

Gesture and Speech for Video Content Navigation

Gary Bradski, Boon-Lock Yeo, Minerva M. Yeung

Microcomputer Research Labs

Intel Corporation,

RNB 6-35,

2200 Mission College Blvd.,

Santa Clara, CA 95052-8119.

Email: {gary.bradski, boon-lock.yeo, minerva.yeung}@intel.com

Abstract

This article describes ongoing research in the use of computer vision gesture and speech recognition techniques as a natural interface for video content navigation, and the design of a navigation and browsing system that caters to these natural means of computer-human interaction. For consumer applications, video content navigation presents two challenges: (1) how to parse and summarize multiple video streams in an intuitive and efficient manner, and (2) what type of interface will enhance the ease of use for video browsing and navigation in a living room setting or an interactive environment. In this paper, we address the issues and propose the techniques that combine video content navigation with gesture and speech recognition, seamlessly and intuitively, in an integrated system. We present a new type of browser for browsing and navigating video content, as well as a gesture and speech recognition interface for this browser.

Keywords: Computer Vision, User Interface, Video Content Navigation, Video Browsing, Gesture Recognition, Speech Recognition.

1. Introduction

The amount of video content is exploding due to the growth of multi-channel satellite and cable TV, push technologies over the web, the proliferation of digital video cameras on connected PCs and the advent of digital television. Providing the user with the ability to preview the content, gain fast and intuitive access to programs or segments of interest, and carry out efficient search and retrieval of video content can enhance the utilization and enjoyment, in other words, add to the value of content. Methods of summarizing and browsing this video content have been active areas of research [1,2]. Section 2 discusses our proposed video content navigation approach.

Yet, for consumer applications in real settings, it is not enough to have capable video content navigation technologies. A workable interface is also needed. For example, it is problematic to use a mouse as a pointing device while sitting on a couch or easy chair. Keyboards

are cumbersome too in such settings. Without a table or surface at hand height, both the mouse and keyboard can be difficult or tiresome to use. Connection to the computer, device power, dust and spills can also be problematic. An unfettered, intuitive and natural interface can help solve these problems. Speech and gesture recognition could form the basis of such an intuitive, natural and unfettered interface.

In section 3, we have combined “off the shelf” speech command recognition [5] with our own computer vision gesture recognition techniques to control our video content browser. There is much new work in gesture recognition for computer interface [6,7,8]. For our purposes, we need to develop rapid and efficient gesture recognition techniques so that the CPU is not too burdened to run the video browser. We also need to focus on gestures that will work intuitively with our video content browser. Thus we concentrate on rapid methods of recognizing waving, pointing, halting and grabbing type gestures to control video browsing. The integration of the gesture system and the browsing system is discussed in section 4.

2. Video Content Navigation

While digital video offers the promises of more powerful playback mechanism, existing playback tools allow only simple VCR-like functionality. This means that one has to sequentially view the individual frames of candidate video clips before zooming in on segments of interests. A desirable provision is to allow a user to go through a video sequence or navigate across different video sequences in a manner as one would browse through a book. In doing so, he or she would flip through the pages to get quick ideas of the content, and then slowly focus on particular chapters or sections of interest. Browsing of video should serve a similar purpose, i.e., as a form of rapid information filtering.

Video browsing should be made available at different granularities and structured in accordance with the hierarchy of video organization. At the lowest level, browsing can be performed by a frame-by-frame sequential playback. This gives the most details and yet is the most inefficient. Higher level abstractions for browsing can be achieved at the shot and scene levels. The segmentation and analysis will allow us to build visual summaries that capture the content and

structure of the underlying story for intuitive understanding and video navigation from one segment to another.

In our browsing system, a video summary of content is built from the processing and analysis of video content, which then offers the indices into meaningful units and provides a high-level visual overview of the story. The visual summary then forms the basis for browsing.

The fundamental unit of film and video is a shot, each of which is captured between a “record” and “stop” camera operation. At the next level, a scene is composed of a collection of shots depicting the same locale or event. Since films and TV programs are composed of shots, scenes and stories, it is best that the units automatically extracted reflect such a hierarchy of video. Video processing therefore seeks to automate, or at least assist, the identification of meaningful units of video and the characterization of these units.

Detecting Shots

Shot detection is the process of automatic detecting of transitions from one shot to the other. There are two types of shot boundaries: the abrupt and gradual. The basic principle to locate shot boundaries is to compare successive frames and determine where there are significant differences. Abrupt transitions most often give rise to large differences in both the luminance values and color histograms of video frames, thus are easier to locate. Gradual transitions like fade-in and fade-out are more difficult to detect, not only due to the absence of “significant” differences, but also because the boundaries may not be precisely located. Techniques to determine the boundaries can be based on global thresholding or relative thresholding in a local window of frames. In addition, the detection can be done directly on compressed video without full decompression. We use the technique proposed in [3]. Compressed video processing can be achieved by fast extraction of spatially reduced image sequence from MPEG compressed video. Each DC image is obtained from block-wise averages of 8x8 block of an image. The extraction of DC images does not require full decompression of the compressed video and thus computationally very efficient. We tested the shot detection on compressed MPEG-1 sequence on an Intel® Pentium™ II 400MHz PC, and the entire processing took 3 seconds for a 30-second video clip coded at a resolution of 352x240 and encoded at 1.5 Mbits/sec. An illustration of the results of shot detection is shown in Figure 1.



Figure 1: Detected shots (illustrated)

Finding Scenes

In the next level, we analyze the shots to aggregate a collection of consecutive shots into meaningful scenes. We use a clustering process called *time-constrained clustering* [4] developed earlier. In time-constrained clustering, shots of similar visual content are grouped into the same cluster and share the same label, provided that they are not too far apart from each other in time. In other words, it takes into account both visual characteristics and temporal locality of shots. We then use the Scene Transition Graph (STG) [2] to model the clusters as nodes and the transitions in time as edges. The STG, when combined with time-constrained clustering, allows the segmentation of meaningful story units through the identification of the cut-edges in the directed graph. The cut-edges reflect the transition from one story (scene) to the next. A story unit closely approximates a scene. The results of the segmentation can be visually represented using the STG that presents the flow of story units from one to the next for navigation. In addition, the segmentation helps to build visual summary representation beyond the STG in our approach. Scenes and shots are thus structured in a hierarchy of abstraction which the user interacts with by gesture and speech as described below.

3. Gesture Recognition

A recognition system that is intended to work as a computer interface must be fast enough to work at or near video frame rate in order to be responsive to the user. The recognition system must also be efficient enough to only absorb a small fraction of the computational resources in order to allow other processing to take place. For gesture recognition, we employ a combination of tracking features or objects as well as possibly recognizing motion [8] or movement energy patterns [10]. In [9], a face tracking system (CAMSHIFT) is described that meets the criteria of operating at video frame rate while only absorbing a small fraction of system computational resources. This tracking system uses a statistically robust method of finding the mode of a probability distribution to find and track the flesh color region produced by faces in video images. To extend this method we seek a way of converting features and objects in video images into probability distributions. This can be done as follows similar to [11]: We decide on a local measurement set M (for example the output of filters, color detection, edge orientations etc) taken from images of an object o_i . A histogram of these measures from many view of the object is taken yielding:

$$H(M | o_i) \quad (1)$$

A rule of thumb is that we want at least as many training samples as there are cells in the histogram. If we normalize the histogram by the number of training samples we get an approximation of the probability density:

$$p(M | o_i) \quad (2)$$

For tracking and recognition, what we really want is $p(o_i | M)$. To get this for a single measure m_j , we use Bayes rule:

$$p(o_i | m_j) = \frac{p(m_j | o_i)p(o_i)}{\sum_k p(m_j | o_k)p(o_k)} \quad (3)$$

If we assume that all objects are equally likely [$p(o_i) = p(o_j)$], and we space our local measurements at least a local measure distance apart so that the measures are independent [$p(ab) = p(a)p(b)$], we get

$$p(o_i | m_1, \dots, m_n) = \frac{\prod_j p(m_j | o_i)}{\sum_k \prod_j p(m_j | o_k)} \quad (4)$$

By this method we can produce probability distributions of features or objects in an image and then use CAMSHIFT [9] to track the object in a statistically robust way. Tracked features such as the hands, face and torso are attached to a five jointed, “2D+” skeletal stick figure model. It’s “2D+”, because the model allows crossover as shown in Figure 2.

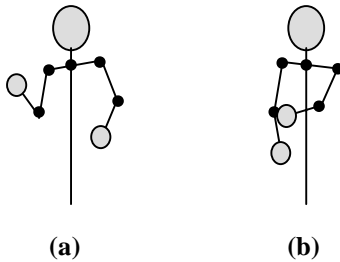


Figure 2: (a) 2D+ skeletal stick figure model of a human from the torso on up with five joints; (b) 2D'+ because this model allows crossover.

In this way, up/down and left/right waving motions can be recognized. To separate up from down and left from right gestures, we take histograms via equation (1) of motion flow regions [8] or motion templates [10]. We then use the resulting learned models in equation (4) to recognize whether an up/down motion is up or down, and whether a left/right motion is left or right. In the same way, a cutting motion across the neck can be detected as a gesture.

Using the skeletal model and the static visual recognition histograms, we can also recognize:

- (1) A flat, open motionless hand held in front of the body (stop).
- (2) A motionless fist held in front of the body (activate).

Speech recognition [5] is also used to supplement the gestures and for additional commands as described below.

4. The Browsing System

Video Navigation Interface

The visual summary constructed facilitates video access. However, the visual representation in the form of graph may not be most suitable for accommodating natural interface like gesture recognition as the manipulation of graph requires more sophisticated commands and incur more computations and heuristics in computer vision to recognize difficult gestures. Towards this end, we propose linear (temporal) representations of shots and scenes for real-time “recognizable” gestures that emulate simple commands, and redesign the browser interface to accommodate the gesture input, and add special features that work around the limitations of gesture recognition. The browser design, combined with gesture and speech inputs, provides hierarchical access to video: at the top level, visual summary of major stories can serve as pictorial indices into individual stories, and gesture input becomes the navigation tool surfing across the main story-line; at the next level, individual shots can be presented. The users can command the playback, stop, rewind, and fast forward at different granularities and in return not only are able to get an overview of the story, but also have more efficient access time. An illustration of the concept is presented in Figure 3.



Figure 3: An illustration of integrating gesture and speech recognition into a smart browsing system

The Commands

We use the following gestures to the video browser:

- (1) Waving right, left, up and down.
- (2) Cutting motion across the neck . (Mute)
- (3) Stop (flat, open motionless hand).
- (4) Activate/Go (face on motionless fist).

The following speech commands are also used:

- (1) Ignore gestures.
- (2) Watch me. (Follow gestures).
- (3) Mute.
- (4) Go.
- (5) Stop.
- (6) On.
- (7) Off.

Details and further progress will be discussed at the poster session.

0 Reference:

1. B. L. Yeo and M. M. Yeung, "Retrieving and Visualizing Video", Communications of the ACM, pp. 43-52, Dec. 1997.
2. M.M. Yeung, B.L. Yeo, W. Wolf and B. Liu, "Video Browsing using Clustering and Scene Transitions on Compressed Sequences", SPIE Vol. 2417 Multimedia Computing and Networking 1995 , pp. 399-413, Feb. 1995.
3. B.L. Yeo and B. Liu , "Rapid Scene Analysis on Compressed Videos", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 5, No. 6, pp. 533-544, December 1995.
4. M.M. Yeung and B.L. Yeo , "Time-constrained Clustering for Segmentation of Video into Story Units", Proceedings, International Conference on Pattern Recognition, Vol. C, pp. 375-380, August 1996.
5. IBM Via Voice Gold TM
6. K. Mase and R. Kadobayashi, "Gesture Interface for a Virtual Walk-through", Workshop on Perceptual User Interfaces, Oct. 1997, pp 20-21.
7. M. Lucente, "Visualization Space: A Testbed for Deviceless Multimodal User Interface", Intelligent Environments Symposium, AAAI Spring Symposium Series, March 23-25, 1998.
8. R. Cutler and M. Turk, "View-based Interpretation of Real-time Optical Flow for Gesture Recognition", Int. Conf. on Automatic Face and Gesture Recognition, 1998, pp 416-421.
9. G.R. Bradski, "Computer Vision Face Tracking For Use in a Perceptual User Interface", Intel Technology Journal, Q2 1998, http://developer.intel.com/technology/itj/q21998/article_s/art_2.htm
10. J.W. Davis and A.F. Bobick, "The Representation and Recognition of Action Using Temporal Templates", CVPR'97, pp 928-934.
11. B. Schiele and J.L. Crowley, "Recognition without Correspondence Using Multidimensional Receptive Field Histograms", M.I.T. Media Lab. Perceptual Computing Technical Report Number 453.