IVDA: INTELLIGENT REAL-TIME VIDEO DETECTION AGENT FOR VIRTUAL CLASSROOM PRESENTATION

R. Y. D. Xu[*], J. S. Jin[**], J.G. Allen[***]

Abstract

Audiovisual streaming has been extensively used in synchronous virtual classroom applications. Until recently, content-based processing has rarely being used in real-time streaming. We present in this paper, an intelligent system that uses state-of-art video processing and computer vision technologies that can automatically respond to various video events defined by a set of pre-programmed rules. This intelligent system performs object acquisition, automatic video editing and student multimedia presentation synchronization which can leverage both the capabilities and efficiencies in multimedia streaming for a real-time synchronous virtual classroom. We present detailed discussions of the four major advantages of the system, namely, inexpensive hardware, automation, versatilities and environment adaptabilities as well as natural teaching flow. We describe the system in detail, illustrating the main cutting edge video processing algorithms being incorporated as well as our own research findings in an effort to enhance the performance over the existing algorithms used in virtual classrooms. We also show the implementation of the current prototype system as well as explore its potential in future E-learning applications.

Key Words: video event detection, multimedia streaming, synchronous virtual classroom, video processing, computer vision, E-learning

1. Introduction

The popularity of E-learning is evidently shown by its rapid increase in market value to 26 Billion USD in 2005. Virtual classroom is one form of E-learning, where instructor and students communicate through the network using audio, video and other multimedia teaching materials. The virtual classroom can be classified based on the streaming methods, into synchronous and asynchronous ones.

In asynchronous virtual classroom streaming, video is captured and the editing takes place offline for later presentation. Since there is relatively more time allowed for authoring, many commercial products and literatures (for example, Rowe *et al.* [1] and Ozeki *et al.* [2]) can be found to provide comprehensive editing, indexing and synchronization, even cinematic effect (Gleicher *et. al.* [3]) to the captured instructor video during offline editing.

The synchronous virtual classroom on the other hand streams multimedia to students in real time, allowing bidirectional interactions between instructor and students. There has been an overwhelming number of commercial products and research literatures (e.g. [4]), where static camera(s) are used to allow students to see and hear the instructor in the teaching environment in real-time.

Comparing with asynchronous streaming, there has been much less literature focusing on achieving sophistication in video editing and multimedia indexing in a real-time synchronous virtual classroom. In most cases, classroom video is unmodified or has only been processed minimally before streaming to students. These modifications are usually limited to compression, changing frame rate or resolution.

The factors which constrain real-time authoring methodology from having the same level of sophistication are primarily due to insufficiencies in time and labour required. For example, real-time editing in the field of live TV broadcasting is achieved by a joining effort of a director with a group of editors and cameramen. This is obviously not feasible in common virtual classroom applications. A simple alternative approach is to predefine the multimedia synchronization rules based on a time sequence. This is also not feasible since real time event and human actions can only be approximated beforehand and are very difficult to be followed exactly, i.e., it's difficult to expect an instructor to perform actions based on the predefined sequences without mistakes.

For this reason, recently, emerging research into applying multimodal computer vision (video, audio and other sensory information) techniques to detect real-time event occurrences have been introduced into the context of E-learning. Although computer vision technology even in the foreseeable future will still be primitive compared with a human editor, it provides a solid future direction for intelligent real-time virtual classroom streaming. There are a number of literatures that have been identified:

Bianchi [5] illustrated a system used in an auditorium presentation where a set of static and tracking cameras are used together for the purpose of capturing the slides and the speaker in an automatic fashion.

[*] Faculty of Information Technology, University of Technology, Sydney (UTS) Broadway, NSW 2007, Australia.; email: richardx@it.uts.edu.au
[**] School of Design, Communication & I.T., The University of Newcastle, Callaghan NSW 2308, Australia.; email: jesse.jin@newcastle.edu.au
[***] Faculty of Information Technology, University of Technology, Sydney (UTS) Broadway, NSW 2007, Australia; email: jallen@it.uts.edu.au

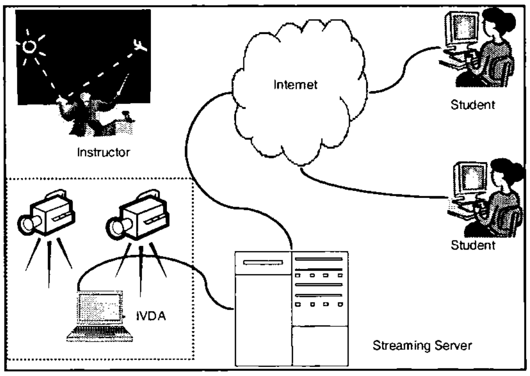


Figure 1. Systematic view of IVDA: Illustrating the relationships between IVDA, streaming server, instructor and students.

Wallick *et al.* [6] and Rui *et al.* [7] illustrated a classroom implemented by a virtual director. The control script described in their paper is used for describing the timeline sequence. .

Shi *et al.* [8] describes smart-camera software used in real-time distance education to perform several recognition tasks. The tasks described are simple and its video processing implementation and evaluation was not discussed. Franklin *et al.* [9] proposed a similar intelligent system where the camera will zoom in close on the writing when the system detects such an instructor action.

Some researchers [10] are applying intelligent real-time video processing before streaming for bandwidth reduction of the exchanged audiovisual data. For example, the chalkboard area can be streamed as a static image and upon refresh once it has been updated by the lecturer.

The remainder of this article is organized as follows, in section 2, we describe our motivations and the uniqueness of IVDA; in section 3 and 4 we describe the system design in detail for both its camera system and computer vision implementation. In section 5 we will describe the end to end prototype system. Finally we will illustrate our immediate and medium term future work in section 6 and 7.

2. IVDA

2.1. Motivation

Apart from traditional video processing for streaming, such as instructor tracking and detecting chalkboard changes in a conventional classroom, IVDA is also aiming to deliver sophisticated event detection where situation and location requires. The examples of such applications are:

Consider a chemistry class; the instructor needs to perform actions between a computer (directing distant students to electronic media), chalkboard (hand writing equations) and in-class laboratory (chemistry experiments). During his chemistry experiment, the instructor wishes the safety procedure of the apparatus he is holding to pop up on the student's PC. This is just one of the many predefined frequently occurring events for this class session. It is difficult for the instructor to signal the system every time on event occurrence, even when he uses voice recognition software.

The second scenario is to setup a virtual classroom before a factory plant's moving production line, which introduces additional approximated occurring events; the instructor wishes the student's PC to be refreshed automatically showing 3D visualization of a manufacturing part as the system detects new parts passing through the production line without the instructor stopping his teaching to signal explicitly.

2.2. Advantages

IVDA has been prototyped to achieve the sophisticated video event detection and synchronized teaching multimedia streaming on the student's PC. Based on our continuing computer vision studies in the relevant areas, IVDA is equipped with the following features:

1. Human and object tracking.
2. Robust real-time recognition (and their pose) of pre-trained objects.
3. Efficient detection of multiple moving regions (background subtraction) on various light conditions and occasional moving background using static cameras.
4. Instructor gesture recognition.
5. Constant video task scheduling to execute several video processing tasks on a single processor high-end PC.
6. Script-based authoring mechanism provides a semantically rich, easy to program and yet comprehensive authoring capabilities to handle real-time events. These scripts are easily portable to suit similar applications.

We are claiming the following four advantages which we argue are crucial for IVDA to broaden its scope in E-Learning applications:

1. **Inexpensive hardware** IVDA's minimum hardware requirements include:
    1.1 Static camera(s) with fixed zooming, two or more are preferable with minimum 640 * 480 resolutions.
    1.2 A single high-end PC for processing.
    1.3 A streaming server is required to support a large number of students.
2. **Automation** IVDA is acting like a virtual TV camera crew, where it incorporates a director, editors and cameramen all into a single software application customized for virtual classroom. This can reduce manual labour cost to almost zero.
3. **Versatilities and environment adaptabilities** The video processing tasks triggered by the predefined scripts can be recognized robustly in real-time. The system is adaptive to different light changes for both indoor and outdoor classroom environments.
4. **Natural instructor teaching flow** IVDA provides instructors with a natural uninterrupted teaching flow, allowing the instructor to concentrate on delivering teaching in a real-time classroom.

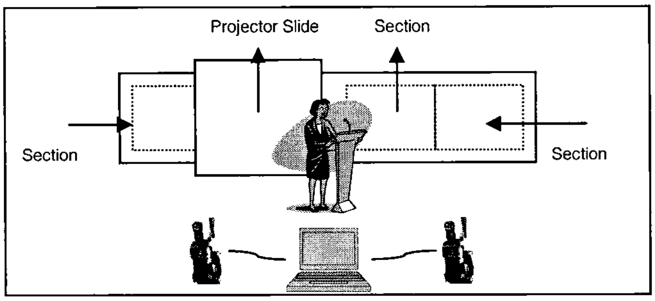


Figure 2. The camera placement shows multiple cameras can be used to cover the entire virtual classroom environment. In addition, the teaching area can be divided into sections for static image streaming.

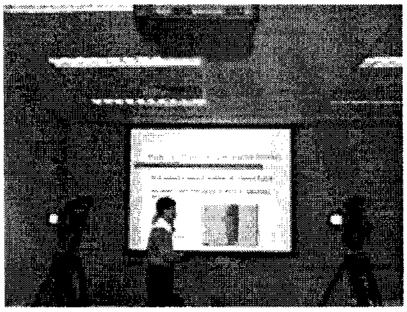


Figure 3. When 2 cameras are used, both are placed facing forward. Some overlap is required for the optional mosaic operation being implemented.

In addition, IVDA can also be used as an instructor teaching training tool. The system can provide effective feedback to the instructor in real-time, based on discrepancies between the instructor's actions with the pre-defined sequence. This feature can be achieved by placing a large size monitor screen to display feedback information.

3. Camera Systems

In an effort to achieve IVDA's first advantage, i.e. cost reduction, we use multiple static cameras to perform software generated camera movements rather than mechanical and optical based such as those used in [5, 6].

The rationale for this approach is that there has been a rapid increase in video camera frame size and drastic reduction in cost. (For example, 640 * 480 CCD web cam cost $100 USD comparing with > $300 3 years ago). On the other hand, the resolution of student's viewing devices has remained relatively stagnant in the last decade. If this trend continues, it is likely that video frames captured at the highest possible resolution may have to be down-sampled in order to display on consumer grade viewing hardware.

We therefore introduce a multiple static camera environment to mimic traditional camera operations. We argue that as the size ratio between camera CCD and PC monitor continues to increase in the future, our approach of using static cameras to simulate camera movement to be even more beneficial.

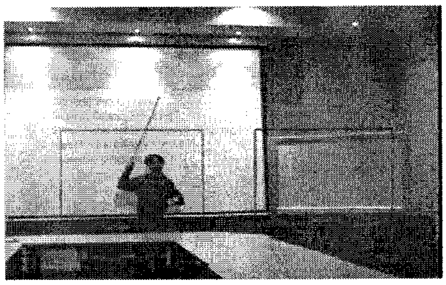


Figure 4. An illustration of software simulation for camera panning.

3.1 Mimic single camera movement from multiple camera views

Multiple cameras are placed horizontally to cover the entire scene shown in Fig. 2 and Fig. 3.

The camera movements in the resulting streaming video can be derived by mimicking the mechanical movement. An illustration of mimic camera panning is shown in Fig. 4. IVDA's predefined script specifies the location of the left and right boxes that indicate where panning starts and ends respectively. These panning positions can also be placed on different camera views and also in any panning directions as required. The number of frames is needed to determine the speed of panning. The resulting video simulates a camera performing a panning operation from one position to another.

In a similar fashion, software generated zooming and object tracking can also be achieved.

The camera systems we have tested include a pair of CCD web cams at 640 * 480 resolutions and PAL DV at 720 * 576 resolutions.

4. Real-Time Video Processing Components

We illustrate the core computer vision techniques that are currently being incorporated. Most of the work has been based on our continuing effort to enhance various areas of video processing technology.

4.1 Robust background subtraction

Background subtraction is used as a pre-step for object tracking, object recognition and gesture recognition where the moving objects are segmented from the background.

Since IVDA will be applied in different classroom environments, we need to have an effective measure against variation in different lighting condition, as well as distinguishing moving background objects (for example, a rotating fan in the classroom) from moving foreground objects.

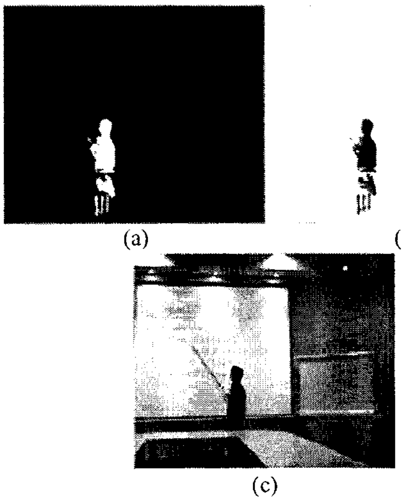


Figure 5. The result of background subtraction: (a) is the binary image; (b) is the segmented foreground; (c) is the original frame.

The commonly referenced paper include Wren *et al.* [11], where the light source variation is handled efficiently by fitting a Gaussian probability density function on each pixel of a running video. For accurate classification when multiple occasional moving background objects are present, Stauffer *et al.* [12] proposed to classify each pixel according to a Gaussian Mixture Model (GMM):

$$P(x_t) = \sum_{i=1}^{k} w_{i,t}\eta(x_t - \mu_{i,t}, \sum{}_{i.t}) \tag{1}$$

where the Gaussians are multivariate in RGB. Each of the K Gaussian is to model one background or foreground object.

IVDA employs both implementations; where [11] is used as the defaulted implementation. The instructor can manually change to the more computational method [12] if required.

Instead of training and classification over the entire video frame, IVDA is performing background subtraction on the down-sampled video frame. We do not notice significant performance degradation compared with the full frame methods. Results were shown in Fig. 5.

4.2 Multiple object localization using mean shift

Video object tracking has been extensively used in IVDA. The tracking is performed after the video frame has been foreground-segmented as described in the previous section.

In order to accurately and efficiently track targets in the lecture video in real-time when several objects (for example, multiple instructors) are present, we use kernel based mean-shift tracking described in [13, 14]. Mean shift is a non-parameterized estimation technique to search for local maxima. Since non-parameter estimation does not assume any underlying probability distribution functions (pdf), kernel density estimation (KDE) technique is use.

The choice of kernel is the famous Epanechnikov kernel:

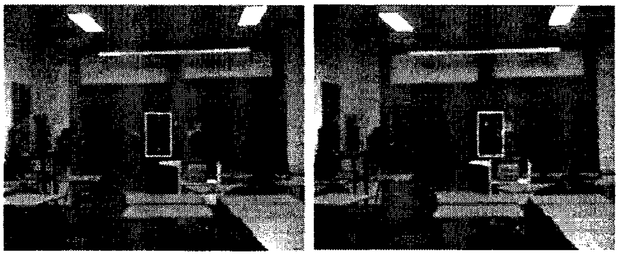


Figure 6. Successful tracking results, with instructor partially occluded behind the projector.

$$Epak(x) = \begin{cases} \frac{1}{2}c_d^{-1}(d+2)(1-x) & \text{if } x \leq 1 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

The tracking process uses the similarity measure between actual target feature pdf **q**, (can be thought of as the pixel features pdf of the moving object at the start of the tracking) with candidate feature pdf **p(y)** (can be thought of as pixel features pdf of potentially correct region centred at **y** in the subsequent frames) is approximated by using Taylor expansion of the Bhattacharyya coefficients:

$$\rho[\mathbf{p}(\mathbf{y}), \mathbf{q}] \approx \frac{1}{2}\sum_{u=1}^{m}\sqrt{p_u(\mathbf{y}_0)q_u} + \frac{1}{2}\sum_{u=1}^{m}p_u(\mathbf{y})\sqrt{\frac{q_u}{p_u(y_0)}} \tag{3}$$

After applying Epanechnikov kernel smoothing and mathematical substitution, the resultant **y**, the centre of the target region is obtained by maximizing the equation:

$$\sum_{u=1}^{nh} w_i Epak\left(\left\|\frac{\mathbf{y}-\mathbf{x}_i}{h}\right\|^2\right) \qquad where \;\; w_i = \sum_{u=1}^{m}\sqrt{\frac{q_u}{p_u(\mathbf{y}_0)}} \tag{4}$$

if pixel feature at location x_i classifies to feature bin u.

The mean shift's basin of attraction property allows this procedure to converge, typically in 4 or 5 iterations.

Although kernel based mean shift is highly efficient, it can easily lose track of its target when there is a large variation between spatial information of the consecutive frames. In IVDA, we applied the variant implementation from our earlier work in Xu *et al.* [15] which is an adaptive approach using fast color thresh-holding similar to [16] and region merging. We have found this to be an effective safeguard measure for mean shift object tracking. Recently, many other variants of mean shift tracking emerges, examples including adaptive background model by Porikli et al. [17] and Principal Component Analysis for the multivariate colour pdf by Han et al. [18]. While we are yet to make formal comparisons, the experiments based on our implementation shows encouraging results.

Fig. 6 shows the successful results when we tested the tracking using lecturer wearing clothes that have similar color distribution to the background, as well as robustness to partial occlusion when lecturer is behind a projector.

For instructor tracking, an adaptive skin colour model, similar to the one used by Zhu et al. [19] has also been used in conjunction with mean-shift tracking to further improve tracking humans with different skin colours.

4.3. Real-time exact object recognition using scale invariant features

One of the key features of IVDA is the ability to identify pre-registered teaching objects in real-time. Unlike general object recognition problems where the aim is to recognize an object from a class containing thousands of similar objects, such as the unsupervised training method in Fergus *et al.* [20], we have simplified the recognition task to identify objects from the exact training images but with different scale and orientation.

The training and recognition process is as follows:

1. The training object is photographed and edited to remove the background, and used as the model image.
2. The invariant features of the model image are computed, shown in Fig. 7.
3. The features are stored in the database.
4. In real-time classroom, the invariant features of the potential video frame region are computed and matched with the images from the database for potential recognition as shown in Fig. 8.

The most important factor for IVDA's robustness in recognition is contributed to the image invariant features we use. We have employed the Scale Invariant Feature Transform (SIFT) methods proposed by Lowe [21].

The SIFT process is known to be invariant to translation, scaling and rotation and is partially invariant to illumination changes and affine or 3D projection.

The SIFT algorithm contains a number of sub-processes to be executed sequentially for a given image. The first process is to detect the peak in the scale-space. A difference of Gaussian (DoG) function is used to detect stable key point locations in scale space. The scale-space extrema is computed from the difference of two nearby scales from the DoG function:

$$D(x, y, \sigma) = (G(x, y, k\sigma)) * I(x, y) \qquad (5)$$

Where k is the scale constant, and $G(x, y, k\sigma)$ is a 2D Gaussian function with variable variance:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \qquad (6)$$

The second process is to calculate the key point localization and orientation assignment illustrated by the length and direction of the arrows shown in Fig. 7.

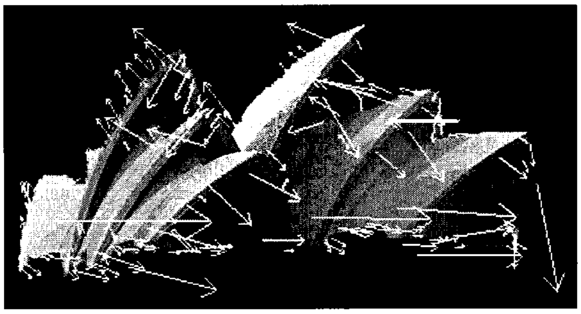
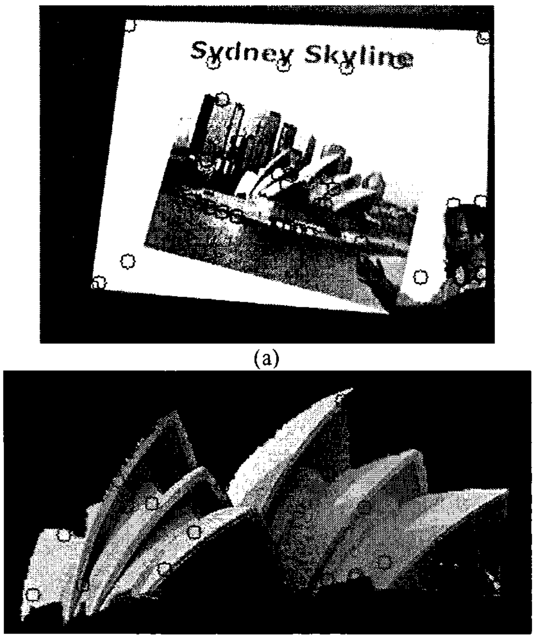


Figure 7. SIFT generation: a training image with background removed and SIFT feature being calculated. The starting point of the arrow is the SIFT point, the length of the arrow indicates the scale factor and the direction of the arrow indicates the orientation.



(a)

(b)

Figure 8: SIFT Matching between video frame (a) and trained model image (b): The image on the poster was deliberately distorted by 10 degrees in orientation. The circles show the matched SIFT keys. The outliers are unstructured. The inliers are structured which can be solved by linear transformation system to indicate a match. We programmed IVDA monitor interface to visualize each matching pair, with the current selected in green.

The final process is to generate the 128-d feature vectors as the key point descriptors.

The object recognition process is completed by firstly inserting the feature vectors from the training images into a database. The matching is performed between the database features with features generated on the current video frame. The matching is performed using Approximate Nearest Neighbors search (ANN).

When a system of transformation equations (which contain translation, scale and rotation parameters) can be solved by least squares method, this then indicates an agreement between poses of the training image with the video frame. Subsequently, the object on this video frame can be classified as a match.

The SIFT process generates feature vectors which robustly describe a video object. More recent approaches of applying Principal Component Analysis (PCA) to SIFT by Ke *et al.* [22] reduce the dimension of SIFT feature vectors dramatically. This method is also used in the current IVDA implementation. The PCA based local descriptors are proved to be more distinctive, more robust to image deformations, and more compact than the standard SIFT representation.

We have further enhanced the performance of IVDA's object recognition implementation in two ways:

1. We have applied SIFT generation to video frame based on the segmented regions instead of the entire video frame.
2. We have also applied Streaming SIMD Extensions 2 (SSE2) on Pentium architecture for faster implementation on a single processor PC.

Both of these enhancements have enabled us to achieve real-time video SIFT matching.

4.4. Instructor gesture recognition

Human gesture recognition has been one of the most challenging tasks in computer vision. Gesture recognition can be applied to many parts of the human body. Using dynamic 2D or 3D models of interrelated shape primitives is a popular approach. However, these methods are computationally expensive, considering that IVDA also needs to perform other video processing tasks on a single processor high-end PC.

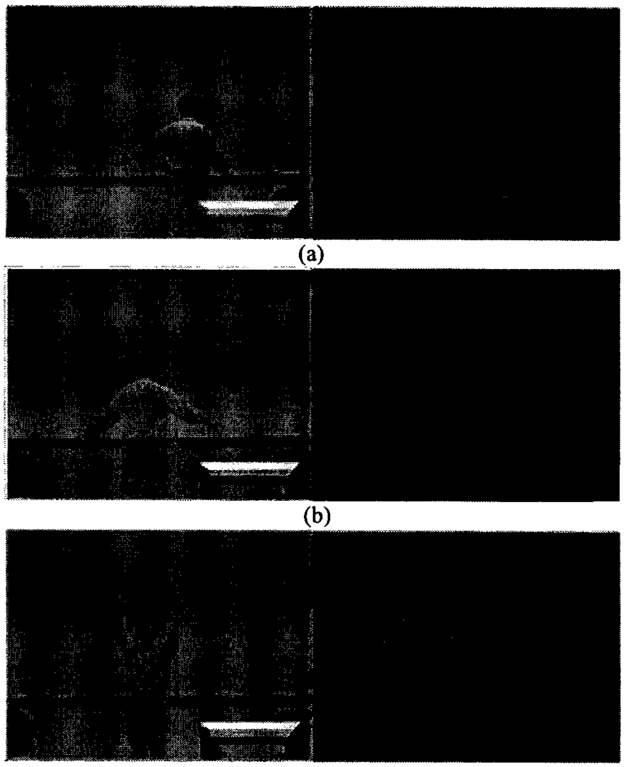
Currently IVDA can recognize action in *facing front*, *facing back*, *writing on board* and *waving hands*. There will be many other gesture recognition functions added in due course.

The instructor gesture recognition used in IVDA is based on [23], where the method uses timed motion history image (tMHI) for representing motion from the gradients in successively layered silhouettes. These methods are sufficient to identify the actions required for IVDA. Fig. 9 shows screen shots of the three captured motion sequences with the corresponding motion history image for turning around from left, turn back from right and waving hands respectively.

The recognition is used in combination with other vision processing techniques in IVDA to train and detect many common instructor actions, for example, the ability to determine the completion of writing on a section of the chalkboard is a combination of instructor's location from tracking and the sequences of gestures.

4.5. Constant frame rate video task scheduler

IVDA requires real-time performance, where each video frame is processed with several of the processing tasks described in this paper.


(a)

(b)

(c)

Figure 9. The tMHI for each of the gestures, the left image is the last frame of the action sequence while the right image is the corresponding motion history image, the direction of the red line indicates the dominant gradient.

Some video tasks have long processing time when performed on our testing PC. They are also asynchronous and do not require completion in a fixed number of frames. (e.g., object recognition spans in order of 30 – 70 frames). We also require IVDA to output video at constant frame rate for streaming.

We proposed a novel approach based on our current research work [24] where we use video content features to schedule tasks, because this scheduling method provides a much more accurate predication of the current and next set of processing times required for a given video task. We can also achieve constant video output in terms of frame rate and resolution while not starving any of the processing tasks. We have employed a two-layered approach in this module, the *scheduling control* and *scheduling delivery* layer respectively, which is illustrated in Fig. 10.

The scheduling control layer generates parameters that are derived from the feature extraction of current video frame processed relevant to a given video task. These parameters are used in conjunction with the past video processing information to form complete dynamic policies. The scheduling delivery layer is the actual implementation of the scheduling mechanisms.

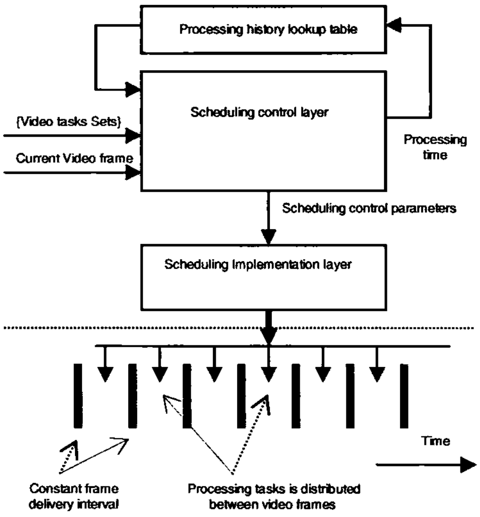

Processing history lookup table
Scheduling control layer
{Video tasks Sets}
Current Video frame
Processing time
Scheduling control parameters
Scheduling Implementation layer
Time
Constant frame delivery interval
Processing tasks is distributed between video frames

Figure 10. The system diagram for constant frame rate task scheduler.

4.6. Predefinition video event programming model

For complex events, we employed simple scripting languages (Java and VB script), which are easy to comprehend by most programmers. An XML based schema is also being studied by another colleague in the lab.

The scripts mostly consist of functions that are driven by generic or custom video events. The following example demonstrates the script syntax:

```
Sub Tracking_RenderVideoTimeInterval(curTimet)
 Me.Tracking_SetRenderDemo(Me.GetPlayType)
 If (curTimet - Form.GetStopTime) >=T_FRAME And
    curTimet Mod T_FRAME = 0 Then
    If(curTimet / T_FRAME) Mod EVENT_N = 0 Then
        Me.SetPlayType TYPE_ONE,
           m_main_buffer, m_static_channel, t_time
    End If
    ...
    Me.SetStopTime(curTimet)
 End If
End Sub
```

Figure 11. The script says on every T_FRAME video frames, we render the output video stream with TYPE_ONE mix type effect where the video streams are from one static window channel and one object tracking channel for *t_time* duration.

Run-time errors within script may occur because the real-time video event is unknown when the script is authored. In light of this problem, we provide the system with two types of error debugging mechanisms. The first is at the compilation stage which irons out most of the user errors (e.g., syntax errors). The second is to catch errors at run-time and to take appropriate just-in-time debugging action. Script can also be added, deleted or modified in real-time in a fashion similar to many commercial debuggers.

Although each video script is application specific, the general mechanism can be applied generically to similar virtual classroom streaming applications. We allow the user to download previously authored script from a similar system and to reapply it with minor local customization. Details of the scripts mechanism can be viewed in our earlier paper described by Xu *et al.* [25].

4.7 Multimedia synchronization on student's PC

We have added support for a synchronized messaging system that allows IVDA to send instructions to a student's computer for synchronized multimedia presentation with the video data. The instructions are encoded in a special text stream. The text stream delivers uncompressed text samples to the student's PC that are interpreted and executed by using IVDA's custom player. Object Linking and Embedding (OLE) technology is used to automate Microsoft® PowerPoint when slide instructions are received, where each student is required to install our customized player before hand.

5. IVDA Prototype System

5.1. System implementation

We have completed prototyping IVDA, and it is currently under laboratory testing.

The system is tested on both medium to high end PCs (Pentium 4 2.2 GHZ, 1G RAM, 128 M Video) and (Pentium M 1.70 GHZ, 1G RAM, 128M Video).

The cameras systems we have tested include:

1. High-end CCD web cam 640 * 480.
2. Medium-end PAL DVCAM (although has optical zooming, but used as a fixed zoom camera) at 720 * 576.

So far, the environment where end-to-end real-time streaming is performed has been indoor classrooms. Future experiments on outdoor and unconventional classroom end-to-end streaming will be conducted once the ordered wireless cameras arrive.

We have tested scripts to perform all the vision functionalities IVDA supports. These tasks include multiple instructor tracking, gesture recognition and object (large size photographs and clustered PC equipments under transformation, rotation and different depth) recognition. Most video events can be detected robustly. The rest of the events can also be detected by performing minor artificial movements (slow down instructor motion or move objects closer to the camera).

5.2. Media streaming implementation

As our current focus is to showcase IVDA video processing capability, we use commercially available technology for media streaming. IVDA's streaming implementation is based on Windows® Media Format 9 Series SDK. We chose the Advanced System Format (ASF) as it is designed primarily for playing synchronized digital media streams and transmitting them over networks. Our current implementation uses manually configured profiles for stream configuration however system profiles could also have been used.

Currently, IVDA streaming is achieved by broadcasting over the HTTP protocol on a given port number for streaming to a small group of students for vision testing. In the immediate future, we will test large scale streaming by pushing data to a publishing point on a Windows® Media Server. Other commercial media stream server technology may also be considered.

Images (static area of the classroom with the moving object subtracted using techniques similar to the ones described in section 4.1), audio and text (multimedia synchronization instructions) are each encoded in a separate stream annotated with an appropriate identifier. On student PC's, streams other than audio and video are interpreted through our customized player. If a different player is used (e.g., Windows Media Player), then the student will view the streams with limited functionality. Therefore, our custom client software is required to install on student's PC, which listens and respond to other streams from IVDA.

We added support for variable video streaming resolution, which now can be any allowable streaming resolution ranging between 320 * 240 to 640 * 480 at 24fps or 30fps across a local area network (LAN). The image stream corresponding to static background also has variable resolution up to the maximum resolution of the capturing camera.

IVDA is written in Visual C++ and incorporates several software development kits (SDK), including Microsoft® DirectX 8.1, Intel® OpenCV Beta 4 and the Microsoft® Windows Media Format 9 Series SDK. The low-level functionality is implemented as a Microsoft® ActiveX control to allow good separation between low-level computer vision functionality with the user interface. IVDA scripting incorporates Windows Scripting Host technology. The scripting and monitoring front-ends are written in Visual Basic 6.0.

5.3. Usability

The IVDA prototype system was evaluated by 3 instructors and 8 students as part of laboratory testing. The systems were installed in different lecture rooms. The streaming scripts were written by authors of this paper.

We worked closely with the instructors, ensuring their classroom requirements have been met. While we have received good feedback from all the instructors, concerns were raised about the training methods for image and gesture, which they feel a comprehensive GUI is required instead of relying on technicians to perform scripting. The students were located in separate rooms connected by a LAN. We receive good feedback from most students.

More usability tests will be performed in future after we have incorporated modifications following the feedback from the initial usability test.

6. Immediate Work

Two enhancements are under construction based on the natural extensions of the current work.

First is to support video detection using a single web-cam at the student's side. It then allows more automatic detection and interaction to be exchanged across both sides of the virtual classroom.

Second is to provide moving camera functionality (with human cameraman) in a dynamic classroom environment. We are currently ordering wireless cameras in order to perform such testing.

One of the problems associated with a moving camera is video stabilization correction as a result of cameraman hand shakes. We have performed extensive research into this field in the past, where an algorithm based on 2.5-d transformation by Jin *et al.* [26] was proposed by the co-author. We have tested several other algorithms; one is similar to Buehler *et al.* [27], which uses an image-based rendering (IBR) technique for non-metric reconstructions. This algorithm does not require the knowledge of camera position or orientation. Other approaches such as the use of an Extended Kaman Filter in Litvin *et al.* [28] are also considered.

7. Conclusion and Future Work

While IVDA is still in its prototyping stage, it has already shown promising results and has the potential to be adopted widely as an intelligent and inexpensive virtual classroom agent used for real-time streaming video. Apart from the two immediate studies in the previous section, much future work is required to leverage the system for commercialization and further enhancing our goals stated in section 2. We believe our continuing research effort in these areas is essential for IVDA to leverage its system performance and have a wide adoption in E-learning.

1. **Continuing research in computer vision** There is no known algorithm in any computer vision field that can perform robustly under all circumstances. Furthermore, even the most state-of-art techniques in the foreseeable future for complex visual scene analysis and event detection will continue to exhibit poor robustness. Therefore continuing research in computer vision is required to keep IVDA up-to-date with the technology advancement in this field.

2. **Intelligent audio support** This includes adding intelligent audio agent support, such as IBM® ViaVoice voice recognition software. We anticipate the introduction of multimodal audio-visual event detection to provide more robust results in event recognition. However the fusions of the multimodal cues present a challenging task ahead which will require substantial research effort.

3. **More research in streaming technology** The main contribution of IVDA is in the area of real-time video editing and multimedia synchronization. Our streaming technology is however based on commercially available SDKs. We also assume reliable network bandwidth since all our experiments were performed across a LAN. Much research in the area of multicasting and unreliable networking is required before applying the system commercially.

4. **Additional multimedia synchronization on student's PC** At the moment, we provide limited multimedia synchronization capabilities on the student's PC. More programming effort is needed to ensure comprehensive function rich synchronized multimedia accesses between instructors and students.

Acknowledgement

This project is supported by Australia Research Council SPIRT Grant (C00107116).

We would like to express our thanks to Yan Ye and Massimo Piccardi for proving us with a copy of PCA-SIFT source code and implementation source code for various background subtraction algorithms.

References

[1] L.A. Rowe, D. Harley, P. Pletcher, and S. Lawrence, *BIBS: A Lecture Webcasting System*, Berkeley Multimedia Research Center, TR 2001-160, Jun, 2001.

[2] M. Ozeki, M. Itoh, H. Izuno, Y. Nakamura, Y. Ohta, *Object Tracking and Task Recognition for Producing Interactive Video Content -- Semi-automatic Indexing for QUEVICO*, Proceedings of the Knowledge-Based Intelligent Information & Engineering System, UK, 2003, 1044-1053.

[3] M. Gleicher, R. Heck and M. Wallick, *A framework for virtual videography*, Proceedings of the 2nd international symposium on Smart Graphics, 2002, pp.9-16.

[4] S. Deshpande and J.N. Hwang, *A Real Time Interactive Virtual Classroom Multimedia Distance Learning System*, IEEE Transactions on Multimedia, vol. 3, num. 4, Dec, 2001, pp. 432-444.

[5] M. Bianchi, *AutoAuditorium: "a fully automatic, multi-camera system to televise auditorium presentations"*, Proceedings of the Joint DARPA/NIST Smart Spaces Technology Workshop, July 1998.

[6] M.N. Wallick, Y. Rui, and L.W. He, *A Portable Solution for Automatic Lecture Room Camera Management, Proceedings of ICME 2004.*

[7] Y. Rui, A Gupta and J. Grudin, *Videography for Tele-presentations*, Proceedings of the Conference on Human factors in computing systems, pp.457 - 464, 2003

[8] Y. Shi, W. Xie, G. Xu, *Smart Remote Classroom: Creating a Revolutionary Real-Time Interactive Distance Learning System*, Proceedings of the International Conference on Web-Based Learning, Hong Kong, Aug, 2002.

[9] D. Franklin and K. Hammond, *The Intelligent Classroom: Providing Competent Assistance*, Proceedings of the International Comference on Autonomous Agents (Agents-2001), 2001.

[10] C. H. Lin, T. Lv, W. Wolf, and I. B. Ozer, *A peer-to-peer architecture for distributed real-time gesture recognition*, Proceedings of the International Conference on Multimedia and Exhibition, IEEE, 2004.

[11] C. Wren, A. Azarbayejani, T. Darrell, and A.P. Pentland, *Pfinder: real time tracking of the human body*, IEEE Trans. on Pattern Anal. and Machine Intell., vol.19, no. 7, pp. 780–785, 1997.

[12] C. Stauffer, W.E.L. Grimson, *Adaptive background mixture models for real-time tracking,* Proceedings of the IEEE CVPR 1999, pp. 246–252.

[13] D. Comaniciu, V. Ramesh, *Mean Shift and Optimal Prediction for Efficient Object Tracking*, IEEE Int. Conf. Image Processing (ICIP'00), Vancouver, Canada, Vol. 3, 70-73, 2000.

[14] D. Comaniciu, V. Ramesh, P. Meer, *Kernel-Based Object Tracking*, IEEE Trans. Pattern Analysis and Machine Intelligence, 25(5): 564-575, 2003.

[15] R. Y. D. Xu, J.G. Allen, J.S. Jin, *Robust Mean-shift Tracking with Extended Fast Colour Thresholding*, Proceedings of the 2004 International Symposium on Intelligent Multimedia, Video & Speech Processing, HongKong, Oct, 2004, pp.542-545.

[16] J. Bruce, T. Balch, & M. Veloso (2000), *Fast and Cheap Color Image Segmentation for Interactive Robots*, Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '00), Vol. 3, Oct, 2000, pp. 2061 - 2066.

[17] F.M. Porikli, O. Tuzel, *Fast Object Tracking by Adaptive Background Models and Mean-Shift Analysis*, International Conference on Computer Vision Systems (ICVS2003), Apr, 2003.

[18] B. Han, L. Davis., *Object Tracking by Adaptive Feature Extraction* , Proceedings of the ICIP, Dec, 2004.

[19] Q. Zhu, K.T. Cheng, C.T. Wu , Y, L, Wu, *Adaptive learning of an accurate skin-color model*, Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, May, 2004 pp. 37- 42.

[20] R. Fergus, P. Perona, and A. Zisserman, *Object Class Recognition by Unsupervised Scale-Invariant Learning*., Proceedings of the IEEE Conf on Computer Vision and Pattern Recognition, 2003

[21] D. Lowe (2004) *Distinctive image features from scale invariant key points*, International Journal of Computer Vision, 60, 2 (2004), pp. 91-110.

[22] Y. Ke, R. Sukthankar. *PCA-SIFT: A More Distinctive Representation for Local Image Descriptors*, Proceedings of the Computer Vision and Pattern Recognition Washington, D.C, 2004, 506-513.

[23] G. Bradski and J. Davis, *Motion Segmentation and Pose Recognition with Motion History Gradients,* Machine Vision and Applications (2002) 13: 174–184.

[24] R. Y. D. Xu, J. G. Allen, J. S. Jin, *Constant video frame control under multiple periodic content dependant video processing*, Proceedings of the Pan-Sydney Workshop on Visual Information Processing (VIP2004), Dec, 2004.

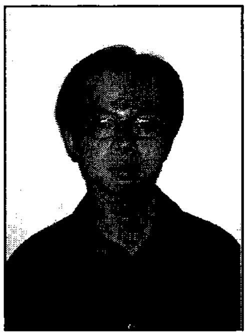
[25] R.Y.D. Xu, J.S.Jin, J.G. Allen, *Framework for Script Based Virtual Directing and Multimedia Authoring in Live Video Streaming*, Proceedings of the 11th International Multi-Media Modelling Conference (MMM2005), Melbourne, Australia, 2005.

[26] Jesse S. Jin, Zhigang Zhu, Guangyou Xu, *Digital Video Sequence Stabilization Based on 2.5D Motion Estimation and Inertial Motion Filtering*, Real-Time Imaging, Vol. 7, No. 4, August 2001, Academic Press, pp. 357-365.

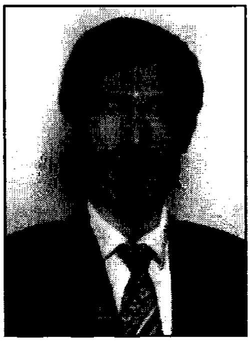
[27] C, Buehler, M, Bosse, L, McMillan, *Non-Metric Image-Based Rendering for Video Stabilization*, Proceedings of the CVPR 2001, pp.609-614.

[28] A. Litvin, J. Konrad and W. C. Karl, *Probabilistic video stabilization using Kalman filtering and mosaicking*, Proceedings of the of SPIE Conference on Electronic Imaging, Santa Clara, CA, 2003.

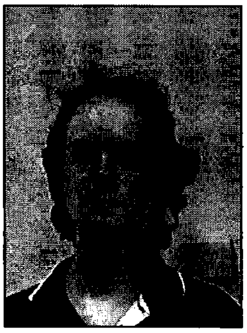
Biographies



Richard Y. D. Xu received Bachelor of Engineering (Computer Engineering) in 2000 from University of NSW, Australia. After shortly worked in IT industry for 1.5 years as an application software developer; he commenced PhD studies in interactive videos and computer vision at University of Sydney and University of Technology, Sydney (UTS). Richard is now a final year PhD student whom has authored and co-authored 12 conference and journal papers. Richard's research interests include object tracking, object recognition, invariant video features, pattern recognition, image watermarking and video standards.



Jesse S. Jin graduated with a B.E. from Shanghai Jiao Tong University, M.E. from CTU and a Ph.D. from University of Otago, New Zealand. He is the Chair Professor of IT in the School of Design, Communication and IT, University of Newcastle, and holds an adjunct appointment in University of New South Wales, University of Sydney and University of Technology, Sydney. He has published 175 journal and conference articles and 12 books. He also has one patent and is in the process of filing 3 more patents. He established a spin-off company and the company won the 1999 ATP Vice-Chancellor New Business Creation Award. He is a consultant of many companies such as Motorola, Computer Associates, ScanWorld, Proteome Systems, HyperSoft, etc. He was a visiting professor in MIT, UCLA, HKPU, Tsinghua University and China Academy of Sciences.



John G. Allen received a Bachelor of Science (Honours) in 2001 from University of Sydney, Australia. After working for half year as a research assistant he started his PhD in Computing Science. John is now a final year PhD student at the University of Technology, Sydney whom has authored and co-authored 10 conference papers. John's research interests include compilation for SIMD hardware, compilers, multimedia, video processing.