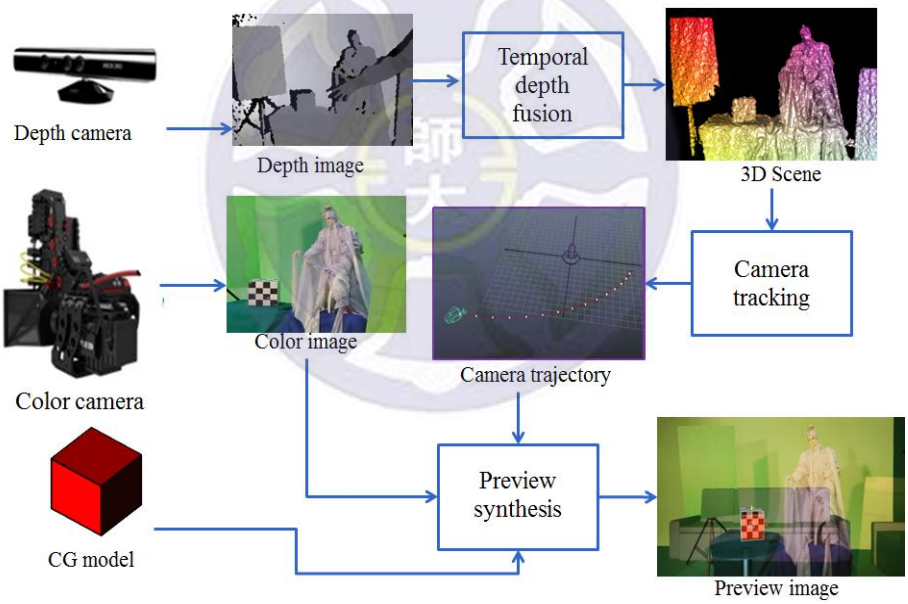


Figure 6.2. A flowchart of VRMM method.

As mentioned, there are three major processes: temporal depth fusion, camera tracking, and virtual-real synthesis preview, involved in the VRMM system. These three processes are discussed in depth in the subsequent sections, separately.

6.2 Temporal Depth Fusion

During temporal depth fusion [79], the depth images acquired by the depth camera are fused through time to lead to a 3D construction of the scene. There are four steps consisting of the temporal depth fusion process: i) converting depth image to local vertex and normal map, ii) finding global vertex and normal map from local one, iii) determining view matrix, and iv) 3D updating. Figure 6.3 graphically illustrates these four steps, which are detailed in the following.

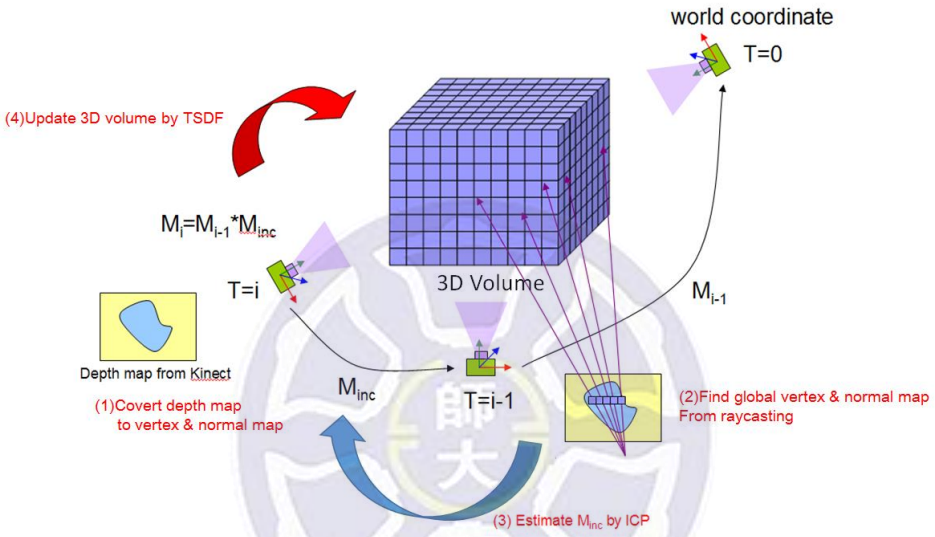


Figure 6.3. Steps of the temporal depth fusion process.

Step 1. Converting depth image to local vertex and normal map:

In this step, the pixels of the 2D depth image acquired by the depth camera at time T_i is first represented as points in the 3D camera coordinate system. A set of vertices and normals of scene surfaces is next derived from the 3D points, which forms a vertex and normal map. We refer to this map as the local map because it is described with respect to the local camera coordinate system. Thereafter, if $T_i = T_0$, go to step 4.

Step 2. Finding global vertex and normal map from local one:

In this step, the local vertex and normal map obtained in the previous step is projected using a ray-casting method onto a discrete 3D volume represented in a real world coordinate system, which is defined as the camera coordinate system at time T_0.

The projected map is hence referred to as the global vertex and normal map. The objective of this step is twofold. First, it preserves multiple 2D views in one finite 3D volume so as to reduce the storage of multiple views. Second, every entry of the 3D volume will progressively increase accuracy due to averaging multiple values projected from different views.

Step 3. Estimating global transformation matrix of view:

In this step, the global transformation matrix M_i of the current view at time T_i is to be estimated with respect to the real world coordinate system by way of $M_i = M_{i-1} * M_{inc}$, where M_{i-1} is the global transformation matrix of the immediately previous view at time T_{i-1}, and M_{inc} is the transformation matrix between the current view and the previous view. M_{i-1} has been known from the preceding steps, whereas M_{inc} is estimated in this step. To estimate M_{inc}, the 3D shape alignment algorithm [80] characterized by an iterative closest point (ICP) process is applied to the global vertex and normal maps of the current view and the previous view.

Step 4. 3D Updating:

In the last step, the entries of the 3D volume corresponding to the projection of the global vertex and normal map of the current view at time T_i are modified according to the projected values using a truncated signed distance function (TSDF). As a result, the entries of the 3D volume would progressively increase accuracy with time. Figure 6.4 shows an example of 3D updating through time. With iterations, the 3D construction of the scene becomes smoother and more accurate.

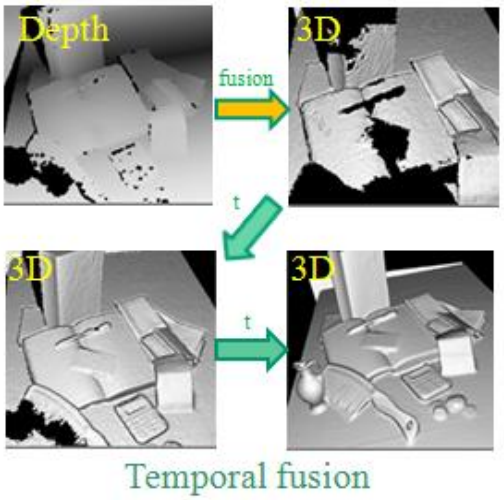
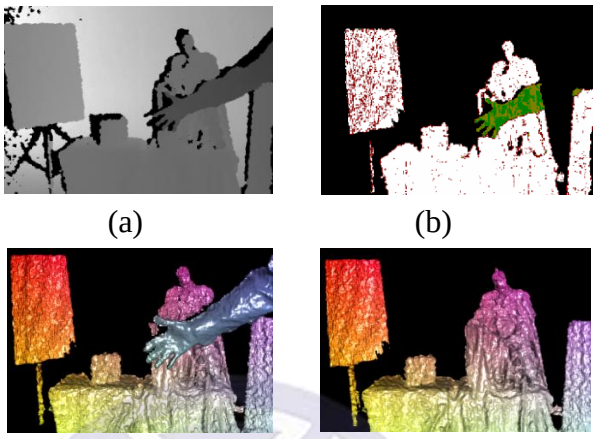


Figure 6.4. Example of 3D updating.

However, the above temporal depth fusion process assumes that the scene under consideration is static. If there are moving objects present in the scene, the precision of the 3D construction of the scene would be adversely affected and as a result influence the accuracy of the predicted trajectory of the color camera. To compensate for this drawback, our strategy is to detect and remove moving objects in advance from each input depth image to the temporal depth fusion process.

First of all, we use the spatial-temporal attention (STA) neural network [31] detailed in Chapter 4 to detect dynamic areas in the input depth image, which may correspond to moving objects. Note that in our application moving objects are primarily humans. We hence introduce a human skeleton model to facilitate to locate humans in dynamic image areas. See the example shown in Figure 6.5, in which the input depth image, the located dynamic region, the 3D construction of the scene with a moving hand, and the 3D construction of the scene without the hand are indicated. Having detected dynamic regions in a depth image, we next locate humans in these regions. Figure 6.6 shows an example of human detection based on human skeleton model, in which a located human in the input depth image, the corresponding color image, the 3D construction of the scene containing the human, and (d) the 3D construction of the

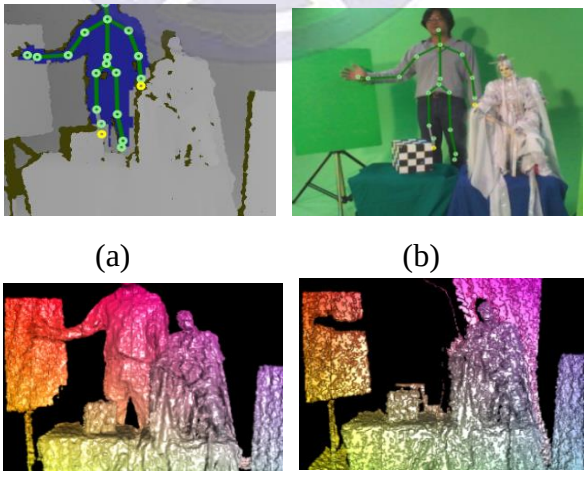
without the human are displayed. This example is specialized for addressing the condition where there are characters or actors interacting in the filming scene and the error caused by a sudden move from an originally motionless character in the scene or a moving character in the scene suddenly becoming still.



(a) (b)

(c) (d)

Figure 6.5. Locating dynamic regions using the STA neural network (a) the input depth image (b) the located dynamic region (c) the 3D construction of the scene with a moving hand (d) the 3D construction without the hand.



(a) (b)

(c) (d)

Figure 6.6. Human detection based on human skeleton model (a) a located human in the input depth image (b) the corresponding color image (c) the 3D construction of the scene containing the human (d) the 3D construction of the without the human.

6.3 Camera Tracking

The proposed camera tracking method can be self-positioning. After temporal depth fusion, the transformation matrix *P* of the depth camera with respect to the real world coordinate system is known. See Figure 6.7, which illustrates the framework of the sensing device shown in Figure 6.1. Let *M* denote the transformation matrix from the color camera to the depth camera, which is known a priori. The transformation matrix *P'* of the color camera with respect to the real world coordinate system can then be determined by:

$$P' = P \times M. \tag{6.1}$$

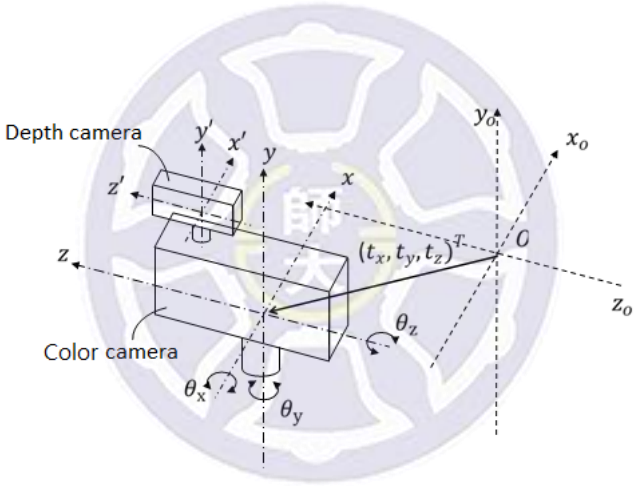


Figure 6.7. The configuration of the sensing device.

Let

$$P=\begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \quad M=\begin{bmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

$$P'=\begin{bmatrix} r'_{11} & r'_{12} & r'_{13} & t_x \\ r'_{21} & r'_{22} & r'_{23} & t_y \\ r'_{31} & r'_{32} & r'_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{6.2}$$

in which $r'_{11} = cos(\theta_y)cos(\theta_z)$, (6.3)

$$r'_{12} = sin(\theta_x)sin(\theta_y)cos(\theta_z) + cos(\theta_x)sin(\theta_z), \tag{6.4}$$

$$r'_{13} = -cos(\theta_x)sin(\theta_y)cos(\theta_z) + sin(\theta_x)sin(\theta_z), \quad (6.5)$$

$$r'_{21} = -cos(\theta_y)sin(\theta_z), \quad (6.6)$$

$$r'_{22} = -sin(\theta_x)sin(\theta_y)sin(\theta_z) + cos(\theta_x)cos(\theta_z), \quad (6.7)$$

$$r'_{23} = cos(\theta_x)sin(\theta_y)sin(\theta_z) + sin(\theta_x)cos(\theta_z), \quad (6.8)$$

$$r'_{31} = sin(\theta_y), \quad (6.9)$$

$$r'_{32} = -sin(\theta_x)cos(\theta_y), \quad (6.10)$$

$$r'_{33} = cos(\theta_x)cos(\theta_y) \quad (6.11)$$

where $(\theta_x, \theta_y, \theta_z)$ are the rotation angles of the color camera about the *x*-, *y*-, and *z*-axes, respectively, and $(t_x, t_y, t_z)^T$ is the translation vector of the color camera. The rotation angle can be calculated according to

$$\theta_x = sin^{-1}(\frac{-r'_{32}}{cos(\theta_y)}) \quad (6.12)$$

$$\theta_y = sin^{-1}(r'_{31}) \quad (6.13)$$

$$\theta_z = sin^{-1}(\frac{-r'_{11}}{cos(\theta_y)}) \quad (6.14)$$

Furthermore, in order to obtain the position relating to the origin position of the color camera, the calibrated coordinate origin must be subtracted from the equation, i.e.,

$$t_{x_0} = t_x - m_{14}, \quad (6.15)$$

$$t_{y_0} = t_y - m_{24}, \quad (6.16)$$

$$t_{z_0} = t_z - m_{34}, \quad (6.17)$$

The rotation angles for the matrix *M* are calculated according to

$$\theta_y^M = sin^{-1}(m_{31}), \quad (6.18)$$

$$\theta_x^M = sin^{-1}(\frac{-m_{32}}{cos(\theta_y^M)}), \quad (6.19)$$

$$\theta_z^M = sin^{-1}(\frac{-m_{21}}{cos(\theta_y^M)}), \quad (6.20)$$

and $\theta_{x_0} = \theta_x - \theta_x^M,\ \theta_{y_0} = \theta_y - \theta_y^M,\ \theta_{z_0} = \theta_z - \theta_z^M$ (6.21)

The moving parameters of the color camera after calibration are:

$(\theta_{x_0}, \theta_{y_0}, \theta_{z_0})and(t_{x_0}, t_{y_0}, t_{z_0})$.

6.4 Preview Synthesis

In order to achieve real time compositions of real and virtual images during a multimedia production process or a movie production process, an image composition interface is adopted in the preview system. In this study, Maya developed by Autodesk Ltd is employed as the interface. Explicitly, all the images and information of the color camera are fed to Maya through its internal memory, and the coordinate system of the virtual camera is positioned matching to the coordinate system of the color camera for facilitating the tracking process of the moving trajectory. After the corresponding relationship between the real object image in Maya and the virtual scene image is determined, the virtual camera can be controlled by a user by controlling the input information of the color camera. Therefore, the user is able to see the result in real time through the preview system realized by Maya. Furthermore, if the color camera and the virtual camera are both 3D stereoscopic cameras, a 3D real-time monitoring operation can be achieved by Maya, since the 3D video signals from the color camera and virtual camera as well as the information of the color camera can be inputted to a stereoscopic camera module of Maya in a synchronization manner.

Refer to Figure 6.8, which shows an example of image composition of the preview system. The figure 6.8a shows the image of a real object. Figure 6.8b is a virtual camera (indicated by a small green spot) looking at a virtual object standing in front of the image of the real object. Figure 6.8c is the virtual image generated by the virtual camera. Figure 6.8d shows a tracking result of the preview system when the color camera pans. Figure 6.8e shows a match between the real and the virtual objects, and figure 6.8f

shows another match after camera motion. The above example is produced under the condition in which the color camera and the depth camera are facing the same direction. Figure 6.9 shows the output for the user to check if the virtual camera moves and couples with the real camera when the real one moves. The virtual images are superimposed on the real image at 50% transparency. In figure 6.9, the red cube (virtual object) adhered to the back-white cube (real object) through time.

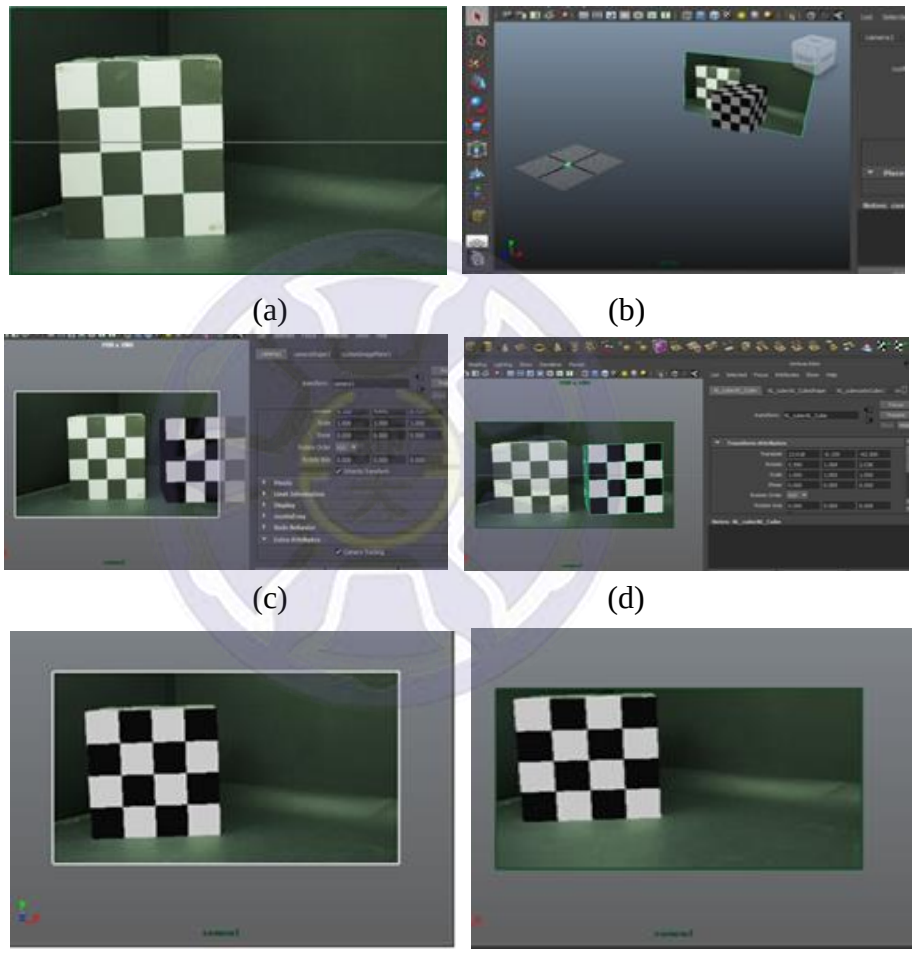


Figure 6.8. Image composite results of a real object and a virtual object (a) the image of a real object (b) a virtual camera (indicated by a small green spot) looking at a virtual object standing in front of the image of the real object (c) the virtual image generated by the virtual camera (d) a tracking result of the preview system when the color camera pans (e) a match between the real and the virtual objects (f) another match after camera motion.

Figure 6.9. The outputs of the preview system.

Chapter 7 Experimental Results

This section describes our preliminary experimental results in various shot conditions. Here we will concentrate on the highlights of three goals. First, we wanted to evaluate the accuracy of VC subsystem. Second, we wanted to compare the overall shot selection quality of our VD system to that of a human director. Moreover, we wanted to show how much MKL improves the shot selection accuracy of VD. Third, we wanted to verify the camera trajectory accuracy of the VRMM system.

7.1 Virtual Cameraman Subsystem

All the host computers used are running 64-bit Windows 7 with a dual-core Intel Core i5 2.5 GHz processor. All VCs was processing in real-time. In our SC, one KINECT and one PTZ camera are used, so the display interface contains three different windows: the bottom-left window is used to show the views from the PTZ camera, and the two top windows are used to show the color views and the range views from KINECT (see Figure 7.1).

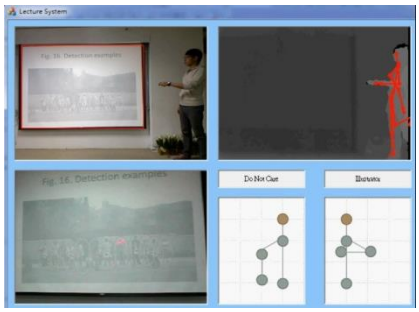


Figure 7.1. User interface of SC system.

Even though several relevant events may occur simultaneously, the PTZ camera is unable to shoot multiple views at once. For example, a baton might be used with a pointing hand, such that two views are eligible for selection: one is the area in which

the baton is waving, and the other is the area in which the speaker's hand is pointing. Suitable definitions of camera actions were required, thus we raised the priorities of laser point detection and baton detection to solve this problem.

Figure 7.2 presents an experimental video recorded by our automated shooting system. In the beginning, the speaker stood beside the screen, so the AdaBoost algorithm was executed to detect the speaker's face, and then the PTZ camera shot the speaker continuously, guided by the results of the mean-shift tracking algorithm (see Figure 7.2).

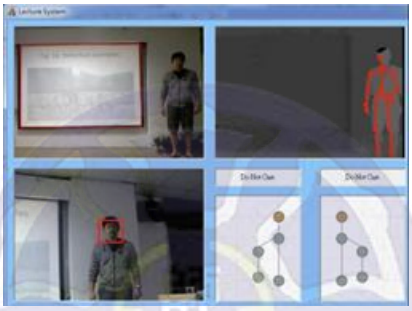


Figure 7.2. PTZ camera shoots the speaker continuously.

While the speaker was using a laser pen, his left hand was illustrating at the same time (Figure 7.3). According to our camera action rules, the PTZ camera did not continue shooting the speaker but instead shot the whole screen area (Figure 7.4). Afterward, the speaker pointed to an area by his right hand (Figure 7.5), and our system recognized the pointing hand, so the PTZ camera shot the area in which the speaker was pointing (Figure 7.6). In the end, the speaker waved a baton around an area of the screen, and his right hand started pointing in the meantime (Figure 7.7). The PTZ camera shot the area in which the baton was being waved, not the area in which his right hand was pointing (Figure 7.8). Figure 7.9 shows an example of AC. The AC controled PTZ camera to take long shot (or wider view) when all audience remained calm. Afterward, someone raised their hand, AC controlled PTZ camera to zoom-in. Table 7.1 shows the overall accuracy results of SC and AC.

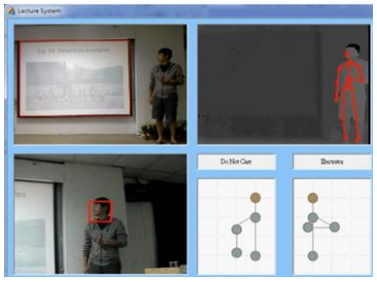


Figure 7.3. Speaker uses a laser pen while his left hand is illustrating.

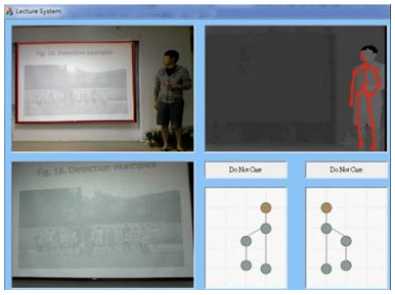


Figure 7.4. PTZ camera shoots the whole screen area.

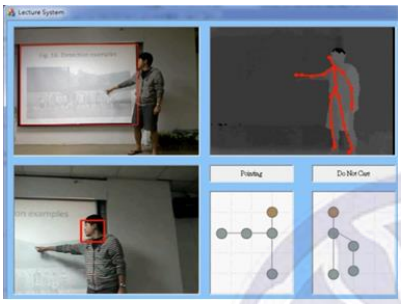


Figure 7.5. Speaker uses his right hand to point to an area.

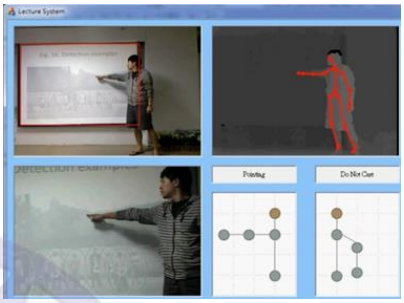


Figure 7.6. PTZ camera shoots the area in which the speaker is pointing.

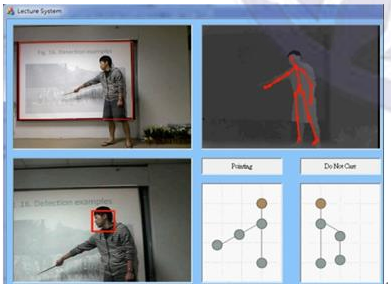


Figure 7.7. Speaker waves a baton to indicate an area and his right hand starts pointing.

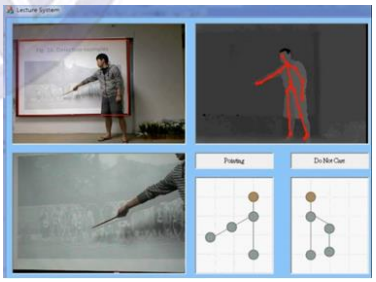


Figure 7.8. PTZ camera shoots the area in which the baton is waving.

(a) (b)

Figure 7.9. Example results of AC (a) AC performed long shot when all audience remained calm. (b) AC performed zoom-in to single audiance who was raising hand.

Table 7.1. Results for overall SC/AC accuracy

	Times	SC	AC
Scene1	8hr06min	**90.3%**	80.35%
Scene2	6hr50min	83.7%	78.65%
Scene3	5hr20min	86.1%	**81.52%**
Avg. Accuracy		87.02%	79.8%

Values in bold indicate the best performance

7.2 Virtual Director Subsystem

The experimental result shows that our VD system can be trained to imitate the distinct styles of several different directors and to produce different categories of videos. In addition, our training scheme made our system more reliable than a system trained with the conventional linear combination scheme.

The VD used the CPN as the decision-making model. Because the CPN involves supervised learning, we improved the quality of learning by inviting a student with real directorial experience to label the shot selection data for the experiment. Only one-third portion of the clips were labeled. For example, if the total length of the video was 47 minutes, and only the first 15 minutes were labeled as training data. The training process was a manual shot selection process (see Figure 7.10), which required the

director to watch three videos from the VC simultaneously and choose one of them for each shot selection. The system recorded all the scores and the label of each chosen shot synchronously. The system was able to work off-line to retrain the model after the manual shot selection (see Figure 7.11). To avoid the output screen shot switching too frequently, the automatic shot selection of VD duration was 3 seconds.

Figure 7.10. Photograph of manual shot selection.

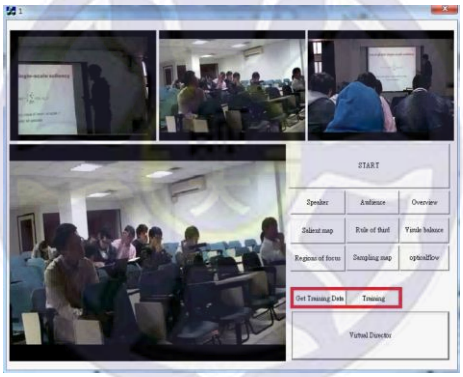


Figure 7.11. Screenshot of manual shot selection.

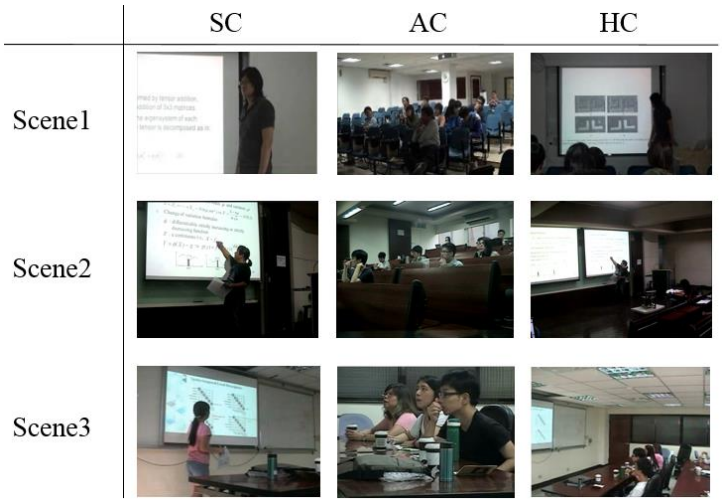


Figure 7.12. The SC, AC, HC shots of the lectures in different types of hall.

Figure 7.12 shows the SC, AC, HC shots of those lectures. Scene1 and Scene2 were both tiered-seating hall, but with different seat deployment. Scene3 was in level hall. Because analyzing shot selection results is usually highly subjective, the results of this experiment were analyzed by a comparison between the proposed virtual system and a conventional decision-making method. In general multiple feature decision-making systems, linear combination (LC) is a common and popular method. As a control group, we used a decision-making model that calculated the LC result from the evaluated scores.

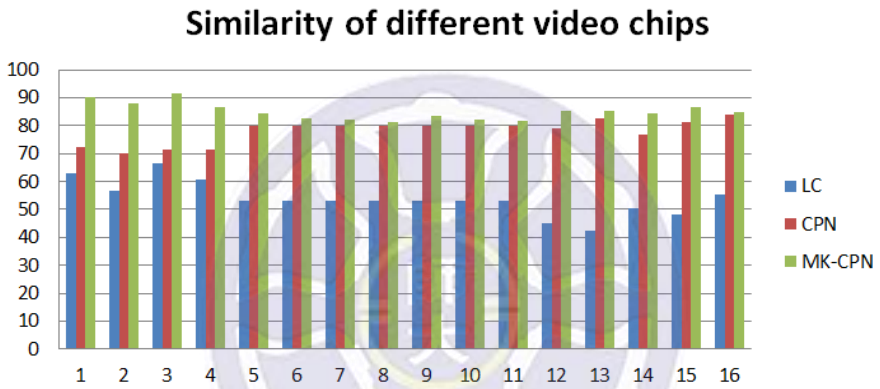


Figure 7.13. Similarity comparison of different video clips.

Figure 7.13 shows the similarity comparison between manual selection (i.e., real director) and selection results by LC, CPN model, and MK-CPN model. Table 7.2 summarizes the average results with different shot selection modules. The results show the selection method of the CPN is closer to the human selection method. The results suggested that MK-CPN module has higher similarity than CPN module (avg. 4%). Figure 7.14 shows the improvement trend while shot decision model changed from LC, CPN to MK-CPN. The results also suggested that the improvement could be applied to different scenes. In other words, the improvement of MK-CPN is scene independent.

Table 7.3 summarizes the total shot change counts of human, random, VD using MK-CPN model. The MK-CPN model shows that it has closer shot change counts than

random ones. It is worth noting that audience shot is not easy to be chosen by VD in scene1 and scene3. Perhaps this is because the speaker continues to act, so the priority of SC has been raised. The average shot change duration time of VD is also very close to human. However, the minimum shot change duration of human is faster than random and VD. The minimum was occurred when an audience asked questions, and human director is able to response in the first time.

Table 7.2. Similarities between manual selection and VD with different shot selection modules.

	Times	LC	CPN	MK-CPN
Scene1	8hr06min	61.7%	73.3%	**89.10%**
Scene2	6hr50min	53.3%	79.9%	**82.52%**
Scene3	5hr20min	48.3%	80.8%	**85.33%**
Avg. Accuracy		55.34%	77.36%	**86.03%**

Values in bold indicate the best performance

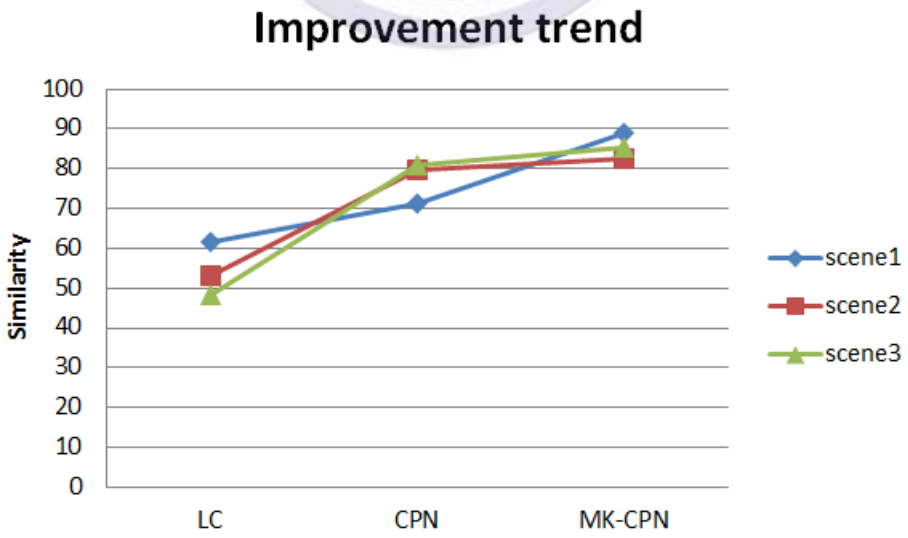


Figure 7.14. Improvement trend chart.

Table 7.3. Shot decision count analysis.

	Times		Shot change counts		
			Human (Label)	Random	VD (MK-CPN)
Scene1	8hr06min	S:	1862	2054	**2102**
		H:	694	2980	**887**
		A:	228	1724	**126**
		Total:	2784	6758	**3115**
Scene2	6hr50min	S:	848	2259	**996**
		H:	691	2106	**684**
		A:	125	2060	**181**
		Total:	1664	6425	**1861**
Scene3	5hr20min	S:	513	1061	**696**
		H:	430	1195	**601**
		A:	310	977	**199**
		Total:	1253	3233	**1496**
Total counts		S:	3223	5374	**3794**
		H:	1815	6281	**2172**
		A:	663	4761	**506**
		Total:	5701	16416	**6472**
Avg. shot change duration			12.79sec	4.44sec	**11.27sec**
Max/min shot change duration			22sec/2sec	12sec/3sec	30sec/3sec

Values in bold indicate the best performance. S, H, and A are representation of speaker shot, hall shot, and audience shot.

Three scenarios were compared for this content: (1) no special event during the speech, (2) audience member asking question, and (3) interaction between the speaker and the audience. Some experimental results are shown in Figure 7.15 and Figure 7.16. In the first case, with no special event, our selection method produced results superior

to those of the LC method. In the second case, when an audience member was asking questions and was caught by the audience camera, our proposed method with the CPN switched the output channel to the audience view. The LC method continued to hold the speaker view without changing shots.

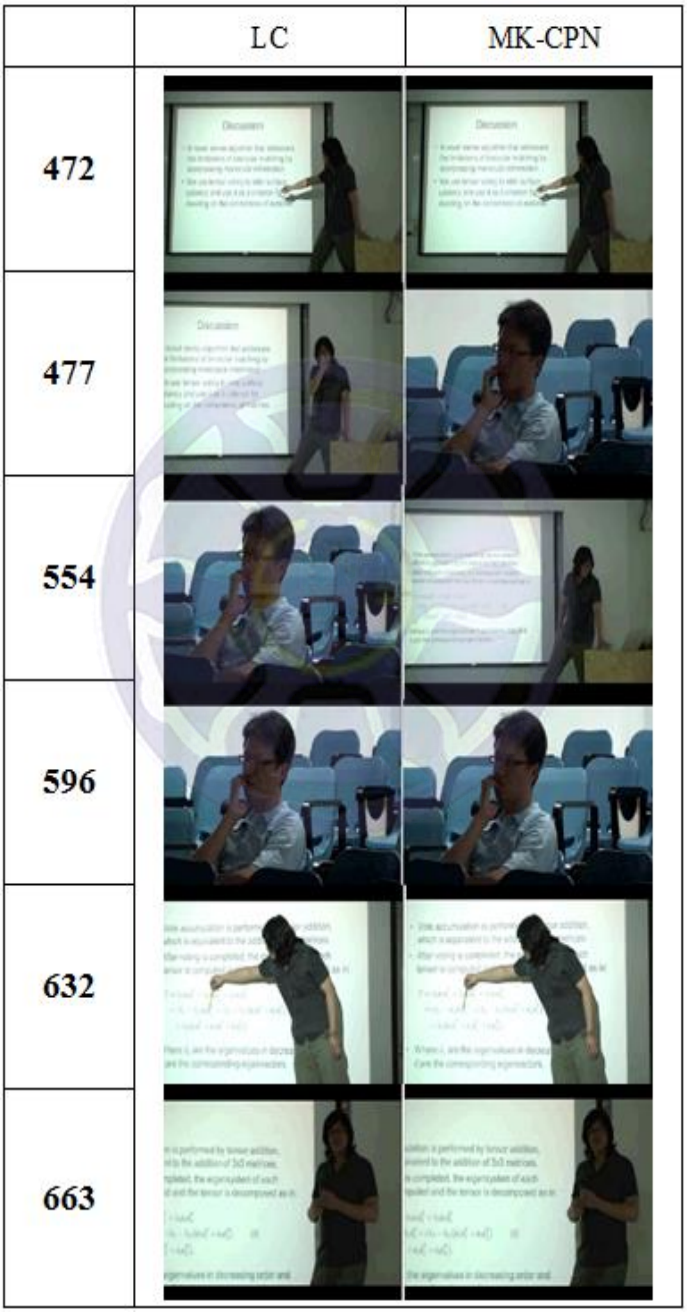


Figure 7.15. Comparison of videos with no special event.

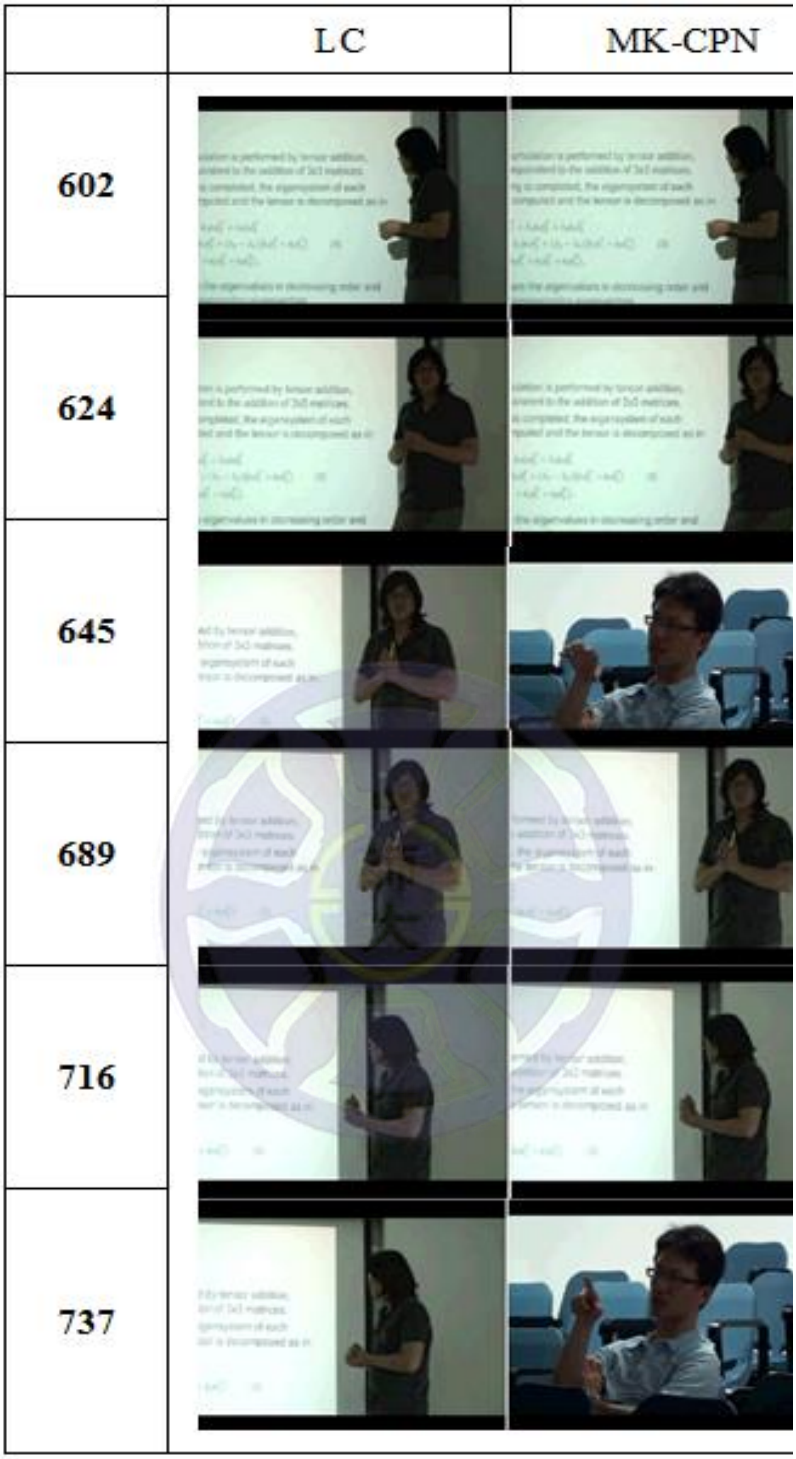


Figure 7.16. Comparison of videos of an audience member asking questions.

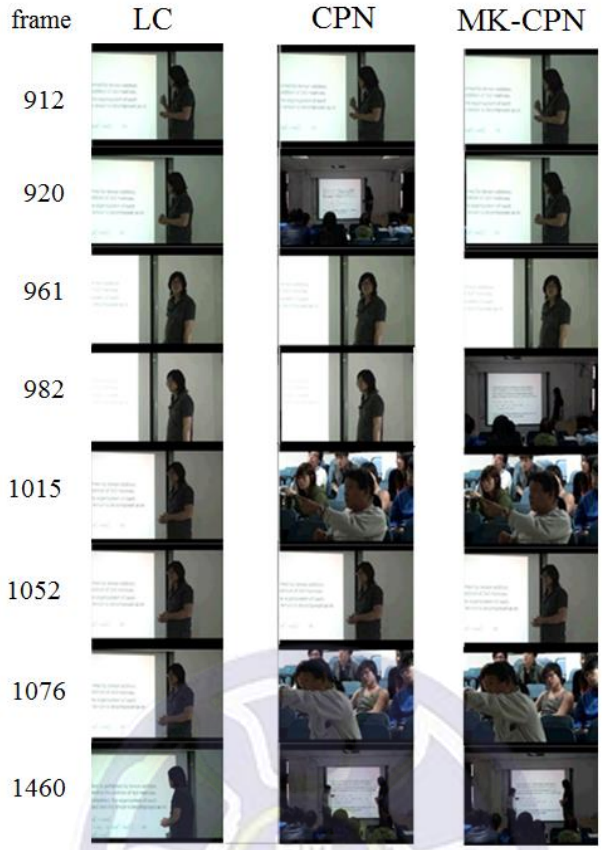


Figure 7.17. Comparison between LC, CPN and MK-CPN for an interaction between speaker and audience.

Figure 7.17 shows successive frame comparison between LC, CPN and MK-CPN. In the third case, interaction occurred between the speaker and the audience. In this case, the speaker view was chosen by the LC method just as in the second case, but the VD based on the CPN produced a variety of shot selections, which were more effective. At frame 920, the speaker was pointing the screen, the shots selected by LC and MK-CPN modules were suitable where CPN module chose HC shot was also acceptable. However, at frame 1015 and 1076, audience was asking question, the LC module still chose SC shot. In particular, at frame 1460, an audience member went to the stage and interacted with the speaker, and the VD selected the hall shot; common sense suggests that the hall shot was the best choice.

To take user experience into count, we measured the SLR's performance by using a few overall quality questions. Questionnaire subjects were 30 people, 21 males and 9 females. The experimental control group was taken by randomly shot selection that alternated every 3 seconds. Each subject watched the 5-minute video and answered the questionnaire. The individual questions we asked and the survey results are summarized in Table 7.4.

Table 7.4. Survey results for individual questions.

(1 = strongly disagree, 5 = strongly agree)	Random		SLR(MK-CPN)	
	Mean	St.dv.	Mean	St.dv
1. The lecture was faithfully presented	3.733	0.853	**3.766**	0.667
2. The content was attractive	2.766	0.725	**3.633**	0.835
3.The system did a good job of shot composition	2.866	0.884	**4.066**	0.573
4.I liked the frequency of shot change	2.2	1.222	**4.1**	0.869
5. Overall, I felt comfortable viewing this video	2.8	1.137	**4.2**	0.832

Values in bold indicate the best performance

Figure 7.18 also shows the histogram of the user survey. In the five questions, the SLR system has a relatively high average performance than random shot selection one. However, the difference in the first question is very small. For further discussion, we divided the user data into two parts: male and female. The histogram of female and male is shown in Figure 7.19 and Figure 7.20, respectively. Compare Figure 7.19 and Figure 7.20, we can find that female users give higher scores in first question than male users. According to the response of male users, they generally believe that there are more audience view can help understand the lecture scene situation. However, some female users think that the audience shot is not appropriate when the speaker speaks.

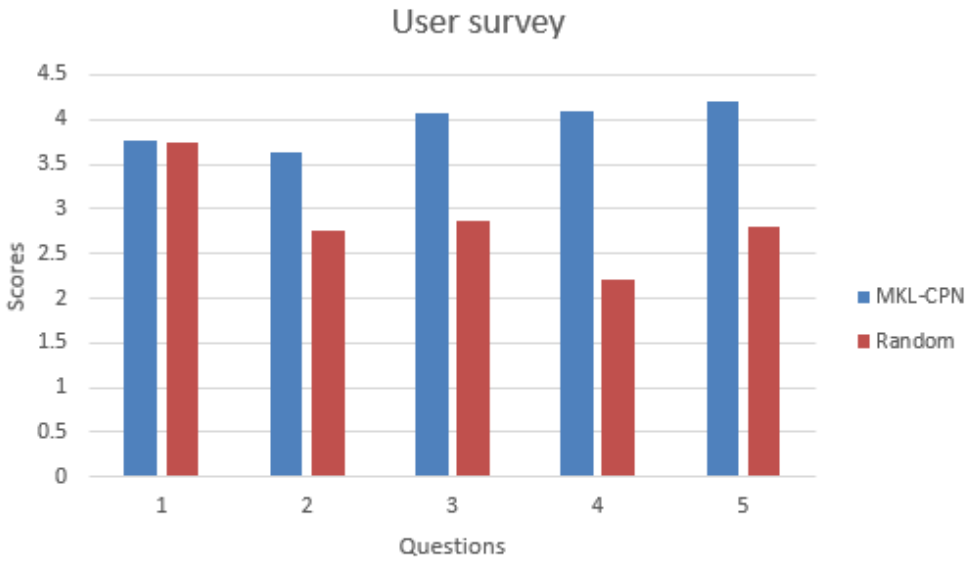


Figure 7.18. User servey histogram between random and MK-CPN methods.

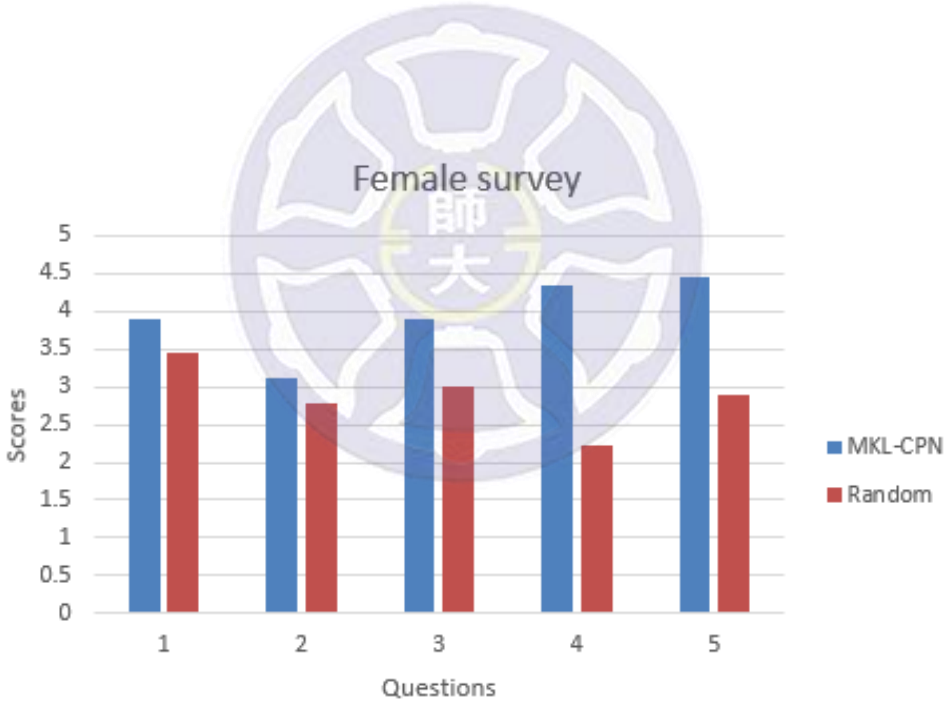


Figure 7.19. Female part of the user servey histogram between random and MK-CPN methods.

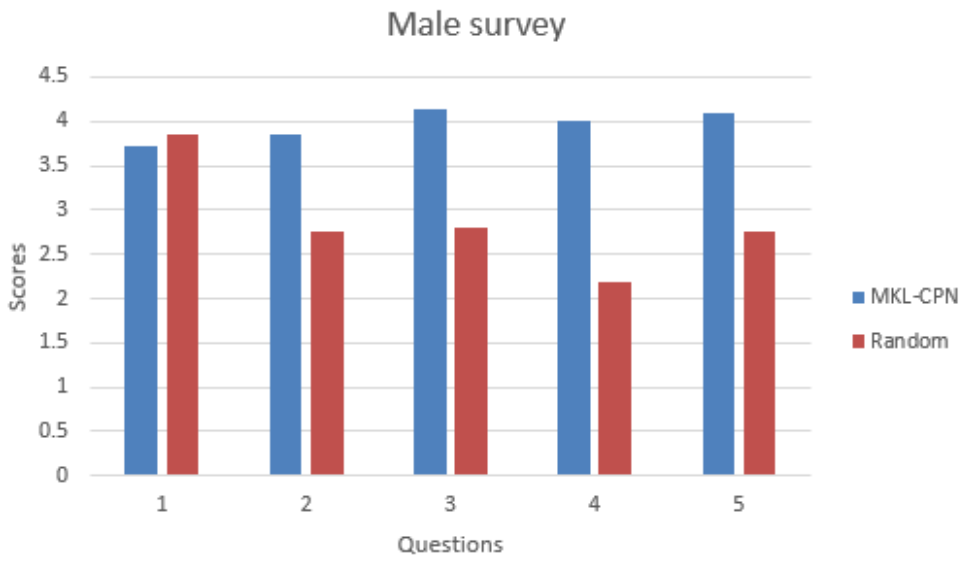


Figure 7.20. Male part of the user servey histogram between random and MK-CPN methods.

7.3 Real-time camera match-moving method

The following experiments were performed to assess the precision of the proposed system and to compare the estimated camera positioning with its actual motion. The experiments comprised an angular measurement and a displacement measurement, whereas each measurement comprised three submeasurements in three different axial directions, namely, X-axis, Y-axis, and Z-axis.

For the angular measurement, an actual camera was mounted on an electric rotary plate (see Figure 7.21.a) for simulating the rotation of the camera about its Y-axis, as would be enabled by a cameraman. Simultaneously, the precise angular variation of the rotation was measured with an electric angle meter (see Figure 7.21.b). At the same time, the movements in the X-axis direction and Z-axis direction of the actual camera were detected and recorded by an electronic level placed above the actual camera.

(a) (b)

Figure 7.21. Angular measurement tools (a) electric rotary plate (b) electric angle meter.

The displacement measurement was performed by mounting an actual camera on a rail for simulating the "dolly" or displacement of a typical camera enabled by a cameraman, that is, the linear displacement of the camera in the X-axis or Z-axis directions. In addition, the displacement was measured with a laser range finder (see Figure 7.22).

(a) (b)

Figure 7.22. Displacement measurement tools (a) dolly rail (b) laser range finder.

Table 7.5. Angular and Linear Displacement Error Measurement

	Angular	**Displacement**
	Average error per degree	**Average error per centimeter**
X-axis	±0.03	±0.04
Y-axis	±0.01	±0.04
Z-axis	±0.07	±0.04
Avg. Accuracy (%)	96%	96%

Both the angular and displacement experimental results are provided in Table 7.5. These results showed that the proposed system can provide satisfactory precision in X-axis and Y-axis angular measurements but limited precision in Z-axis rotation. However, practical video recordings involve limited amounts of Z-axis rotation. The displacement measurements also showed the accuracy of the system for each 1-cm displacement, and the error was less than 0.04 cm.

Chapter 8
Conclusions and Future Work

Our automatic lecture recording system could be applied to a wide range of presentations, ranging from large academic or business seminars, speeches, and presentations, to small general lectures and classroom teaching. In addition to recording the valuable information in a speech, our system could cut personnel costs required for shooting presentations. Whereas a single-camera system might produce a monotonous video, our system uses a variety of shooting rules to produce diverse videos, which audiences experience as unusually immersive and engaging.

We proposed an automatic real-time lecture recording system, the smart lecture recording (SLR) system. The proposed system combines three virtual cameramen (VCs) and a virtual director (VD).

There are three VCs in the proposed system: speaker cameraman (SC), audience cameraman (AC), and hall cameraman (HC); all three use camera subsystems that include pan-tilt-zoom (PTZ) cameras. The SC is composed of a PTZ camera and a depth camera. The SC can track the speaker and perform pose recognition in real time. Three postures are predefined for lecture scenarios, namely pointing, illustrating, and relaxing.

The VD system has two main jobs: shot selection and visual instruction. This VD system gets multiple views from VC and then considers four types of content analysis criteria to evaluate the quality of each view. We also implemented a learning mechanism for choosing the most suitable shot of the input shots. Our learning mechanism is more flexible than a traditional decision-making model. The experimental result demonstrated that our system can actually be applied in the real world. This research mainly focused on three topics: image feature extraction, assessment of image quality

by content analysis, and multiple decision-making for shot selection.

Previous systems have commonly applied optical analysis and aesthetic analysis to estimate the quality levels of shots. Our VD applies common optical analysis and aesthetic analysis, but it also applies action analysis to consider the factors that may interest viewers, and continuity analysis to consider the fluency of a shot when switching.

Real directors do not follow fixed, standardized rules for shot selection. In our design of a multiple decision-making system for shot selection, we proposed a learning mechanism based on our counter propagation network (CPN). Our system can learn skills from real directors, and experiments have proven that our system works like a professional director in the real world. Because our system's learning patterns involve supervised learning, it must be provided with training materials that contain labeled expected output. In the present work, the expected output was provided by a student with directorial experience. If possible, professional directors could provide more extensive expected output to later versions of this system, which would bring the experimental results of future versions closer to professional broadcasting standards.

To facilitate the experimental recordings, the current VD system was implemented on a laptop. The system simultaneously operated on signals received from three cameras, and it was required to calculate vast quantities of information at any given moment. To prevent latency and jitter on the output screen, salient object detection was implemented on cameras. The VC calculated salient information and passed it to the VD. Thus, the VD host efficiently allocated its computing power to producing an extremely smooth broadcast.

For video editing and postproduction, we present a system that provides a real-time automatic camera match-moving method for virtual–real synthesis before film postproduction. The proposed system consists of two subsystems. One is the high-

accuracy camera-tracking system, which can reconstruct the camera angles of live-action footage in real time. The other subsystem is the real-time virtual–real preview system, which controls virtual cameras and optimizes the stereo parameter settings. The preview system is implemented as a computer graphics software plug-in that inputs the virtual camera key frames photographed from specific camera angles and outputs the rendered footage. The goal of the proposed method is to build a quick and robust match-moving method for VFX synthesis before postproduction.

Because the field of view of each depth camera is limited to approximately 57 degrees and the real object is generally illuminated by high-power lighting during the shooting, the depth information acquired from the depth camera may not be perfect, and thus, the precision of the measured depth map can be adversely affected. Therefore, to enlarge the depth measurement range of the depth camera, a future depth camera unit could combine several depth cameras facing in different directions to generate different depth frames; the depth frames of those depth cameras could be composed into one depth map.

In the future, the VRMM component and speech transcription are considerate to integrate into online SLR system. To combine the VRMM component into the SLR system, the temporal depth fusion need to be modified to PTZ camera version, because PTZ camera only have pan, tilt, and zoom operations but without sift operation. Nowadays, thanks to the development of deep learning technology, the accuracy of speech transcription has been significantly improved. The speech transcription allows the SLR system to instantly synthetic subtitles to recording video that make the SLR system more practical. Furthermore, we intend to upgrade the PTZ cameras to 1080p (Full HD) high-resolution cameras to replace the existing 480p PTZ cameras. High-resolution camera is now the mainstream equipment and allows users to have a better experience. However, the use of high-resolution camera will also bring additional load

to the system. For example, high-resolution images can cause memory usage to rise to more than six times. In addition, the content analysis of the rating algorithm must also take into account the operational efficiency of the problem.

In addition, we want to integrate deep learning and active learning into our system. As we discussed in Chapter 1, the limitation of CNN is its local feature learning property where content analysis usually require global and temporal information. However, CNN is very suitable for object recognition or detection. For example, it is possible to use CNN to detect humans and tools in the shot. Therefore, we could use the information to increase event detection accuracy. In some situations, unlabeled data is abundant but manual labeling is exorbitantly expensive. In this scenario, learning algorithms can actively query for labels. The labeling task would be done automatically by a decision system or manually by a professional user. This type of iterative supervised learning is called active learning. Because the learner chooses the examples, the number of examples to learn a concept can often be much lower than the number required in normal supervised learning. The concept diagram is shown in Figure 8.1. The benefits of active learning are as follows: it enables the achievement of online learning, it converges rapidly, and it demonstrates a high recognition rate.

The proposed SLR system with the deep learning and active learning concepts will not be limited to shooting live speeches; it will also be practicable for recording live performances, video games, concerts, product launches, and numerous other events. Compared to the common criteria system, which is more dependent on the shooting situation, our system could utilize diverse methods to make the application system more flexible by offering various types of training materials.

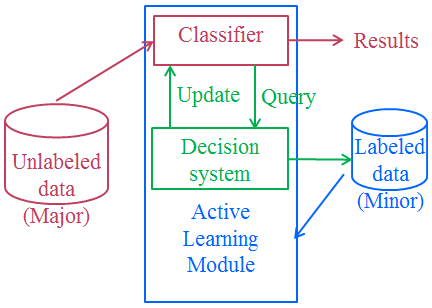


Figure 8.1. Concept of active learning.

Appendix

1. Projection Screen Location

In general, the projection screen area is much brighter than the other areas, so this characteristic can be used to detect the screen area. First, convert the color image from the RGB color space to the HSV color space, and then connect the pixels with high values (V) into areas (Figure A.1). Unsuitable areas such as lights and windows must be filtered.

Figure A.1. Result of screen detection (red/green rectangles).

Before we detect the screen area, a color transform is performed. The color space proposed here is the HSV color space (Figure A.2). The three channels of HSV are Hue, Saturation, and Value (intensity).

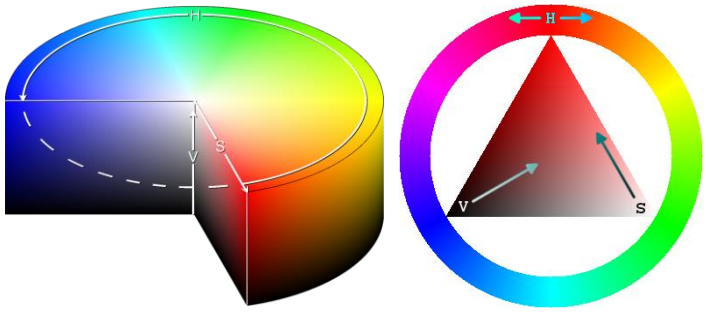


Figure A.2. HSV color space (from Wikipedia).

To find the screen, we consider the value channel. By Otsu's method, which automatically performs clustering-based image thresholding and the connected component technique, we can obtain several candidate regions (see Figure A.3). Then, we use the size filter to find the most appropriate region and locate the area (as in

Figure A.4).

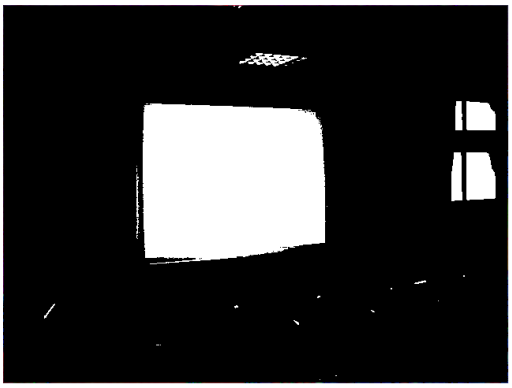


Figure A.3. Candidates of projection screen (including light and windows).

Figure A.4. Detected projection screen (green rectangle).

2. Laser Point Detection

When a speaker shines a laser spot on the screen, it encourages the audience to focus on the highlighted text and pictures. The laser points irradiated by the laser pen have a higher intensity in the screen area. The detection of any laser spot depends on whether that laser spot is located on the screen area or not; the spot will have the highest intensity in the screen area. By using those features, we could design an algorithm to detect whether the speaker was using a laser pointer.

The detection procedure is performed between the frames of each successive frame pair within the projection screen area. First, try to find any pixels that differ from those of the previous frame and check whether the difference is higher than the constant

threshold. Next, check whether the intensity of the pixels is higher than the threshold. If both answers are “yes,” then perform the connected component technique to group the pixels together and filter unsuitable areas by size; the laser light point can be located within a relevant region.

After finding the laser point in the projection screen area, our system orders the PTZ camera to fill the whole area of the shot with the projection screen (see Figure A.5) and simultaneously to send a message notifying the VD that the speaker is using a laser pointer now.

Figure A.5. Laser spot detection. The PTZ camera moves and shoots the whole screen area.

3.Baton Detection

When a speaker uses a baton, that speaker typically waves that baton to emphasize the content on the projection screen and to instruct the audience. Suppose the speaker waves the baton continuously in the screen area when the baton is used to assist his or her speech; then our system can apply motion detection to find the area in which the baton moves. Just like laser spot detection, baton detection is limited to the projection screen. The baton is thinner and more elongated than the speaker (see Figure A.6). By using those features, our algorithm can detect whether the speaker is using a baton. The detection procedure is performed between the frames of each successive frame pair; only the pixels within the projection screen area are compared. First, we calculate the image difference and compare them against a predefined threshold to extract the pixels that suggest a large motion. Then, we perform the connected component technique to

find the rectangular bounding box and we filter unsuitable areas by size; the baton can be located within the region returned by this method.

In addition, when a speaker uses a baton, the focus of the picture is also in the baton's region. After finding the area where the baton is waving in the screen area, our system controls the PTZ camera to zoom in, to shoot the baton's area, and simultaneously to send a message to notify the VD that the speaker is using a baton. If the system has been tracking the baton, but at some point, the baton is no longer detected, then the VC must zoom out and go back to tracking the speaker.

Figure A.6. Baton detection. The PTZ camera moves and shoots the area in which the baton is waving.

4. Camera Control Table

Table A.1: Event and camera-control table.

Location	Laser pen	Baton	Pointing	Illustrating	Speaker Direction	PTZ control
disappear						Pan slowly
In projection screen					Left	Pan, tilt, zoom out. The speaker's region covers 1/3 of the area of the screen; the size of a face is 1/48 of image width.
In projection screen		●			Left	Pan, tilt, zoom in. The size of a baton is 1/4 of the size of the projection screen.
In projection screen		●	●		Left	Pan, tilt, zoom in. The size of a baton is 1/4 of the size of the projection screen.
In projection screen			●		Left	Pan, tilt, zoom-in. The size of hand is 1/4 of the size of the projection screen.
In projection screen				●	Left	Pan, tilt. Track speaker.
In projection screen					Front	Pan, tilt, zoom out. The speaker's region is located at the middle of the screen; the size of a face is 1/48 of image width.
In		●			Front	Pan, tilt, zoom-in.

projection screen						The size of a baton is 1/4 of the size of the projection screen.
In projection screen		●	●		Front	Pan, tilt, zoom-in. The size of a baton is 1/4 projection screen.
In projection screen			●		Front	Pan, tilt, zoom-in. The size of a hand is 1/4 of the size of the projection screen.
In projection screen				●	Front	Pan, tilt. Track the speaker.
In projection screen					Right	Pan, tilt, zoom out. The speaker's region covers 2/3 of the area of the screen; the size of a face is 1/48 of the image width.
In projection screen		●			Right	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
In projection screen		●	●		Right	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
In projection screen			●		Right	Pan, tilt, zoom-in. The size of a hand is 1/4 of the size of the projection screen.

In projection screen				●	Right	Pan, tilt. Track speaker.
Outside projection screen					Left	Pan, tilt, zoom out. The speaker's region covers 1/3 of the area of the screen; the size of a face is 1/24 of the size of the image width.
Outside projection screen	●				Left	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●		●		Left	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●			●	Left	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen		●			Left	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
Outside projection screen		●	●		Left	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
Outside projection screen			●		Left	Pan, tilt, zoom-in. The size of a hand is 1/4 of the size of the projection screen.
Outside projection screen				●	Left	Pan, tilt. Track speaker.

Outside projection screen					Front	Pan, tilt, zoom out. The speaker's region is located at middle of the screen; the size of a face is 1/24 of the image width.
Outside projection screen	●				Front	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●		●		Front	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●			●	Front	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen		●			Front	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
Outside projection screen		●	●		Front	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
Outside projection screen			●		Front	Pan, tilt, zoom-in. The size of a hand is 1/4 of the size of the projection Screen.
Outside projection screen				●	Front	Pan, tilt. Tracking speaker.
Outside projection screen					Right	Pan, tilt, zoom out. The speaker's region is covers 2/3 of the area of the screen, the

						size of a face is 1/24 of the image width.
Outside projection screen	●				Right	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●		●		Right	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen	●			●	Right	Pan, tilt, zoom-in. Fit image to projection screen.
Outside projection screen		●			Right	Pan, tilt, zoom-in. The size of a baton is 1/4 of the size of the projection screen.
Outside projection screen		●	●		Right	Pan, tilt, zoom-in. The size of a baton is 1/4 projection screen.
Outside projection screen			●		Right	Pan, tilt, zoom-in. The size of a hand is 1/4 of the size of the projection screen.
Outside projection screen				●	Right	Pan, tilt. Track speaker.

5. Visual Instruction List Table

Table A.2. Visual instruction list (speaker posture: pointing)

Event	Speaker video (posture)	Audience video (movement)	Hall video (movement)	Director instruction
No. 1	Pointing	(Big motion, Big range)	(Big motion, Big range)	AC: zoom out, HC: pan/ tilt
No. 2	Pointing	(Big motion, Big range)	(Big motion, Small range)	AC: zoom in
No. 3	Pointing	(Big motion, Big range)	(Small motion, Big range)	SC: zoom in, HC: pan/ tilt
No. 4	Pointing	(Big motion, Big range)	(Small motion, Small range)	HC: pan/ tilt
No. 5	Pointing	(Big motion, Small range)	(Big motion, Big range)	AC: zoom in
No. 6	Pointing	(Big motion, Small range)	(Big motion, Small range)	SC: zoom in, AC: zoom in
No. 7	Pointing	(Big motion, Small range)	(Small motion, Big range)	AC: zoom in
No. 8	Pointing	(Big motion, Small range)	(Small motion, Small range)	AC: zoom in
No. 9	Pointing	(Small motion, Big range)	(Big motion, Big range)	AC: zoom out, HC: pan/ tilt
No. 10	Pointing	(Small motion, Big range)	(Big motion, Small range)	AZ: zoom in
No. 11	Pointing	(Small motion,	(Small motion,	SC, AC: zoom in

		Big range)	Big range)	
No. 12	Pointing	(Small motion, Big range)	(Small motion, Small range)	AC: zoom out, HC: pan/ tilt
No. 13	Pointing	(Small motion, Small range)	(Big motion, Big range)	SC: zoom in, AC: zoom out
No. 14	Pointing	(Small motion, Small range)	(Big motion, Small range)	HC: pan/ tilt
No. 15	Pointing	(Small motion, Small range)	(Small motion, Big range)	SC: zoom in
No. 16	Pointing	(Small motion, Small range)	(Small motion, Small range)	SC: zoom in, HC: pan/ tilt

Table A.3. Visual instruction list (speaker posture: illustrating)

Event	Speaker video (posture)	Audience video (movement)	Hall video (movement)	Director instruction
No. 17	Illustrating	(Big motion, Big range)	(Big motion, Big range)	AC: zoom out
No. 18	Illustrating	(Big motion, Big range)	(Big motion, Small range)	AC: zoom out
No. 19	Illustrating	(Big motion, Big range)	(Small motion, Big range)	HC: pan/ tilt
No. 20	Illustrating	(Big motion, Big range)	(Small motion, Small range)	SC: zoom in, AC: zoom out
No. 21	Illustrating	(Big motion, Small range)	(Big motion, Big range)	SC, AC: zoom in
No. 22	Illustrating	(Big motion, Small range)	(Big motion, Small range)	AC: zoom in, HC: pan/ tilt
No. 23	Illustrating	(Big motion, Small range)	(Small motion, Big range)	AC: zoom in, HC: pan/ tilt
No. 24	Illustrating	(Big motion, Small range)	(Small motion, Small range)	AC: zoom in
No. 25	Illustrating	(Small motion, Big range)	(Big motion, Big range)	AC: zoom out, HC: pan/ tilt
No. 26	Illustrating	(Small motion, Big range)	(Big motion, Small range)	AZ: zoom in
No. 27	Illustrating	(Small motion, Big range)	(Small motion, Big range)	SC, AC: zoom in

No. 28	Illustrating	(Small motion, Big range)	(Small motion, Small range)	AC: zoom out, HC: pan/ tilt
No. 29	Illustrating	(Small motion, Small range)	(Big motion, Big range)	HC: pan/ tilt
No. 30	Illustrating	(Small motion, Small range)	(Big motion, Small range)	SC: zoom in
No. 31	Illustrating	(Small motion, Small range)	(Small motion, Big range)	SC: zoom in, HC: pan/ tilt
No. 32	Illustrating	(Small motion, Small range)	(Small motion, Small range)	SC: zoom in, AC: zoom in

Table A.4. Visual instruction list (speaker posture: relaxing)

Event	Speaker video (posture)	Audience video (movement)	Hall video (movement)	Director instruction
No. 33	Relaxing	(Big motion, Big range)	(Big motion, Big range)	AC: zoom in
No. 34	Relaxing	(Big motion, Big range)	(Big motion, Small range)	HC: pan/ tilt
No. 35	Relaxing	(Big motion, Big range)	(Small motion, Big range)	HC: pan/ tilt
No. 36	Relaxing	(Big motion, Big range)	(Small motion, Small range)	AC: zoom out
No. 37	Relaxing	(Big motion, Small range)	(Big motion, Big range)	AC, SC: zoom in
No. 38	Relaxing	(Big motion, Small range)	(Big motion, Small range)	HC: pan/ tilt
No. 39	Relaxing	(Big motion, Small range)	(Small motion, Big range)	AC: zoom in
No. 40	Relaxing	(Big motion, Small range)	(Small motion, Small range)	AC: zoom out
No. 41	Relaxing	(Small motion, Big range)	(Big motion, Big range)	AZ: zoom in
No. 42	Relaxing	(Small motion, Big range)	(Big motion, Small range)	SC, AC: zoom in
No. 43	Relaxing	(Small motion, Big range)	(Small motion, Big range)	AC: zoom out, HC: pan/ tilt

No. 44	Relaxing	(Small motion, Big range)	(Small motion, Small range)	SC: zoom in, AC: zoom out
No. 45	Relaxing	(Small motion, Small range)	(Big motion, Big range)	HC: pan/ tilt
No. 46	Relaxing	(Small motion, Small range)	(Big motion, Small range)	SC: zoom in
No. 47	Relaxing	(Small motion, Small range)	(Small motion, Big range)	SC: zoom in, HC: pan/ tilt
No. 48	Relaxing	(Small motion, Small range)	(Small motion, Small range)	SC, AC: zoom in

References

[1] L. A. Rowe, D. Harley, P. Pletcher, and S. Lawrence, “BIBS: A Lecture Webcasting System,” Berkeley Multimedia Research Center, 2001.

[2] Y. Rui, L. He, A. Gupta, and Q. Liu, “Building an Intelligent Camera Management System,” Proceedings of the ACM International Conference on Multimedia, vol. 9, pp. 2-11, 2001.

[3] M. Bianchi, “AutoAuditorium: A Fully Automatic, Multi-Camera System to Televise Auditorium Presentations,” Proceedings of the Joint DARPA/NIST Smart Spaces Technology Workshop, 1998.

[4] M. Bianchi, “Automatic Video Production of Lectures Using an Intelligent and Aware Environment,” Proceedings of the International Conference on Mobile and Ubiquitous Multimedia, pp. 117-123, 2004.

[5] G. D. Abowd, “Classroom 2000: An Experiment with the Instrumentation of a Living Educational Environment,” IBM Systems Journal, vol. 38, no. 4, pp. 508-530, 1999.

[6] G. Cruz and R. Hill, “Capturing and Playing Multimedia Events with STREAMS,” Proc. ACM Int’l Conf. on Multimedia, pp. 193-200, 1994.

[7] C. Zhang, Y. Rui, J. Crawford, and L.W. He, “An Automated End-to-end Lecture Capture and Broadcasting System,” Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP), vol. 4, no. 1, pp. 2-11, 2008.

[8] R. Yong, G. Anoop, G. Jonathan, and L.W. He, “Automating Lecture Capture and Broadcast: Technology and Videography,” Multimedia Systems, vol. 10, no. 1, pp. 3-15, 2004.

[9] R. Baecker, “A Principled Design for Scalable Internet Visual Communications with Rich Media, Interactivity, and Structured Archives,” Proceedings of the Centre for

Advanced Studies on Collaborative research, pp. 16-29, 2003.

[10] M. Onishi and K. Fukunaga, “Shooting the Lecture Scene Using Computer-Controlled Cameras based on Situation Understanding and Evaluation of Video Images,” Proceedings of the International Conference on Mobile and Ubiquitous Multimedia, pp. 781-784, 2004.

[11] C. F. Juang and C. M. Chang, “Human Body Posture Classification by a Neural Fuzzy Network and Home Care System Application,” Transactions on Systems, Man, and Cybernetics - Part A: Systems and Humans, vol. 37, no. 6, pp. 984-994, 2007.

[12] M. Ozeki, Y. Nakamura, and Y. Ohta, “Human Behavior Recognition for an Intelligent Video Production System,” Proceedings of the IEEE Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing, pp. 1153-1160, 2002.

[13] K. H. Cheng, C. H. Hsieh, C. C. Wang, “Human Action Recognition Using 3D Body Joints,” Proceedings of the International Conference on Computer Vision, Graphics, and Image Processing (IPPR), session D2-2, 2011.

[14] S. Y. Lin, Z. H. You, and Y. P. Hung, “A Real-Time Action Recognition Approach with 3D Tracked Body Joints and Its Application,” Proceedings of the International Conference on Computer Vision, Graphics, and Image Processing (IPPR), session B5-2, 2011.

[15] C. M. Huang, Y. R. Chen, and L. C. Fu, “Visual Tracking of Human Head and Arms Using Adaptive Multiple Importance Sampling on a Single Camera in Cluttered Environments,” IEEE Transactions on Sensors, vol. 14, no. 7, pp. 2267-2275, 2014.

[16] C. T. Lu and S.W. Chen, “Automatic Lecture Recording System,” Proceedings of the International Conference on Computer Vision, Graphics, and Image Processing (IPPR), session D1-3, 2011.

[17] C. Zhang, Y. Rui, L. He, and M. Wallick, “Hybrid speaker tracking in an automated lecture room,” Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 81-84, 2005.

[18] T. Yokoi, and H. Fujiyoshi,”Virtual camerawork for generating lecture video from high resolution images,” Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 751-754, 2005.

[19] Q. Huang, Y. T. Cui, and S. Samarasekera, ” Content based active video data acquisition via automated cameramen,” Proceedings of the IEEE International Conference on Image Processing (ICIP), p.p.808-812, 1998.

[20] M. Wallick, Y. Rui, and L. He “A portable solution for automatic lecture room camera management,” Proceedings of the IEEE International Conference on Multimedia and Expo, pp. 987-990, 2004.

[21] T. Y. Li and X. Y. Xiao, “An Interactive Camera Planning System for Automatic Cinematographer,” Proceedings of the IEEE International Conference on Multimedia Modelling, 2005.

[22] F. Lampi, S. Kopf, M.Benz, and W. Effelsberg, “An automatic cameraman in a lecture recording system,” Proceedings of the International Workshop on Educational Multimedia and Multimedia Education, p.p. 11-18, 2007.

[23] Y. Cheng, “Mean Shift, Mode Seeking, and Clustering,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 8, pp. 790-799, 1995.

[24] P. Ekman and W. V. Friesen, “Nonverbal Behavior and Psychopathology,” The psychology of Depression, pp. 203-233, 1969.

[25] M. Gleicher and J. Masanz, “Towards Virtual Videography,” Proceedings of the ACM International Conference on Multimedia, pp. 375-378, 2000.

[26] S. Okuni, S. Tsuruoka, G. P. Rayat, H. Kawanaka, and T. Shinogi, “Video Scene

Segmentation Using the State Recognition of Blackboard for Blended Learning," International Conference on Convergence Information Technology, pp. 2437-2442, 2007.

[27] M. Kumano, Y. Ariki, M. Amano, K. Uehara, "Video Editing Support System Based on Video Grammar and Content Analysis," Proceedings of the International Conference on Pattern Recognition (ICPR), vol. 2, pp. 1031-1036, 2002.

[28] T. Wang, A. Mansfield, R. Hu, and J. Collomosse, "An Evolutionary Approach to Automatic Video Editing," Proceedings of the International Conference on Visual Media Production (CVMP), pp. 127-134, 2009.

[29] E. Machnicki and L. Rowe, "Virtual Director: Automating a Webcast," SPIE Multimedia Computer Network, pp. 208-225, 2002.

[30] S. Goferman, L. Zelnik-Manor, and A. Tal, "Context-Aware Saliency Detection," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 10, pp. 1915-1926, 2012.

[31] C. J. Fang, S. W. Chen, and C. S. Fu, "Automatic Change Detection of Driving Environments in a Vision-Based Driver Assistance System," IEEE Transactions on Neural Networks, vol. 14, no.3, pp.646-657, 2003.

[32] F. Wang, C. W. Ngo, and T. C. Pong, "Synchronization of Lecture Videos and Electronic Slides by Video Text Analysis," Proceedings of the ACM International Conference on Multimedia, pp. 315-318, 2003.

[33] S. Fiori, "A theory for learning based on rigid bodies dynamics," IEEE Transactions on Neural Networks, vol. 13, no. 3, pp. 521-531, 2002.

[34] Y. H. Hsiao and C. C. Chen, "A Sparse Sample Collection and Representation Method Using Re-weighting and Dynamically Updating OMP for Fish Tracking,

“ Proceedings of the IEEE International Conference on Image Processing, pp. 3494-3497, 2016.

[35] B. F. Wu, C. T. Lin, and C. J. Chen, “Real-time Lane and Vehicle Detection Based on a Single Camera Model, ” International Journal of Computers and Applications, vol. 32, no.2, pp. 149-159, 2010.

[36] R. Setiono, W. K. Leow, and J.M. Zurada, “Extraction of rules from artificial neural networks for nonlinear regression,” IEEE Transactions on Neural Networks, vol 13, no.3, pp. 564-577, 2002.

[37] C. C. Balázs, “Approximation with Artificial Neural Networks,” Faculty of Sciences, Eötvös Loránd University , 2001.

[38] Y. Bengio, “Learning Deep Architectures for AI,“ Foundations and Trends in Machine Learning. vol.2, no.1, pp.1-127, 2009.

[39] Y. Bengio, A. Courville, and P. Vincent, ”Representation Learning: A Review and New Perspectives,“ IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 35, no.8, pp. 1798-1828, 2013.

[40] L. Deng, and D. Yu, ”Deep Learning: Methods and Applications, ” Foundations and Trends in Signal Processing. vol.7 no.3-4, pp.1-199, 2014.

[41] J. Schmidhuber, “Deep Learning in Neural Networks: An Overview, ” Neural Networks, vol. 61 pp.85-117, 2015.

[42] Y. Bengio, “Practical recommendations for gradient-based training of deep architectures, ” 2012.

[43] C. Metz, “Facebook's 'Deep Learning' Guru Reveals the Future of AI, ” Wired, 2013.

[44] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, “Backpropagation Applied to Handwritten Zip Code Recognition, ” Neural Computation, vol.1, pp. 541-551, 1989.

[45] Y. LeCun, L. D. Jackel, B. Boser, J. S. Denker, H. P. Graf, I. Guyon, D. Henderson, R. E. Howard, and W. Hubbard, “Handwritten digit recognition: Applications of neural net chips and automatic learning, ” IEEE Transactions on Communication, p.p. 41-46, 1989.

[46] Y. LeCun, L. Bottou, and Y. Bengio, “Reading checks with graph transformer networks, ” International Conference on Acoustics, Speech, and Signal Processing, vol. 1, p.p.151-154, 1997.

[47] D. Scherer, A. C. Müller, and S. Behnke, “Evaluation of Pooling Operations in Convolutional Architectures for Object Recognition, ” International Conference on Artificial Neural Networks (ICANN), pp. 92-101.2010.

[48] S. Lawrence, C. G. Lee, A. C. Tsoi, and A. D. Back, “Face Recognition: A Convolutional Neural Network Approach, ” IEEE Transactions on Neural Networks, vol.8, no.1, p.p. 98-113, 1997.

[49] N. Srivastava, C. G. Hinton, A. Krizhevsky, I. Sutskever; and R. Salakhutdinov, “Dropout: A Simple Way to Prevent Neural Networks from overfitting, ” Journal of Machine Learning Research. vol.15, no.1, p.p. 1929-1958, 2014.

[50] P. LeCallet, V. G. Christian, and B. Dominique, “A Convolutional Neural Network Approach for Objective Video Quality Assessment, ” IEEE Transactions on Neural Networks, vol.17, no.5, p.p. 1316-1327, 2006.

[51] A. Krizhevsky, I. Sutskever,and G. Hinton, “ImageNet Classification with Deep Convolutional Neural Networks, ” International Conference on Neural Information Processing Systems (NIPS), 2012.

[52] D. Ciresan, M. Ueli, M. Jonathan, M.G. Luca, and S. Jurgen, “Flexible, High Performance Convolutional Neural Networks for Image Classification, ” Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence, vol.2, p.p.1237-1242, 2011.

[53] S.W. Ji, W. Xu, M. Yang, and K. Yu, Kai, “3D Convolutional Neural Networks for Human Action Recognition, ” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.35, no.1, p.p. 221-231, 2013.

[54] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning, ” Nature, vol.521 ,no.7553, p.p. 436-444, 2015.

[55] J. Baker, L. Deng, J. Glass, S. Khudanpur, C.H. Lee, N. Morgan, and D. O'Shaughnessy, “Research Developments and Directions in Speech Recognition and Understanding, Part 1, ” IEEE Signal Processing Magazine, vol.26, no.3, p.p.75-80, 2009.

[56] G.Hinton, L. Deng, D. Yu, G. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T.Sainath, and B.Kingsbury, “Deep Neural Networks for Acoustic Modeling in Speech Recognition --- The shared views of four research groups,” IEEE Signal Processing Magazine, vol.29, no.6, p.p. 82-97, 2012.

[57] R Collobert and J. Weston, “A unified architecture for natural language processing: Deep neural networks with multitask learning, ” Proceedings of the ACM International Conference on Machine learning, 2008.

[58] C. Y. Chen, A. Seff, A. Kornhauser, J.X. Xiao, “DeepDriving: Learning Affordance for Direct Perception in Autonomous Driving, ” IEEE International Conference on Computer Vision (ICCV), pp. 2722-2730, 2015.

[59] B. Firner, B. Flepp, K. Zieba, L. Jackel, M. Bojarski, and U. Muller, “End-to-End Deep Learning for Self-Driving Cars, ” Parallel Forall. NVIDIA, 2016.

URL: https://devblogs.nvidia.com/parallelforall/deep-learning-self-driving-cars/

[60] R. Hadsell, A. Erkan, P. Sermanet, M. Scoffier, U. Muller and Y. LeCun, ” Deep belief net learning in a long-range vision system for autonomous off-road driving,” IEEE International Conference on Intelligent Robots and Systems, p.p. 628 – 633, 2008.

[61] B. Kisačanin, “Deep Learning for Autonomous Vehicles,” IEEE International Symposium on Multiple-Valued Logic (ISMVL), p.p. 142-142, 2017.

[62] R. Hecht-Nielsen, “Counterpropagation Networks,” Applied Optics, vol. 26, no. 23, pp. 4979-4983, 1987.

[63] M. G¨onen and E. Alpaydın, “Multiple Kernel Learning Algorithms,” Journal of Machine Learning Research, vol. 12, pp. 2211-2268, 2011.

[64] Y. Y. Lin, T. L. Liu, and C. S. Fuh, “Multiple Kernel Learning for Dimensionality Reduction,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 6, pp. 1147-1160, 2011.

[65] M. Gleicher and J. Masanz, “Towards Virtual Videography,” Proceedings of the ACM International Conference on Multimedia, pp. 375-378, 2000.

[66] Q. Liu, Y. Rui, A. Gupta, and J. J. Cadiz, “Automating Camera Management for Lecture Room Environments,” Proceedings of the SIGCHI International Conference on Human Factors in Computing Systems, Seattle, Washington, USA, pp. 442-449, 2001.

[67] T. Dobbert, “Matchmoving: The Invisible Art of Camera Tracking,” Sybex, 2005, ISBN 0-7821-4403-9.

[68] The Pixel Farm PFTrack http://www.thepixelfarm.co.uk/product.php?productId=PFTrack

[69] S. Y. Bao and S. Savarese, “Semantic Structure from Motion,” Proceedings of the

IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), pp. 8, 2011.

[70] A. J. Davison, “Real-Time Simultaneous Localisation and Mapping with a Single Camera,” Proceedings of the International Conference on Computer Vision (ICCV), 2003.

[71] G. Klein and D. Murray, “Parallel Tracking and Mapping for Small AR Workspaces,” Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR), 2007.

[72] A. J. Davison, “SLAM++: Simultaneous Localization and Mapping at the Level of Objects.” Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition (CVPR), 2013.

[73] P. Johann and R. Hamböker, “Parametric statistical theory,” Walter de Gruyter, Berlin, DE. pp. 207-208. ISBN 3-11-013863-8. 1994.

[74]M. O. Robin and D. Scott, “Finite automata and their decision problems,” IBM Journal of Research and Development. vol., 3 no.2, pp.114-125, 1959.

[75]E. Rosten and T. Drummond, “Fusing points and lines for high performance tracking,” Proceedings of the IEEE International Conference on Computer Vision (ICCV), vol. 2. pp. 1508-1511, 2005.

[76] B. D. Lucas and T. Kanade, “An Iterative Image Registration Technique with an Application to Stereo Vision,” Proceedings of Imaging Understanding Workshop, pp. 121-130, 1981.

[77] T. Liu, Z. Yuan, J. Sun, J. Wang, N. Zheng, X. Tang, and H. Y. Shum, “Learning to Detect a Salient Object,” IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 33, no. 2, pp. 353-367, 2011.

[78] G. Abdollahian, C. M. Taskiran, Z. Pizlo, and E. J. Delp, “Camera Motion-Based

Analysis of User Generated Video," IEEE Transactions on Multimedia, Vol. 12, No. 1, 2010

[79] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, A. Fitzgibbon, "KinectFusion: Real-time 3D Reconstruction and Interaction Using a Moving Depth Camera," Proceedings of the ACM Symposium on User Interface Software and Technology, pp. 559-568, 2011.

[80] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," International Conference on 3D Digital Imaging and Modeling, pp. 145-152, 2001.

PUBLICATION LIST

Refereed Papers

[1] Hsing-Cheng Yu, Xie-Hong Tsai, An-Chun Luo, Ming Wu, and Sei-Wang Chen, "Study of Three-Dimensional Image Brightness Loss in Stereoscopy," Journal of Applied Sciences, Vol. 5, No. 4, pp. 926–941, June 2015 (SCI). (IF:1.49)

[2]An-Chun Luo, Sei-Wang Chen and Chiung-Yao Fang, "Gaussian Successive Fuzzy Integral for Sequential Multi-decision Making," International Journal of Fuzzy Systems, Vol. 17, No. 2, pp. 321–336, June 2015 (SCI). (IF:1.09)

[3] An-Chun Luo , Wen-Shiou Luo , Wei-Hao Huang, Wen-Chao Chen, "An Auto-stereoscopic Augmented Reality System Using Hybrid Depth Sensing," Pioneer, Vol. 19, pp.33-39, May 2012.

Conference Paper

[1]An-Chun Luo, Kun-Lung Tseng, and Sei-Wang Chen, "A Real-time Camera Match-moving Method for Virtual-real Synthesis Image Composition Using Temporal Depth Fusion," Proceedings of the IEEE International Conference on Optoelectronics and Image Processing, Warsaw, Poland, Jun. 10-12, 2016.

[2]An-Chun Luo, Kun-Lung Tseng, Wen-Chao Chen, Chai-Chen Chen, and Sei-Wang Chen, "Hybrid Depth Generation Method for Auto-stereoscopic Augmented Reality Applications," Proceedings of the IPPR Conference on CVGIP, Yi-Lan, Taiwan, Aug. 2013.

[3] Chia-Ju Lu, An-Chun Luo, Chih-Fan Hsu, and Sei-Wang Chen, "Virtual Director - Real-Time Automatic Shot Selection," Proceedings of the IPPR Conference on CVGIP, Yi-Lan, Taiwan, Aug. 2013.

[4] An-Chun Luo, Wei-Jia Huang, Wen-Chao Chen, ChungWei Lin, and Sei-Wang Chen, “A Stereoscopic Content Analysis System with Visual Discomfort-Aware,” Proceedings of the IEEE International Conference on 3D Imaging, Liège, Belgium, Dec. 3-5, 2012.

[5] Wei-Jia Huang, An-Chun Luo, Wen-Chao Chen, and Wei-Hao Huang, “Perceptual Based Stereoscopic Content Analysis using Salient Information, Dense Disparity Maps, and Modified Random Walk Framework,” Proceedings of the IEEE Workshops on CVPR, Providence, RI, USA , Jun. 16-21, 2012.

[6] Kun-Lung Tseng, Wei-Jia Huang, An-Chun Luo, Wei-Hao Huang, Yin-Chun Yeh , and Wen-Chao Chen, “Automatically optimizing stereo camera system based on 3D cinematography principles,” Proceedings of the 3DTV-CON, Zurich, Swiss, Oct. 15-17, 2012.

[7] Chung-I Li, An-Chun Luo, Chia-Ju Lu, and Sei-Wang Chen, “Automated Lecture Recording System –the Virtual Cameraman Subsystem,” Proceedings of the IPPR Conference on CVGIP, Nantou, Sun Moon Lake, Taiwan, Aug. 2012.

[8] Ya-Yu Kuo, An-Chun Luo, Chia-Ju Lu, and Sei-Wang Chen, “Automated Lecture Recording System,” Proceedings of the IPPR Conference on CVGIP, Nantou, Sun Moon Lake, Taiwan, Aug. 2012.

[9] Chung-Wei Lin, Wei-Hao Huang, An-Chun. Luo, Wei-Jia Huang, and Wen-Chao Chen, “Blur-Based Extending Visual Comfort without Reducing Disparity Range,” Proceedings of the International Conference on 3D Systems and Applications, Hsinchu, Taiwan, Jun. 25-27, 2012.

[10]An-Chun. Luo, Wen-Chao Chen, Chung-Wei Lin, Wei-Hao Huang, and Sei-Wang Chen, “Hybrid Depth Reconstruction for Autostereoscopic Augmented Reality

Applications," Proceedings of the International Conference on 3D Systems and Applications, Seoul, Korea, Jun. 20-22, 2011.

[11] An-Chun Luo, Sei-Wang Chen, and Jung-Min Wang, "A People Counter Using Top-View Video Sequences," Proceedings of the IPPR Conference on CVGIP, Kaohsiung, Taiwan, Aug. 2010.

[12] Chung-Wei Lin, Wen-Chao Chen, Hong-Tu Yu, An-Chun Luo, Fang-Hsuan Cheng, "A Depth Map Reallocation Method for Improving 3D Effect," Proceedings of the International Conference on 3D Systems and Applications, Tokyo, Japan, 2010.

[13] Fu-Jen Hsiao, Chih-Jen Teng, Chung-Wei Lin, An-Chun Luo, Jinn-Cherng Yang, "Dream Home: a multi-view stereoscopic interior design system," Proceedings of The Engineering Reality of Virtual Reality 2010 , SPIE EI2010 in San Jose, USA, 2010.

[14] An-Chun Luo, Wen-Chao Chen, De-Jin Shao, Chung-Wei Lin, "Occlusion size aware multi-viewpoint images generation from 2D plus depth images," Proceedings of Stereoscopic Displays and Applications XXI, SPIE EI2010 in San Jose, USA, 2010.

[15] Cko, C. C., Chen, M. C., Chen, L. Y., Guo, J. N., Liu, J. F., Luo, A. J. "Developing a TriAccess reading environment," Journal of Lecture Notes in Computer Science, vol. 4061, pp. 839-846.

Best Paper Award

[1] Proceedings of the IEEE International Conference on Optoelectronics and Image Processing, Warsaw, Poland, 2016.

[2] Proceedings of the International Conference on 3D Systems and Applications 3D Systems and Applications, Seoul, Korea, 2011.

Summited Paper

[1] Chiung-Yao Fang , An-Chun Luo, Yu-Shan Deng, Chia-Ju Lu and Sei-Wang Chen, "Building a smart lecture recording system," Neural Computing and Applications, Springer, 2017.