Eyes in the Interface

Francis K.H. Quek
Electrical Engineering and Computer Science Department
The University of Illinois at Chicago
Chicago, IL 60607
Correspondence: quek@eecs.uic.edu

July 7, 1995

Abstract

Computer vision has a significant role to play in the human-computer interaction (HCI) devices of the future. All computer input devices serve one essential purpose. They transduce some motion or energy from a human agent into machine useable signals. One may therefore think of input devices as the 'perceptual organs' by which computers sense the intents of their human users. We outline the role computer vision will play, highlight the impediments to the development of vision-based interfaces, and propose an approach for overcoming these impediments. Prospective vision research areas for HCI include human face recognition, facial expression interpretation, lip reading, head orientation detection, eye gaze tracking, three-dimensional finger pointing, hand tracking, hand gesture interpretation, and body pose tracking.

For vision-based interfaces to make any impact, we will have to embark on an expansive approach which begins with the study of the interaction modality we seek to implement. We illustrate our approach by discussing our work on vision-based hand gesture interfaces.This work is based on information from such varied disciplines as semiotics, anthropology, neurophysiology, neuropsychology and psycholinguistics. Concentrating on communicative (as opposed to manipulative) gestures, we argue that interpretation of a large number of gestures involves analysis of image dynamics to identify and characterize the gestural stroke, locating the stroke extrema in ordinal 3D space, and recognizing the hand pose at stroke extrema. We detail our dynamic image analysis algorithm which enforces four constraints: directional variance, spatial cohesion, directional cohesion and path cohesion. The clustered vectors characterize the motion of a gesturing hand.

1 Introduction

The evolution of the computer, seen from the vantage point of human-computer interaction (HCI), may be measured by the *location* of the interface along a human-machine continuum. At one end of this continuum, the human takes full responsibility for any interaction. Seminal computer systems of the past which required humans to activate bootstrap mechanisms by toggling micro-switch sequences by hand lie near this end of the continuum. At the other end of the continuum lie the fully sentient machines of science fiction which can surmise uncommunicated intentions, resolve ambiguous situations through discourse, form generalizations of user desires from specific observations and deduce specific responses from knowledge of general user profiles. One could conceive of the computer's journey along this continuum as an evolution toward the real world inhabited by its creators. Within this conceptual framework, every technological advancement endows the computer with the ability to adapt more to how humans do things as opposed to what is dictated by the limitations of technology. Advances in computation speed allow the computer to devote more time to interface concerns. Advances in memory technology permit more storage resources to be dedicated to graphic screen management, sound synthesis and speech recognition. Advances in video and graphics technology facilitate the implementation of various interface paradigms such as two and three dimensional metaphoric interaction spaces, virtual reality interaction and a plethora of information visualization or presentation approaches. This paper investigates the contribution of computer vision technology to this computer evolution. If computers could see, what would change in the way we work with them?

Computer graphics, software engineering, database technology, operating systems, and artificial intelligence research have all made their mark on the human-computer interface. Computer vision, on the other hand, has not made much of an impact on the human-computer interface after approximately three decades of research. Computer vision research with an eye to human-computer interaction is sorely lacking, and requires redress. We shall discuss the impediments to computer vision playing a role in human-computer interaction and advance the application of computer vision as a mechanism for human input. We shall discuss vision-based computer input in general and concentrate on the visual recognition of hand gestures as a concrete example.

2 Vision-Based Input

All computer input devices serve one essential purpose. They transduce some motion or energy from a human agent into machine useable signals. One may therefore think of input devices as the 'perceptual organs' by which computers sense the intents of their human users.

2.1 Computer Input Devices

The evolution of computer technology has seen a corresponding evolution of computer input and human-computer interaction technologies. In this evolution, the interface has steadily moved from a being centered around the computer to being centered around the human. At first, humans had to throw particular switches in precise sequence and prepare punch cards and paper tapes to reduce the interface load on the machine. As machines became more capable, they were able to read keyboards and provide 'realtime' feedback on tele-type terminals. For a time, video terminals patterned after tele-type terminals (with 80 columns and 24 lines of printout that scrolls into oblivion) were ubiquitous in computing. Recently, advances in memory and computation technology has permitted machines to enter the era of graphical interfaces with a proliferation of two-dimensional pointing devices, screen metaphors and graphical user interfaces. In each of these stages of evolution, the computer has had to transduce human actions into data streams amenable to machine interpretation.

User input devices may be divided into those which provide symbol input such as keyboards and voice recognition and those which provide spatial information such as mice, digitizing tablets, joysticks, touchscreens and trackballs. With the popularization of the graphical user interface, the mouse has proliferated to become the defacto standard graphical input device. Most recently, pen-based computers have made inroads into the marketplace. Such devices track the motion of a stylus on a flat display which provides immediate graphical feedback. These devices facilitate symbolic data entry by means of character recognition and *pen gestures* and spatial input.

2.2 The Prospect of Vision-Based Input

The spatial input devices discussed in the previous paragraph work in the two-dimensional world and provide graphical pointing within a computer screen. They will work with limited success when interaction moves beyond the the confines of the computer screen to wall-sized displays and three-dimensional environments. The recognition that humans operate constantly in a three-dimensional (3D) world, has already prompted such promising innovations as virtual reality, 3D visualization and 3D interaction with problem spaces. Xerox PARC's *3D/Rooms* and *Information Visualizer* [1] user interface metaphors, for example, exploit this human 3D spatial cognition. In systems which facilitate collaboration among humans, the collaborators may interact with tools (such as electronic 'blackboards') which exploit the user's sense of space for locating each other and artifacts in the collaborative workspace. In telerobotic control systems, existing 3D control devices (which are typically custom tailored to particular robot configurations) often require considerable manual dexterity to control multiple degrees of freedom. For such paradigms to be effective, new input methodologies which permit users to communicate 3D information are needed. Avant garde input devices which operate in 3D include gloves which detect hand configurations and orientation, a variety of 3D joysticks, head trackers, eye-gaze trackers and 3D styluses. This is where we believe computer vision has a significant role to play on the stage of computer input. All the 3D devices mentioned require the user to be instrumented with encumbering devices. Such encumberances are further exacerbated by the fact that the devices are typically tethered with clumsy cables. The application of computer vision to sense human communication unobtrusively will enable users to interact with computers in a truly natural fashion. We envision the day when machines will acquire input from human users the same way as other humans do – by observing, hearing and feeling (tactile sensing).

Already, seminal computer vision research is being performed in human face recognition [2, 3], facial expression interpretation [4, 5, 6, 7, 8], lip reading [9, 10, 11, 12], head orientation detection [4, 13, 14], eye gaze tracking [15, 16], three-dimensional finger pointing [17], hand tracking [18, 19], hand gesture interpretation [20, 21, 22, 23] and body pose tracking [24, 25, 26, 27]. The National Science Foundation of the United States of America has recently concluded workshops on facial expression recognition [28] and facial animation [29]. Each of these areas have potential roles to play in the way humans and computers interact in the future. Face recognition will permit machines to be customized to the preferences and characteristics of each user without explicit 'logins' by each user. This will be particularly useful in systems designed for cooperative work where multiple users share an electronic and physical workspace, or for access control by computer systems embedded in automobiles, home security systems and communication systems. Facial expression interpretation will permit machines to detect a user's puzzlement or frustration in using a system and provide more assistance. Facial expression may also be a means for the physically impaired to communicate with and through a computer. Lip reading provides another channel of input for speech recognition. This will facilitate the development of robust speech recognition-based computer interfaces. Lip reading interfaces may also prove effective as interfaces for the speech impaired. Models developed for representing facial expressions and lip movement may also be used to build multimedia documents which are able to represent affect. While today's documents are better at representing facts than emotions, documents of the future may permit users to embed affect markers. A computer system may synthesize a face which reads the document with the appropriate affect. Body pose tracking will facilitate the development of unencumbering immersive virtual reality systems. Users may move freely within virtual environments, reach out and touch objects, conduct virtual symphonies etc. All these areas deserve infusion of research effort and resources. For this paper, we shall highlight the area of vision-based hand gesture interpretation to illustrate the research approach and philosophy which we believe will prove fruitful.

3 Impediments to Computer Vision in HCI

Several issues conspire to impede the application of computer vision to the human-computer interface. These may be divided into two categories: technological challenges and research approach.

Technological challenges have yet to be met in the areas of speed, cost and robustness. Computer vision systems must process voluminous amounts of data at very high rates. Data compression techniques may alleviate data transmission and storage problems, but computer vision requires processing of the underlying data at resolutions appropriate to the desired interpretation. The real world is dynamic and three-dimensional. Taking just a monocular video stream at a 640×400 resolution at 30 frames per second, a system would have to process 7.5 million pixels every second. Most computer vision approaches involve feature extraction and matching processes. The combinatorics inherent to such processes add to the computation cost. The upshot of this processing load is that computer vision is often performed in a 'batch' mode computing paradigm, taking an input image or sequence of images, processing for a while, and then producing an output. Human-computer interaction processing requirements, on the other hand, are response-time critical. The perception of 'responsiveness' is essential to the implementation of a user interface. The most of effective human-interaction strategies may be rendered frustratingly clumsy by time lags and processing latency.

Related with the speed impediment is that of cost. Until recently, the idea of committing vision hardware and software to the human-computer interface was unthinkable. In a world where some considered even pointer driven graphical user interfaces extravagant for 'ordinary users', interfaces incorporating computer vision was a prohibitively costly luxury.

A third challenge to the application of computer vision to the human-computer interface is inherent to vision algorithms. Human-computer interfaces are required to operate under conditions varying illumination, image background complexity and user-particular variations. Computer vision systems are typically sensitive to these same conditions and are robust only within specific domains and fairly constrained environments.

Compounding the technological impediments are issues of research approach and resource dedication. Until recently, the few intellectual resources and research programs in computer vision have been allocated to the interface. Research efforts tended to emphasize 'autonomous' machines rather than interactive ones. Systems which emphasized real-time response were typically in the domain of 'robot' vision to provide input and feedback to robots for such operations as pick-and-place and navigation. Such systems have markedly different requirements than systems designed with a high degree of human interactivity in mind.

The issue of computational speed is partially addressed by advances in computational hardware and memory. Desktop machines with several hundred MIPs in processing speed are becoming common. Modern systems are incorporating dedicated processors for such interface concerns as speech recognition, speech synthesis, image compression/decompression and graphic screen management. With the growth of the multi-media market, the cost of imaging devices have fallen and will continue to fall. We already see computers equipped with cameras and realtime frame acquisition as their standard configurations. True digital cameras are making their presence felt in the marketplace. The issue at hand is not availability computational power and imaging capability as it is the development of clearly defined computer vision applications for the user interface. Such applications must be 'bounded' in computational needs to permit realtime interactive implementation and be robust. We believe that the way to achieve this is to engage in strident interdisciplinary research. Rather than waiting for some Percivallian researcher to achieve the Holy Grail of autonomous general purpose vision, we have to study the interface methodologies we wish to implement and exploit that information to devise domain-specific algorithms. This will allow us to build simpler vision systems by constraining the problem with the details of each particular interface approach. We shall discuss our work on hand gesture interfaces as an example of such an approach.

4 Hand Gesture Interfaces

We approach gesture interpretation by asking two basic questions – what ought to be done, and what is reasonable for machine vision to do? The chief motivation for using gesture is that humans have the facility and intuition for gestural communication. It will, therefore, not do to pay lip service to such intuition and then develop systems in a completely ad hoc fashion attending only to hand tracking technology. To answer our first question, we explore gesture interpretation and usage in humans. The second question addresses the technology of visual gesture interpretation. Machine vision provides the promise of interpretation of the motion and configurations of the unencumbered hand. A survey of the literature on general gesture usage and of the American Sign Language [30] will reveal, however, that a solution of the general gesture interpretation problem is beyond our capabilities for the foreseeable future. Humans bring a broad range of contextual information, general knowledge, cultural background, linguistic capabilities, and general cognition to bear on the formation and interpretation of gestures. We extract a reduced taxonomy of gestures that makes sense from the standpoint of human usage, and that is amenable to machine vision. We present an *optical flow*-based gesture interpretation approach.

4.1 Gesture Input Technology

It has been suggested that 3D applications such as 3D CAD require such 3D input modalities as gesture [31]. The 'virtual reality' community has designated gesture (e.g. pointing and picking things up) as an input modality of choice [32, 33, 34, 35, 36, 37, 38, 39]. What exactly is gesture-based computer input, however, is unclear in the literature. A 1987 round table discussion of gesture focused on tracking and interpreting the motions of a stylus on a tablet [40]. Researchers at MIT labeled 3D tracking of a stylus with respect to a reference frame a gesture input system [41]. In another gesture input project, a user wearing a wrist-mounted Polhemus cube manipulated figures of ships on a screen by pointing and speaking [42]. At Carnegie Mellon University, a gesture input project involved having a user hold a detector which was scanned by three orthogonal sweeping planes of light. The intersection of the three-planes specified a location to which a robot end-effector should move [43].

In lieu of the more precise discussion to follow, we shall use the term *gesture* to designate the configuration and motion of the human hand. The gesture may be performed in three-dimensional space or on a two-dimensional table. Baudel and Beaudouin-Lafon [18], for example, discuss the use of gestures for managing a document on a projection screen (e.g. turning pages, marking areas etc.) in which the user wears a glove device and gestures in three-dimensions. Pausch and Williams [44] describe a system which uses three-dimensional gestures of hand configurations to drive an articulator-based speech synthesizer. Segen [20] and Wellner [45], on the other hand, describe systems in which the user performs gestures in two-dimensions on a table top.

4.1.1 Gesture Transducing Gloves

The work done in tracking finger and hand motion have involved the use of special glove devices [46, 47]. We review three representative technologies. The *Dexterous Hand Master* by Exos Inc. is an elaborate hand exoskeleton which uses Hall effect sensors to measure the bends in the joints of each finger. The *DataGlove* [48] (which was produced by VPL Research, Inc.) uses specially fabricated optical fibers to determine the bend of each finger joint. The fibers are made so that they leak when they are bent. The amount of leakage determines the bend of each knuckle. Fels and Hinton, for example, have developed a neural network approach to interpret the DataGlove output [49]. The *Power Glove* (by Mattel, Inc.) measures the bend in the fingers with flat plastic strain gauges. The Dexterous Hand Master is the most accurate in the measurement of finger displacement and the Power Glove is the least. The Dexterous Hand Master and the DataGlove do not have intrinsic mechanisms for tracking the gross motion of the whole hand. Applications using these devices usually use the electromagnetic devices for hand tracking. The Polhemus sensor, for example, makes use of low frequency magnetic fields to determine the position of a cube (usually strapped onto the wrist) to 6 degrees of freedom. The Power Glove makes use of two ultrasonic transmitters mounted on the glove and three receivers mounted on the computer screen to determine the location of the hand to within a quarter of an inch.

These systems require the user to wear an encumbering device on the hand. This essentially 'dedicates' the hand to the interface rather than the interface to the user. A user of a glove-type device cannot, for example, pick up a pen and take notes, type on a keyboard, play a musical instrument or pick up a cup of coffee conveniently. The stiffness of gesture transducing gloves and the need to be tethered to the machine render cumbersome an otherwise highly versatile mode of interaction. Electromagnetic tracking technology will not work in metal-laden workspaces (such as an aircraft cockpit). The sonic sensors of the Power Glove require the front of the glove to be pointed at the sensors.

4.1.2 Vision-Based Approaches

Recently, several efforts in vision-based hand gesture interpretation have been reported. Wirtz and Maggioni describe a system which tracks a hand wearing a silk glove with fiduciary markings on it using a downward looking camera [19]. Their system is capable of determining hand orientation from the perspective distortion of the marking. It determines hand position more accurately in the plane of the desktop on which the hand moves than in dimension of the height of the hand above the desktop. This is because they use only image size cues to determine range from the camera. They demonstrated their system for navigating in a virtual three-dimensional space. Segen describe a system for recognizing gesture symbols performed over a light table [20]. The system recognizes static hand poses from the silhouette on the light table using a backpropagation neural net. Fukumoto et al studied realtime three-dimensional pointing with a typical pointing gesture [17]. Their system operates with two cameras: one looking directly down on the hand and another from the side. Assuming a pointing hand pose, the extremal point in both images away from the subject is the finger tip. They posited that pointing rays project from a *virtual projection origin* (VPO) through the finger tip and onto the projection screen which served as the target area. They were able to calibrate their system for different users by computing their VPO's from calibration trials where the users pointed to known locations. The side camera also detected the abduction angle between the thumb and the index finger to constitute a mouse-button-like thumb-switch. Cipolla et al describe an approach for qualitative three-dimensional hand gesture interpretation using motion parallax [50]. Making the assumption that small neighborhoods of individual hand points are rigid, their algorithm computes the relative distances among the points from the camera as a qualitative partial order (i.e. points are can be ordered by closeness to the camera without determining the exact distances). Marking the user's hand (or glove) with four markers, they are thus able to determine qualitative rotation of the hand between frames without computing the exact rotations. The user utilizes visual feedback and incrementally positions his/her hand until the desired orientation in the display is achieved.

4.2 Exploring Gesture

Our goal is to apply vision techniques to interpret a real-time video stream of a user's gestures with an unadorned hand. Before one can begin devising computer vision approaches and designing vision algorithms to analyze hand gestures, one needs to know what the salient aspects of gestures are. Do we need to extract the angle of each joint of each digit? If yes, how accurately must this be done? Do we need to know the angular velocity and acceleration of the joints? Must we model every finger of the hand as articulated three segment entities? How important is the shape of the palm? How much information is contained in the location of the hand with respect to the communicator's torso? Are there different modes of hand gesture communication, and if so, which modes should we address to facilitate human-computer interaction? Are there particular modes which are especially suited for particular interface needs (such as virtual reality input)? What is the relationship between speech and gesture when they are used in tandem? These and many more questions need to be answered before one can determine the computer vision primitives and processes necessary to hand gesture interpretation.

4.2.1 Communicative and Manipulative Gestures

We make a distinction between gestures intended for communication and those intended for manipulating objects. A orchestral conductor's hand motions are intended to communicate temporal, affective and interpretive information to the orchestra. A pianist's hand movements are meant to perturb the ivories. While it may be possible to observe the pianist's hands, the hand and finger movements are not meant to communicate with anyone. This distinction is important when considering vision-based gesture interpretation. Communicative gestures are meant for visual interpretation. No visually obscured part of the hand will carry information necessary to understanding the gesture. Manipulative gestures, on the other hand, are not subject to such constraints. There is ultimately no guarantee that such gestures are visually interpretable. This is not to say that manipulative gestures are unimportant. If, however, one intends for users to manipulate objects in a virtual environment, it may be more appropriate to use implements like glove devices to transduce the hand motion. One may in fact even think of the computer keyboard as a manipulative gesture device which detects downward finger thrusts. Some manipulative gestures are amenable to visual interpretation, and where this is so, vision-based systems may be built. This important communicative-manipulative distinction is sorely lacking in the discussion of gesture in human-computer interaction and computer vision literature. In the rest of this paper, we shall concentrate on communicative gestures.

4.2.2 Communicative Gestures

Turning to the work of semioticists who have investigated the subject, we derive a sense of how gesture is used and perceived among human beings. The purpose of this exercise is not to build an interface system that is able to interpret all the nuances of human gesture. The limitations of current technology will invariably limit our ability to do so. Ultimately, the human user will have to meet technology 'halfway'. Machine vision technology holds immediate promise for interpreting a sufficient number of gestures to make a hand gesture

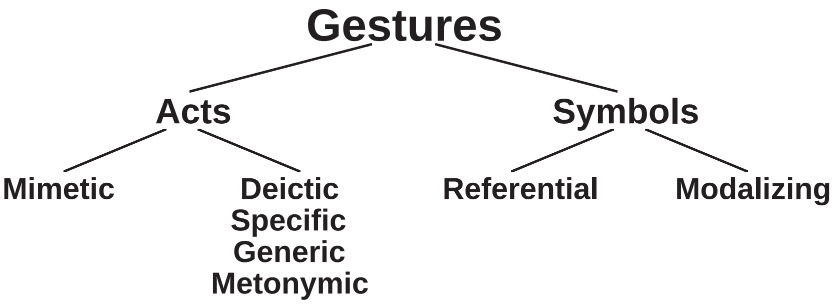

Figure 1: A taxonomy of gestures

interface practicable. We begin our inquiry, therefore, by sifting the available information on its use by humans to discover reasonable reductions. For successful application of machine vision, we need to constrain the space of admissible hand configurations, to know the spatio-temporal characteristics of gesture (and other cues that may be available) for segmenting gesture streams, and to constrain the types of gesture movements.

While the application of hand gestures to human-computer interaction is a relative new frontier of research, the study of how humans use and interpret gestures has a long and rich history. The study of its use in oratory dates back to the seventeenth century [51]. More recently, semioticists from the fields of anthropology, neurophysiology, neuropsychology and psycholinguistics have long been interested in the study of gesture [52, 53]. The most general definition from the 1977 *Lexis* dictionary says that gestures are "movements of body parts, particularly the arms, the hands or the head conveying, or not conveying, meaning." Nespoulous and Lecours [54] divide gestures into *centrifugal* gestures ("having obvious communicational intent") directed toward the interlocutor, and *centripetal* gestures which may be observed and interpreted as mood and desire indicators although they are not directed at anyone. Although it has been suggested that in order to realize "conversational computer interfaces," machines will have to pick up on unintended gestures (a computer may interpret fidgeting and other body language signs as indicating boredom in the user and alter the content of information presented) [42, 55], we restrict our investigation only to deliberately expressive movements. We are interested in having a computer determine hand gestures performed purposefully as instructions to a machine.

4.2.3 Symbols and Acts

A useful way to view gestures is in terms of the distanciation (or abstraction) of the movements from the intended interpretation [54]. *Symbol* gestures are a kind of motion shorthand, serving in a linguistic role. Sign languages are largely symbolic. In *act* gestures, the movements performed relate directly to the intended interpretation. Figure 1 outlines the taxonomy of gesture discussed in this section.

Symbol gestures are *arbitrary* in nature. The *transparency-opacity* dichotomy applied commonly by semioticists is useful in our consideration of such gestures. On initial inspection, there appears to be a class of symbolic transparent gestures that are immediately evident to all observers without the predetermination of some prior convention. Anthropological semioticists, however, have concluded that these are not dominant or even practical as a class. Nespoulous and Lecours observed that "if some gestures do have more cross-cultural extension than most linguistic signs, it remains that cultural differences are far from negligible. If some gestures can indeed be thought to be somewhat transparent at some stage of their social history, their diachronic evolution turned them into more and more opaque segments." In their study of American Sign Language, for example, Klima and Bellugi concluded that "few signs are so clearly transparent in their iconicity [having 'pictorial' origins] that a nonsigner can guess their meaning without some additional cues" [56, 57].

Symbol gestures may be classified as *referential* and *modalizing*. The former operates independently to designate objects or concepts. Rubbing the index finger and the thumb in a circular fashion, for example, is referential to money (the referent). Modalizing gestures serve in conjunction with some other means of communication (e.g. speech) to indicate the opinion of the communicator. For example, at a party, one might say to another, "Have you seen her husband?" (holding her hands apart to indicate that he is overweight). The resulting chuckle would not be understandable if one listened only to an audio transcript of the exchange. The gesture is said to modalize the statement. Another example is the common continuation symbol (repetitive circular motion of the hand the index finger pointing up) which can mean "keep giving me examples," "keep scrolling the page," "move the object closer to the target," etc. depending on context and accompanying communication by other modes.

Act gestures may be divided into two classes: *mimetic* and *deictic*. Mimetic gestures are characterized by their *iconicity*. They are usually performed as pantomimes of the referent. For example, a smoker going through the motion of 'lighting up' with a cigarette in his mouth indicates that he needs a light. Such gestures are usually generated on-the-fly without predetermined convention. The more novel the pantomime, the more exaggerated the motion would be to convey its intent.

Deictic (or pointing) gestures are of great interest for computer input. Such gestures may be classified into *specific*, *generic* and *metonymic* deictic gestures. The classification is determined solely by context. *Specific* gestures are issued when the subject selects a particular object or location. Running a program by pointing at an icon, for example, is a specific deictic gesture. *Generic* deictic gestures elicit the identity of a class of object by picking one of its members. *Metonymic* deictic gestures are issued when a subject points at an object to signify some entity related to it. When a subject points to a picture of skyscrapers to signify New York city, the subject is performing a metonymic deictic gesture. Pointing to an object to indicate its function (e.g. pointing to a crane to tell the robot to lift an object) is also metonymic. Since all three deictic gesture types are performed in the same way mechanically (the difference is in higher level interpretation), we shall deal only with the recognition and interpretation of specific deictic gestures.

Kendon [58] makes a distinction between *autonomous gestures* (gestures performed in the absence of speech) and *gesticulation* (gestures performed concurrently with phonological utterances). He notes that in conversational contexts, "gestures of interactional management are done mainly with the head and face, rarely with the hands." Such gestures are used by the listener to signal that what is said is understood. When performed by the speaker, the head is used in "gestures of assent and negation," and pointing. He notes that gesticulations "appears largely to be confined to discourse segmentation function." Hand gestures are used often in the narration of events to "depict actions and to portray the spatial structuring of situations." Kendon describes an experiment in which subjects who viewed a motion picture were required to describe what they saw. Most of the gestures recorded were of the head-and-eye variety. Hand gestures dominated when subjects were describing events where space and time are important. The predominant role of hand gesture used in relation to speech is as a spatial/temporal qualifier (to specify location, orientation, spatial relation and spatial contour) and as a volumetric qualifier (to specify size).

4.2.4 Gesture Segmentation

We turn our attention to the separation of gesture movements from each other and from hand movements with no communicative intent. With the exception of 'beats'[53], three distinct motion phases typically constitute a gesture [59, 58, 53, 60]: preparation, stroke and retraction. The stroke is the salient gestural movement. The preparation phase orients the hand for the stroke and the retraction phase returns the hand to rest or orients the hand for the next gestural stroke. Kendon [58] calls this sequence a "gesture phrase". The stroke is differentiable from the other two phases in velocity and acceleration.

Kendon [59], whose interest is anthropological, reports an experiment in which subjects (25) were allowed to view film segments as many times as desired, there was good consensus as to what constituted gestures. He observed that "Deliberately expressive movement was movement that had a sharp boundary of onset and that was seen as an *excursion*, rather than as resulting in any sustained change of position."

Key observations relevant to gesture segmentation are: limb movements beginning with sharp movements away from the body and returning to the original positions are deliberately expressive; head movements (up-down or rotations) which did not remain in their new positions are deliberately expressive; whole-body movements that return to their original positions after the gesture are deliberately expressive; movements involving object manipulation are never seen as deliberately expressive; and movements in which the conversant touches oneself or one's clothing are not deliberately expressive. Out of this body of work, we extract the following rules:

- Movements which comprise a slow initial phase from a rest position, proceed with a phase at a rate of speed exceeding some threshold (the stroke), and returns to the resting position are gesture laden.

- The configuration of the hand during the stroke is in the form of some recognized symbol (defined by convention in a library).

- Motions from one resting position and resulting in the and in another resting position are not gestures. These will be interpreted as 'practical' motions.

- Hand movements outside some work volume will not be considered as pertinent gestures. For example, a hand movement to adjust eye glasses moves too close to the body of the subject and away from the specified work volume.

- The user will be required to hold a static hand gestures for some finite period for them to be recognized.

- Repetitive movements in the workspace will be deemed gestures to be interpreted.

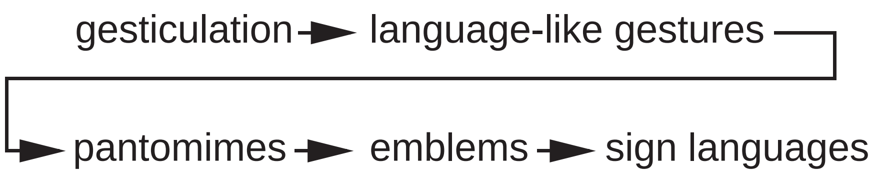

Figure 2: A continuum of gestures (reproduced from *Hand and Mind*, David McNeill, 1992

4.2.5 Gesture Assumptions and Constraints

While most natural gesture usage involves two hands, our current research investigates gestures performed by one hand. This simplification permits later extension to two-handed gestures.

Given the opacity of gestures in human usage, we consider only those gestures that have been included into a library by convention. Such a library must be chosen to reflect common social convention. Other than noting its existence, we do not address pantomime gestures generated on-the-fly. Any mimetic gesture deemed useful to the system can be codified and included in the library.

Since we consider only hand gestures which are performed purposefully by the user, gestures must be deliberately aimed at the machine's vision system. For example, the common 'OK' sign will be performed with the open palm facing the camera. This eliminates most of the problems of three-dimensional occlusion from the recognition process.

The conceptual space occupied by gestures in human-computer interaction lies somewhere between the fluid free-form gestures performed by humans in discourse and the rigidly specified gestures employed in full-fledged sign languages [53]. Sign languages have to carry all the linguistic information necessary for symbolic communication. For human-computer interaction, gestures make sense within contexts. They serve the same purpose as *emblems* (see figure 2). Humans use such gestures in communication as a kind of a shorthand expression to augment and modulate other communication or when the context is clear. The 'thumbs-up' gesture to signify 'all is well' is an example of such a gesture.

The dominance of spatio/temporal description in human hand gesture usage argues that the same emphasis be applied to hand gesture interfaces for computers. The application of gestures for purposes such as mode specification and as a replacement of symbolic data entry should be done sparingly and only when there are compelling reasons to do so (e.g. in interfaces for the speech impaired). We restrict modalizing gestures to the class of scale modifiers that determine speed, relative size and distance. In our 'continuation gesture', the user may instruct a robot to move further down by saying "Down, down, down ..." while issuing the circular hand motion. In this case the speed of the hand movement will determine the speed at which the robot should move.

We entertain four classes of act gestures:

Locative: pointing to a location, in a direction or at an object;

Orientational: placement of objects or viewpoints by specifying rotations by hand;

Spatial pantomime: tracking the gross motion of the hand to determine some shape, path or similar spatial contour;

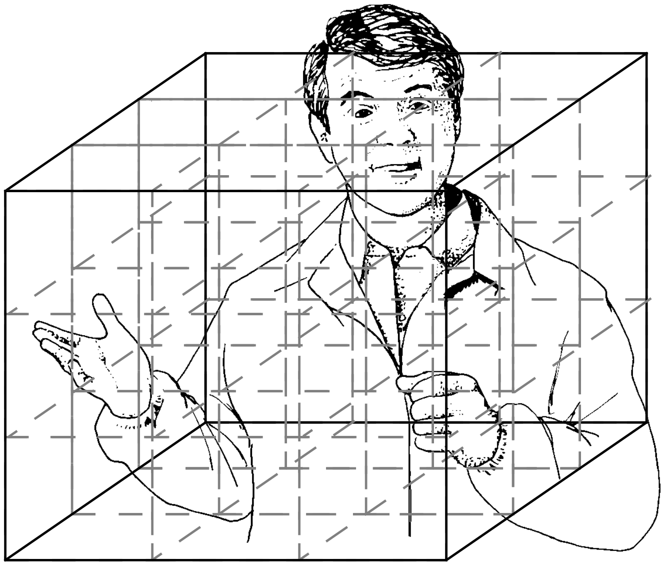

Figure 3: The discrete segmentation of the gesture space

Relative spatial: spatial relationships such as nearer, further, further right, further left, etc.

Furthermore, we observe that few signs in American Sign Language involve movement of both the fingers and hand together [30, 57]. Except for some obvious finger movements (e.g. wriggling all fingers with down-turned hands moving away from oneself to denote 'river'), human observers find precise finger movements on a moving hand difficult to detect. This permits us to make two simplifying rules:

1. When the hand is in gross motion, we have a 'hand gesture' in which movement of individual fingers is unimportant.

2. Gestures in which fingers move with respect to each other will be performed with the hand relatively still.

Changes in hand shape in the midst of gestures of gross hand motion are more often perfunctionary (to bring the hand shape to the final position of the gesture such as when a hand advances from a closed configuration to a pointing configuration as the hand moves forward) than functional. Human perception is incapable of extracting the details of hand-shape change with rapid gross hand movement. This means that the majority of hand gestures (as opposed to finger or hand shape gestures with the hand held stationary) may be processed by extracting the hand configurations at the start and end of the stroke and analyzing the gross motion of the stroke while disregarding the shape changes during the stroke.

Our gesture model defines a gesture space in front of the subject. For gestures of gross hand motion, this space is divided into 3 × 3 × 3 discrete subspaces (see figure 3). The strokes of gestures may originate and terminate in any of these subspaces. In our model, communicative gesture interpretation entails:

1. Segmenting the gesture stream to identify the stroke

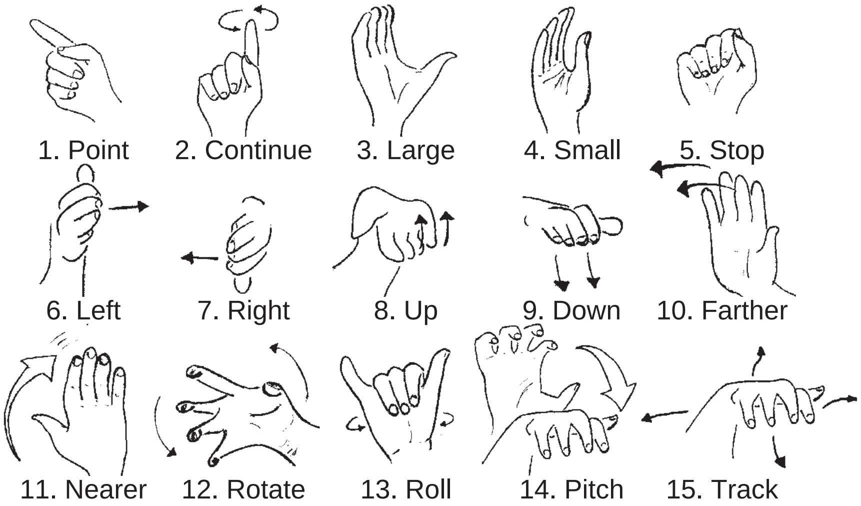

Figure 4: Gesture Vocabulary

2. Determining the subspace of gesture origin and termination (or the subspace of the extrema of repetitive gestures).

3. Determining the image dynamics of the hand during the stroke.

4. Recognizing the hand configurations at the ends of the stroke

4.2.6 Gesture Vocabulary

Figure 4 is a set of fifteen features formulated from the rules outlined. This vocabulary emphasizes gestures which describe spatial relationships, structure and motion. Hence, we have not included any simple referential symbols (these may be added for particular applications). We envision this vocabulary to be applied in applications which require 3D input such as virtual reality and telerobotics.

The *point* symbol is the primary *locative* gesture. The *large*, *small*, *left*, *right*, *up*, *down*, *farther* and *nearer* symbols make up the *relative spatial* gestures. All but the first two of these may be performed in repetition. *Large* and *small* can denote extent of space (larger, wider or more) and time. The *orientational* gestures in our vocabulary are *rotate*, *roll* and *pitch*. The degree and direction of hand rotation constitutes the actuating signal. The *track* gesture in which the motion of the hand is a *spatial pantomime* specifying desired motions.

5 Extracting Image Flow from Hand Gestures

Of the three phases of a gesture, the stroke is the information laden segment. Psychologists and semioticists have known that there is a dynamic difference between the stroke and the preparation and retraction gesture phases, but this has never been quantified. This suggests that studying this dynamic is essential to the segmentation of hand gestures from video sequences. The ability to extract the stoke will provide information about the starting and ending locations of the stroke and the path taken by the gesture. Furthermore, the

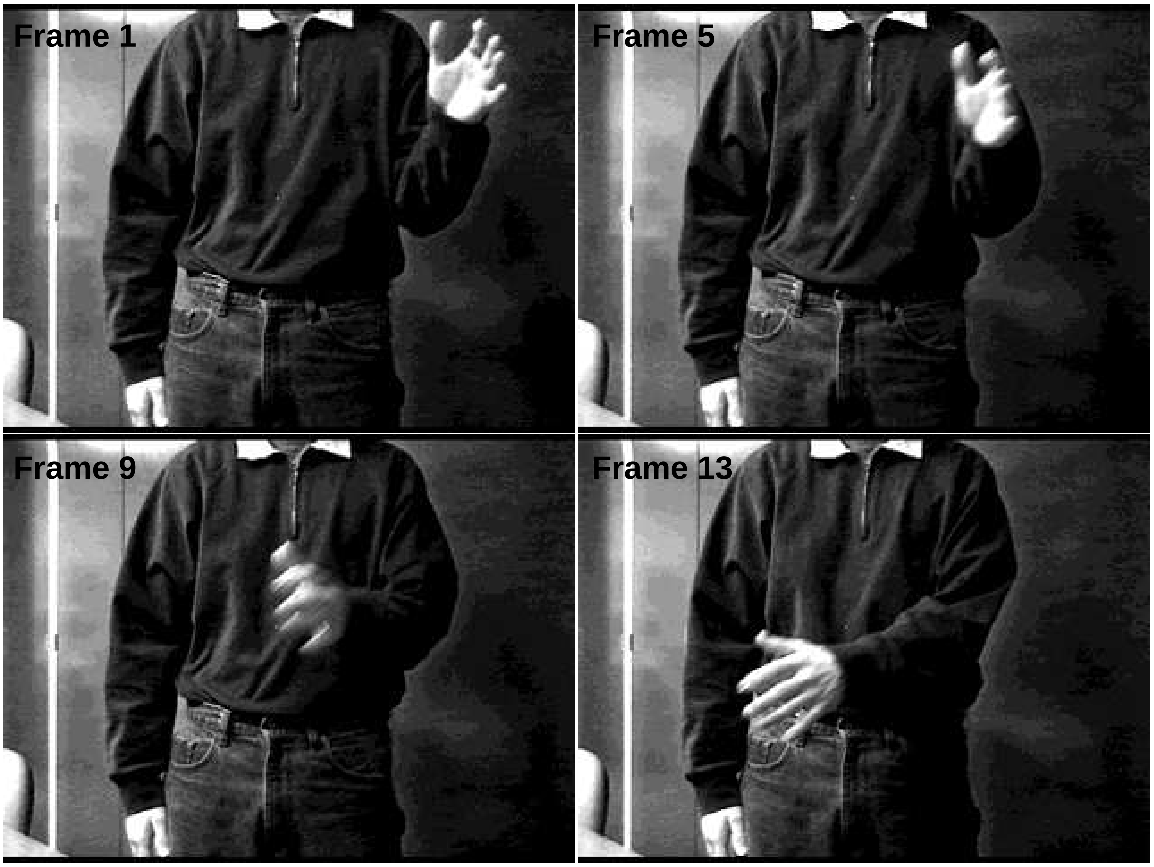

Figure 5: Frames 1, 5, 9, and 13 of the 15 frame (0.5 second) gesture sequence

velocity of the gesture often determines the intensity of the information conveyed [53]. Hence, understanding the image plane dynamics of hand gestures in a video sequence will afford us significant progress in hand gesture interpretation.

Our approach is to compute a set of spatio-temporal vectors which track the boundary of the moving hand. The computation is in four stages.

1. We process the images to find the location of moving edges;

2. We compute an initial flow field describing the velocities at moving edge points; and,

3. We refine the flow-field by applying a variance constraint to smooth it.

4. We cluster the vectors to find the cluster representing the moving hand.

Figure 5 shows frames 1, 5, 9 and 13 of a typical video sequence processed by our system. This particular video sequence which contains 15 frame covering a 0.5 exhibits a fair amount of motion blur. This is typical of our data because of the speed of most hand gestures. We capture the data at 30 frames per second.

5.1 Detecting Moving Edges

Rather than extracting edges in successive video images and then computing the correspondences to find moving edges, our algorithm enhances moving edges during edge extraction. This is important since no assumption can be made about the background (which includes clothing texture of the subject). We make an assumption that the gesturing hand is the fastest moving object in a generally static image. Applying a variant of the moving edge detector due to Haynes and Jain [61], we obtained strong edges for the hand when it moves in the image. Our operator computes both the temporal and spatial gradients in consecutive images and combines them with a fuzzy operator. We then extracted and thinned the edges using *non-maximal suppression.*

A video scene containing gestures can be modeled as a three-dimensional signal $I(x, y, t)$ in the image-time domain. An estimate of the partial derivative of this signal with respect to time is given by:

$$\frac{\partial I(x,y,t)}{\partial t} = D(x,y,t) = |I(x,y,t) - I(x,y,t+1)| \tag{1}$$

The partial derivative of the signal with respect to varying x and y may be computed by applying the Sobel operator:

$$\begin{aligned} \frac{\partial I(x,y)}{\partial x} &= S_x \otimes I(x,y) \\ \frac{\partial I(x,y)}{\partial y} &= S_y \otimes I(x,y) \end{aligned} \tag{2}$$

where S_x and S_y are the x and y directional Sobel operators respectively and $\otimes$ denotes convolution.

The Sobel operators yield gradient images in which the image value is a measure of edge intensity. We combine such a image with a corresponding time-derivative image by performing a pixelwise multiplication of the images. This yields a *fuzzy-AND* operation between the images. The *fuzzy-AND* image is a multi-valued image in which a pixel intensity is a measure of moving edge intensity. We obtain four such images from each pair of images in a video sequence.

$$\begin{aligned} M_x(x,y,t) &= \frac{\partial I}{\partial x}(x,y,t) \cdot \frac{\partial I}{\partial t}(x,y,t) \\ &= S_x \otimes I(x,y,t) \cdot D(x,y,t) \\ M_y(x,y,t) &= \frac{\partial I}{\partial y}(x,y,t) \cdot \frac{\partial I}{\partial t}(x,y,t) \\ &= S_y \otimes I(x,y,t) \cdot D(x,y,t) \\ M'_x(x,y,t+\delta t) &= \frac{\partial I}{\partial x}(x,y,t) \cdot \frac{\partial I}{\partial t}(x,y,t) \\ &= S_x \otimes I(x,y,t+\delta t) \cdot D(x,y,t) \\ M'_y(x,y,t+\delta t) &= \frac{\partial I}{\partial y}(x,y,t) \cdot \frac{\partial I}{\partial t}(x,y,t) \\ &= S_y \otimes I(x,y,t+\delta t) \cdot D(x,y,t) \end{aligned} \tag{3}$$

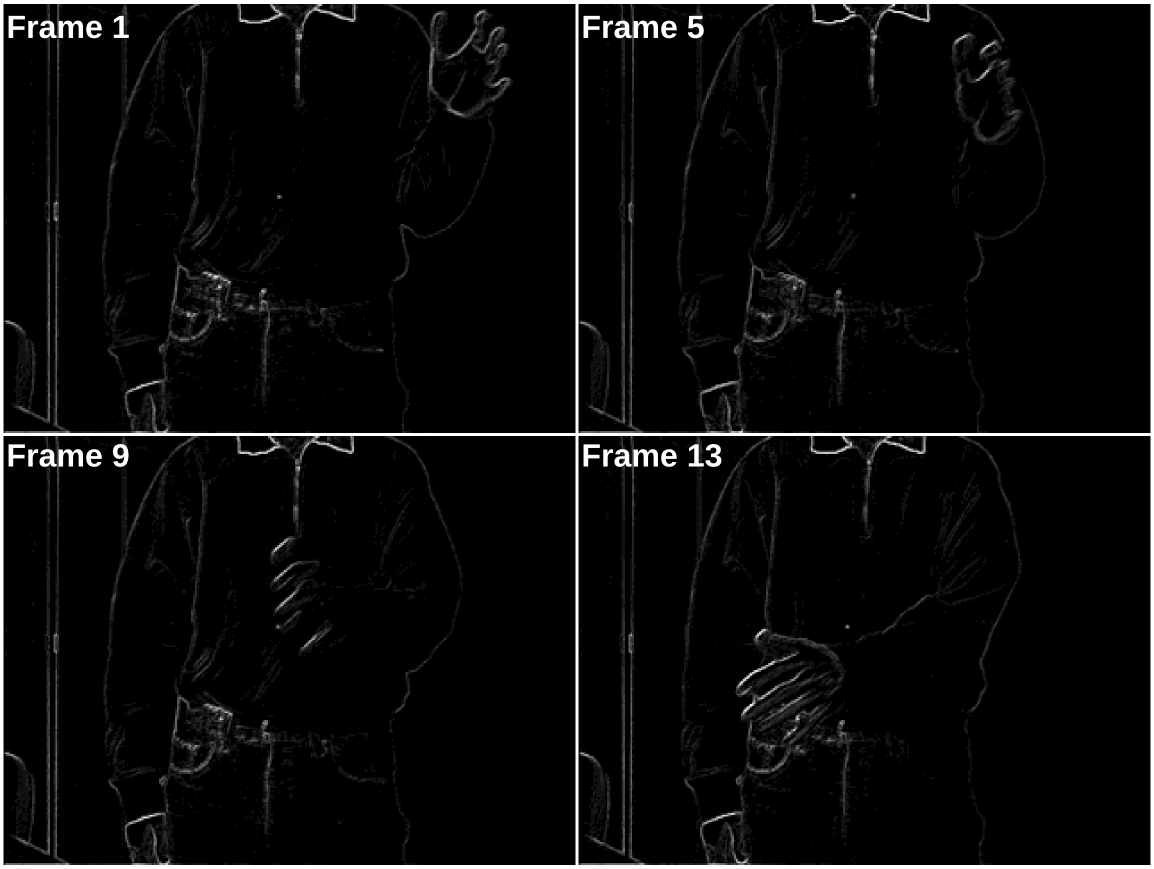

Figure 6: Sobel gradient magnitude of frames 1, 5, 9, and 13 of the 15 frame (0.5 second) gesture sequence

Since the Sobel operator produces 'gradient ridges' at edges, one may locate the edge by tracing the top of these ridges. We apply *non-maximal suppression* to achieve this. The idea of non-maximal suppression is to eliminate edge points which have neighboring pixels with greater gradient values in the edge direction. In practice, however, edge directions computed from the Sobel gradients are not very accurate. Hence, we used only the two diagonal directions. We choose the diagonal edge directions because performing the suppression in the horizontal and vertical directions will result in missing edge points for diagonal edges. Diagonal suppression, on the other hand, functions properly for horizontal and vertical edges, so the appropriate diagonal suppression was used in all cases. The appropriate diagonal was determined by observing the signs of the M_x and M_y edge directions as shown in equation 6 (the diagonal was picked arbitrarily in zero cases).

The final binary moving edge image was then computed. Note that this is actually performed twice, once to calculate $E(x, y, t)$, the moving edges with respect to the first image, and again, to calculate $E'(x, y, t)$, the moving edges with respect to the second image. The threshold of the moving edge magnitude is denoted T as follows:

$$E(x,y,t) = \begin{cases} 1 \text{ if } n_0 > T \text{ and } n_0 = \max(n_0, n_1, n_2) \\ 0 \text{ otherwise} \end{cases} \tag{4}$$

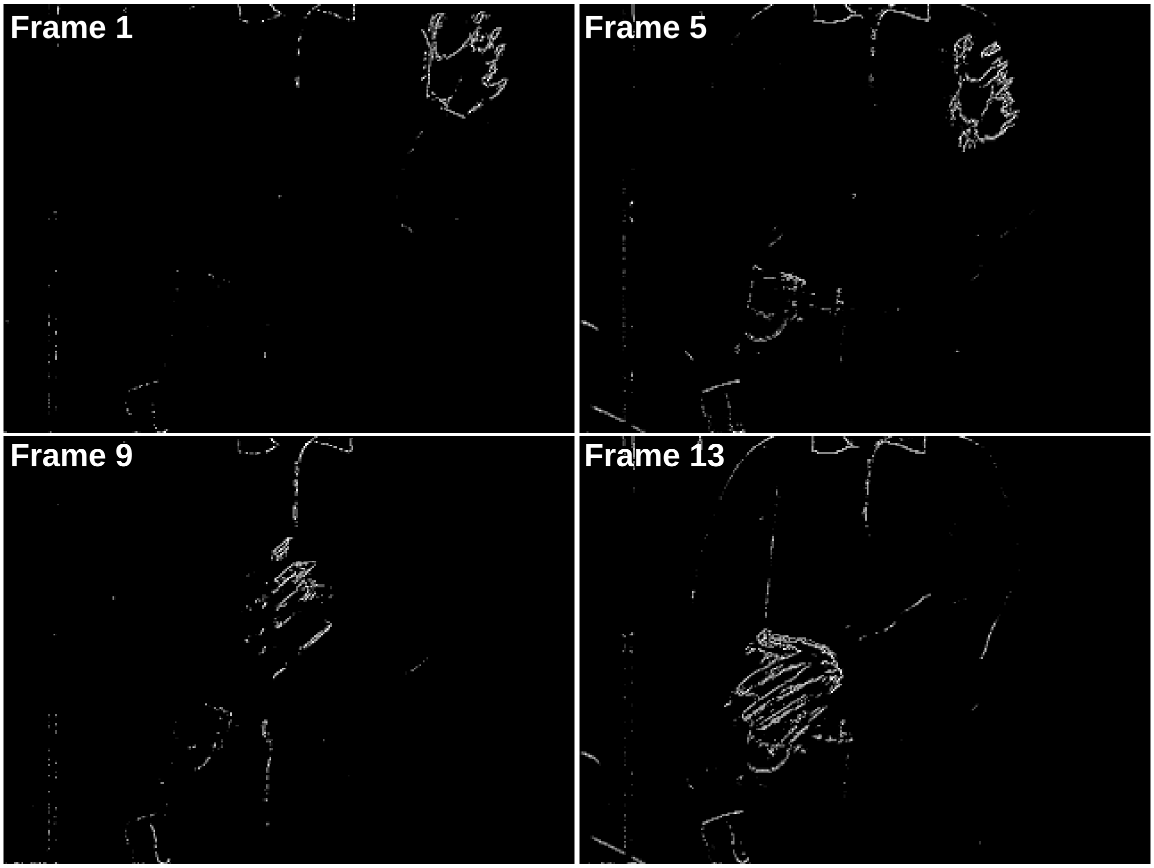

Figure 7: Moving edges detected in frames 1, 5, 9, and 13 of the 15 frame (0.5 second) gesture sequence

To compute, n_0, n_1, and n_2, we define an M vector:

$$\vec{M}(x,y,t) = (M_x(x,y,t), M_y(x,y,t)) \tag{5}$$

The relation between M_x and M_y determines the diagonal vectors applied in equation 4 as follows:

$$\begin{array}{rcl} \text{if}(M_x > M_y) \rightarrow & & \\ n_0 & = & ||\vec{M}(x,y,t)|| \\ n_1 & = & ||\vec{M}(x+1,y+1,t)|| \\ n_2 & = & ||\vec{M}(x-1,y-1,t)|| \\ \text{else} \rightarrow & & \\ n_0 & = & ||\vec{M}(x,y,t)|| \\ n_1 & = & ||\vec{M}(x+1,y-1,t)|| \\ n_2 & = & ||\vec{M}(x-1,y+1,t)|| \end{array} \tag{6}$$

Figure 6 shows the gradient magnitude image computed using the Sobel operator at these video frames. Figure 7 display the moving edge points computed by our system after non-maximal suppression. Notice that a large number of non-moving edge points which exhibit high gradient magnitude (in figure 6) are eliminated in the moving edge image.

5.2 Initial Flow Field Computation

The goal of this step is to find the velocity of the moving edges. The edge velocities are computed by first choosing dominant edge points where the velocities are computed. This reduces the computational load in the interest of speed. Next, the edge point correspondences from different edge images is computed by a correlation process.

5.2.1 Dominant Edge Point Computation

The edge points where velocities were calculated were chosen as follows. At most one edge point was chosen in each 10 × 10 pixel window of our images (our system works with 640 × 480 images). Within each window only a moving edge point with two adjacent edge points was chosen. If no edge points satisfying that constraint exists in a given window, no velocity was calculated in that window. This constraint increased the chance that the velocities we are calculating are important edge points in the image. This computation is formalized as follows:

$$\mathbf{E}''(t) = \left\{ \begin{array}{c} \forall W \in \mathbf{W}, (x_e, y_e) \in W, E(x_e, y_e, t) = 1, \\ \text{and } \sum_{i=-1}^{1} \sum_{j=-1}^{1} E(x_e + i, y_e + j, t) \geq 3 \end{array} \right\} \tag{7}$$

Where W is a window in the image, $\mathbf{W}$ is the set of all image windows, and (x_e, y_e) are edge coordinates.

5.2.2 Edge Point Correlation

In order to compute the velocities of the edge points, we have to determine the correspondence from the dominant spatio-temporal edge points of one image frame to pixels in the next image frame. There are a number of difficulties in determining these velocities. Chief among these is the aperture problem which renders the computation of velocities at any orientation other than those perpendicular to the edge unreliable. Global information is needed to determine the true velocity. Hildreth [62] addressed this problem by introducing a constraint of smoothness of the velocity field. Our approach is to impose a similar smoothness constraint within the vector field. One may conceive the computation of such a smooth field, given a set of vector origins as the selection of an appropriate set of vector endpoints from sets of candidate endpoints. In order to do this, a set of initial vectors must be computed as well as a set of candidate vector endpoints. The dominant edge points described in section 5.2.1 serve as the vector origins.

Let the set of candidate velocity vectors from edge point (x_e, y_e) be $\mathbf{V}(x_e, y_e)$. We obtain this set of vectors by determining the set of candidate endpoints that correlate with (x_e, y_e) by applying *Absolute Difference Correlation* (ADC) [63]. ADC computes point correlations by minimizing the absolute difference between the image region surrounding the vector origin with a set of sliding windows in the target image (see equation 8).

$$\mathrm{ADC}(x_e, y_e, V_x, V_y) = \sum_{i=x_e-N}^{x_e+N} \sum_{j=y_e-N}^{y_e+N} |I(i,j,t) - i(i+V_x, j+V_y, t+1| \ \forall (x_e, y_e) \in \mathbf{E}''(t) \quad (8)$$

We further observe that the candidate matches must be on or close to the next spatio-temporal edge. Hence, we further constrain the sliding windows to move within a single pixel of the subsequent edge image. The general ADC form (equation 8) is hence modified to yield:

$$\mathbf{V}(x_e, y_e) = \{(V_x, V_y)\} \text{ s.t.} \left\{ \begin{array}{l} |V_x| \le D, |V_y| \le D, \\ \sum_{i=-1}^{1} \sum_{j=-1}^{1} E'(x_e + V_x + i, y_e + V_y + j, t) \ge 1, \text{and} \\ \mathrm{ADC}(x_e, y_e, V_x, V_y) \le T \end{array} \right\} \quad (9)$$

where T is a threshold to remove low correlation points from the candidate set, N is the ADC neighborhood (we used $N = 2$), D is the maximum interframe edge displacement ($D = 30$).

It is important to notice that the ADC is computed on the original intensity images. The spatio-temporal edge images are used simply to constrain the search space for candidate matches. Hence, our algorithm does not suffer from cumulative errors from previous computation stages.

We obtain our initial estimate of the vector field by selecting endpoints which yield the minimum absolute difference correlation. Figure 8 presents the initial velocity field computed by our algorithm on the same frames displayed in figure 5. While the vectors are describe the general direction of the hand movement, there is a fair number of erroneous vectors and noise.

5.3 Variance Constraint Application

Finally, we select the best set of vectors from the endpoint candidates. The best set of vectors are chosen to minimize the following expression which represents the normalized velocity change across the image's vector field. This enforces the spatial smoothness constraint on the velocity field. .

Equation 10 is the variance function which we minimize in our algorithm:

$$\min \iint_A \frac{\frac{d\vec{V}}{dS}}{||\vec{V}||} dS = \sum_{(x_e, y_e) \in \mathbf{E}''} \sum_{(x'_e, y'_e) \in \mathbf{E}''} \left\{ \begin{array}{ll} \frac{||\vec{\mathbf{V}} - \vec{\mathbf{V}}'||}{||\vec{\mathbf{V}} - \vec{\mathbf{V}}'|| \cdot ||\vec{\mathbf{V}}||} & \text{if } ||(x_e, y_e) - (x'_e, y'_e)|| \le N \\ 0 & \text{otherwise} \end{array} \right\} \quad (10)$$

where $\vec{\mathbf{V}} = \vec{V}(x_e, y_e)$, $\vec{\mathbf{V}}' = \vec{V}(x'_e, y'_e)$, and velocity pairs outside the N-neighborhood do not affect the sum.

We actually minimize the square of the expression on the right to avoid unnecessary square root computations. The goal, then, is to find the globally optimal $\vec{V}(x_e, y_e)$ at every chosen edge point from the candidates in $\mathbf{V}(x_e, y_e)$ which minimizes the global smoothness function. This general problem is NP-Complete.We use a greedy hill-climbing algorithm that works well for our data set. This approach minimizes the variance between points by switching velocity vectors at points and eliminating deviant velocity vectors.

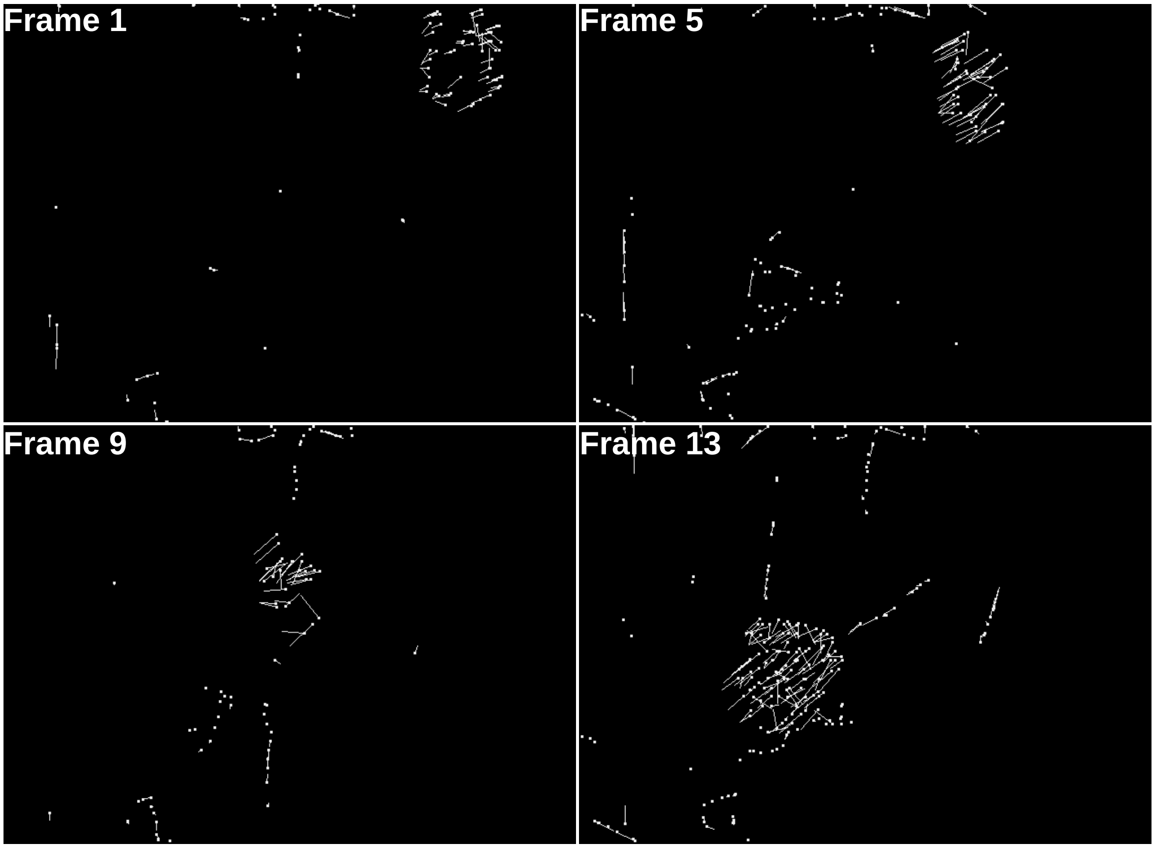

Figure 8: Initial velocity field computed in frames 1, 5, 9, and 13 of the 15 frame (0.5 second) gesture sequence

The algorithm is to starts with an estimate of the best velocity field, and progressively refines it. Refinements are continued until the refinement would make a difference of less than a threshold or the maximum iteration count is exceeded. There are three refinements that can be made in one iteration. The operations in each iteration are summarized below:

1. Switch vector $\vec{V}(x_e, y_e)$ with another vector $(V_x, V_y) \in \mathbf{V}(x_e, y_e)$ that causes the greatest decrease in expression 10. The switch is only performed if the decrease is greater than a small threshold T_s. Refinements (2) and (3) are permitted if the switch decrease is less than a threshold $T_{sd} > T_s$.

2. Remove a deviant point (x_e, y_e) from $\mathbf{E}''$. A point is deviant if removing it would decrease the sum by more than a threshold T_d. The most deviant point is always removed first.

3. Add non-deviant excluded point (x_e, y_e) back into $\mathbf{E}''$. A point is non-deviant if adding it would increase the sum by less than a threshold T_d. The most non-deviant excluded point is added first.

For efficiency, we precompute the possible variance contribution $C(x_e, y_e, V_x, V_y)$ for each $(V_x, V_y) \in \mathbf{V}(x_e, y_e)$ in the current set of $\vec{V}(x_e, y_e) \in \mathbf{E}''$ (Equation 11).

$$C(x_e, y_e, V_x, V_y) = \sum_{(x'_e, y'_e) \in \mathbf{E}''} \left\{ \begin{array}{ll} \frac{||\vec{\mathbf{V}} - \vec{\mathbf{V}}'||}{||\vec{\mathbf{V}} - \vec{\mathbf{V}}'|| \cdot ||\vec{\mathbf{V}}||} & \text{if } ||(x_e, y_e) - (x'_e, y'_e)|| \leq N \\ 0 & \text{otherwise} \end{array} \right\} \tag{11}$$

where $\vec{\mathbf{V}} = \vec{V}(x_e, y_e)$, and $\vec{\mathbf{V}}' = \vec{V}(x'_e, y'_e)$

To avoid repeated computations of this value, the contributions are first initialized to zero and incremented for each pair of $\vec{V}(x_e, y_e) \in \mathbf{E}''$ and $\vec{V}(x'_e, y'_e) \in \mathbf{E}''$. Later, when refinements are made, only affected contributions are updated.

Figure 9 shows the smoothed velocity field computed on the video sequence shown in figure 5. Observe how this field is significantly 'clean' when compared with the initial velocity vectors shown in figure 8. In frame 13 (lower righthand image) of figure 9, the hand is coming to rest, and the cluster of short vectors labeled 'hand' describes the correct hand vectors. The set of longer vectors nearer the center of the frame actually originates at the location of the sleeve-hand boundary. Owing to the strength of the sleeve-hand edge points, and the regularity of the edge, a set of false matches were detected, giving rise to a larger-than-expected vector field in the direction of the hand movement.

5.4 Vector Clustering and Path Smoothing

Up to this point, our system deals only with individual vectors. We finally cluster the vectors to determine a congruent set of clusters which cohere spatially, temporally and in direction of motion.

Vectors within a frame which are close together and point in the same general direction are deemed to belong to the same cluster. The algorithm determines the appropriate clusters in each frame in an iterative fashion. Each vector is considered in turn. If there is no existing

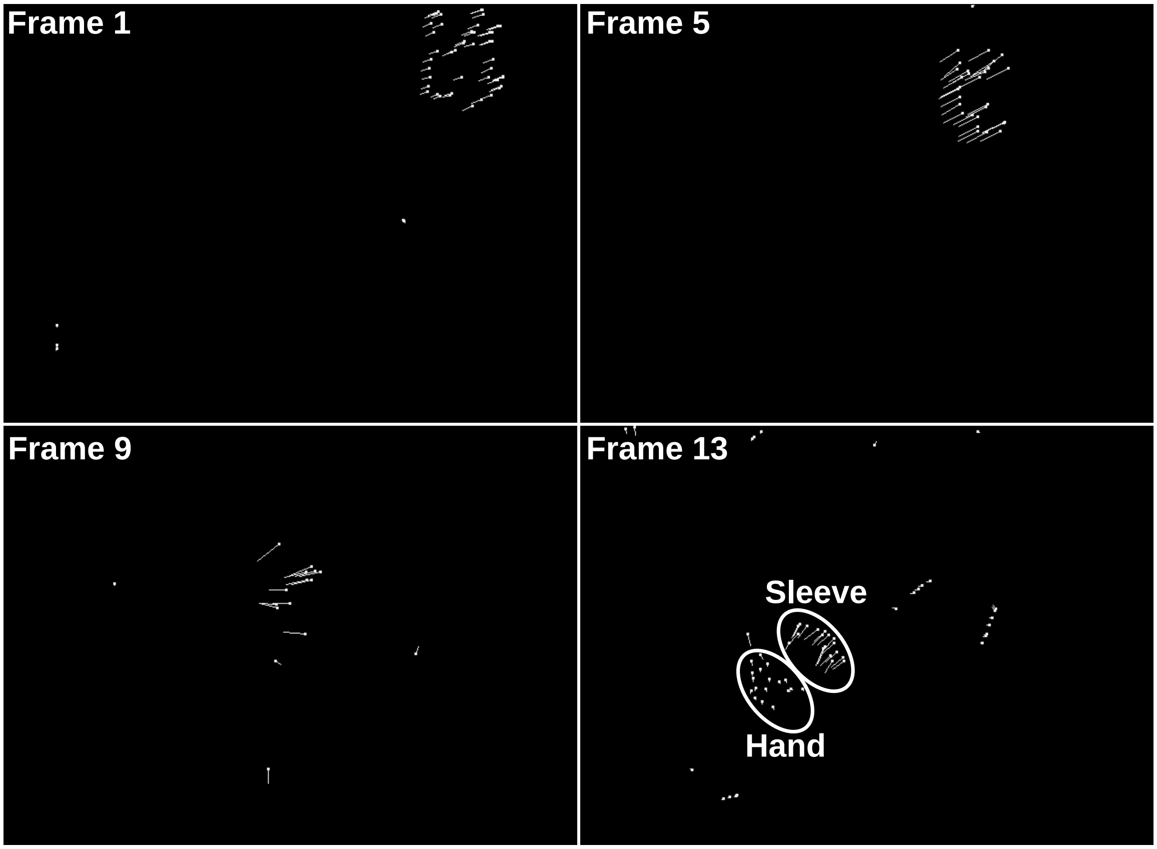

Figure 9: The smoothed velocity field computed in frames 1, 5, 9, and 13 of the 15 frame (0.5 second) gesture sequence

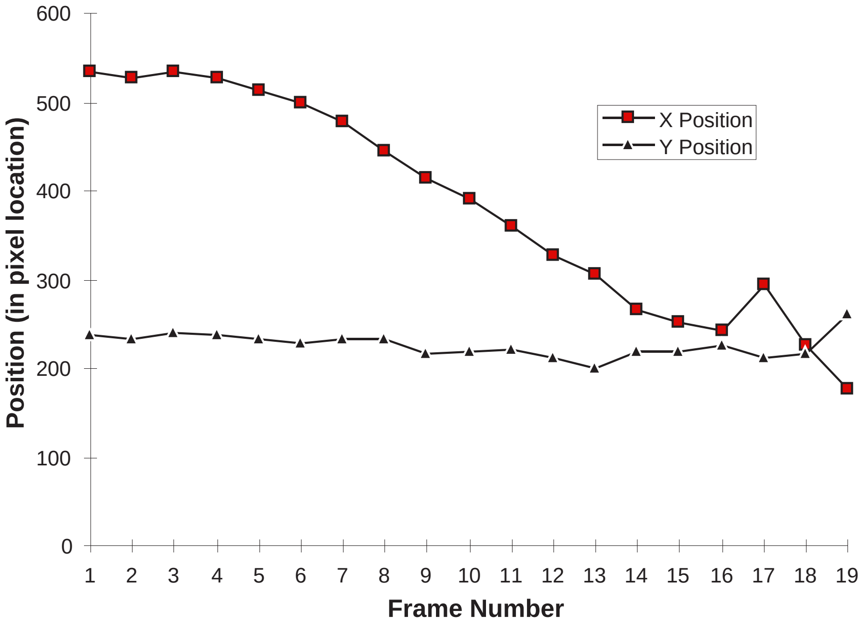

Figure 10: Position plot of a left hand performing a 'right' gesture

cluster to which it is compatible according to said criteria, a new one is created to include it. If one exists, the vector is included to the cluster. Whenever a new vector is added to a cluster, the centroid of the vector origins and the average direction of the vectors in the cluster are updated.

Once the vectors in each frame are clustered, the system locates the gesturing hand as the fastest moving cluster across the frames. Here, we apply a path cohesion criterion. The location of centroid and average direction of each cluster is used to predict the location of the hand in the next frame by linear projection. The fastest moving cohesive path is assumed to be the one representing the moving hand.

6 Results

We performed our experiments on video data by first taping the gesture on a Hi-8 video tape and digitizing the sequence frame by frame. We are therefore able to process 30 frame-per-second data at the full 640 × 486 resolution of our frame acquisition system.

Figure 10 is a plot of the centroid position of a cluster of vectors representing a gesturing hand (the centroid is computed on the vector origins of the cluster). The gesture is the 'right' gesture performed by the left hand (the left-handed version of gesture 6 of the gesture vocabulary in figure 4). In this gesture, the hand moves in a horizontal path from right to left across the video image. Hence, the y-positions of the cluster are relatively constant while the x-position decreases. At frame 17, the system missed the vectors representing the bottommost digit of the hand, resulting in a shift of the centroid of the cluster upwards.

Figure 11 plots the x and y components of the average velocity of the cluster. The vector velocities computed by the absolute difference correlation and variance-based smoothing. Note that these velocities are computed independently from the centroid positions in fig-

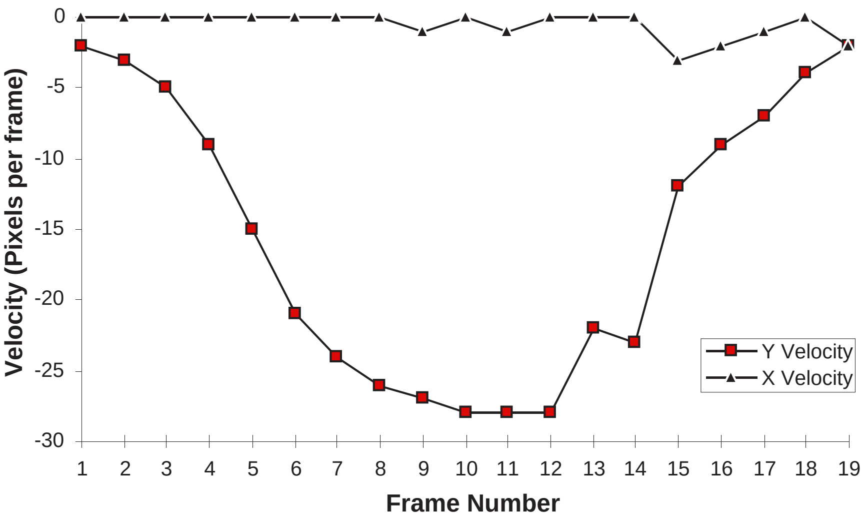

Figure 11: Velocity plot of a left hand performing a 'right' gesture

ure 10. The x-velocity shows a characteristic acceleration-deceleration curve as the hand moved from left to right. The y-velocity remained relatively constant for the horizontal gesture.

Figure 12 is a plot of the average vector magnitudes and directions of the same cluster. In this plot, the direction of the hand motion is clearly seen to be constant except for the end of the gesture when the subject lowered his hand slightly.

The system was able to process the full 640 × 486 video at a quarter of a frame per second in software running on a 150 MHz Silicon Graphics Indigo^{II} workstation. Reducing the image size to 320 × 240, the system operated at a half a frame per second.

7 Conclusion

In the science fiction motion picture *Star Trek IV: The Journey Home*, the Star Trek crew was transported back in time to the present. When "Scotty" was confronted with a contemporary Apple Macintosh computer, he picked up the mouse, and, with a look of great bewilderment, brought it to his mouth and intoned in his characteristic brogue, "Hello computer." This illustrates the point of this paper well. The interface devices which we use to communicate with computers are by no means the 'way things ought to be'. They are the result of the accidents of history and the innovation of researchers working with the tools of their day. The layout of our keyboard, for example, results from the antiquated manual typewriter.

We believe that computer vision has a significant role to play in the human-computer interaction devices of the future. We have outlined our vision for the role computer vision will play, highlighted the impediments to the development of vision-based interfaces, and proposed an approach for overcoming these impediments. We illustrated our approach by discussing our work on vision-based hand gesture interfaces.

We believe that for vision-based interfaces to make any impact, we will have to embark on an expansive approach which begins with the study of the interaction modality we seek to implement. For gesture interfaces, this involves a familiarity with the relevant literature

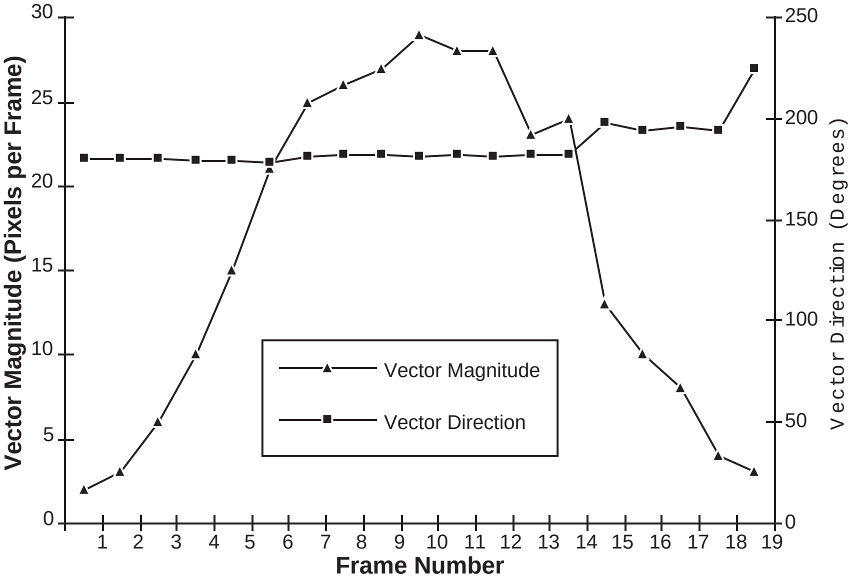

Figure 12: Magnitude and direction vector plot of a left hand performing a 'right' gesture

from such varied disciplines as semiotics, anthropology, neurophysiology, neuropsychology and psycholinguistics. It also involves a dialogue with researchers in these fields. Armed with such information, we can develop models which can be reasonably realized with computer vision.

In our survey of the literature in vision-based interaction, as well as in our work on hand gesture interpretation, we have found that dynamic vision plays a key role in interfaces. This comes as no surprise since humans operate in a dynamic world and interact within the domain of space and time. We believe that color vision will also play a significant role in realizing vision-based interfaces.

We have developed a system which computes the motion of a gesturing hand based on the information gleaned from human gesture usage. Our system employs a moving edge detector which accentuates moving edges and suppresses stationary ones. The edges were localized by *non-maximal suppression*. A window-based algorithm extracted dominant edge points from the localized edges. These points were used as seed points for *absolute difference* correlation between successive images in the video stream. A flow field conforming to four constraints: directional variance, spatial cohesion, directional cohesion and path cohesion and is computed from vector candidates generated by ADC. Within eeach frmae, a field which minimizes a local directional variance constraint is computed by a local hill climbing algorithm. The application of the variance constraint effectively smoothed the field of vectors and aligned local fields while allowing variation across the entire image. These vectors are then clustered by spatial location and direction. Finally, these clusters are grouped across frames by path cohesion to determine the dynamic path traced by the hand. We present plots of the typical output of our system which validate our computational approach.

Our unoptimized code was able to achieve 0.25 to 0.5 frame-per-second operation on a general purpose single CPU workstation. We expect that the target update rate for real-time implementation should be about 60 frames-per-second (the typical mouse pointing device update rate). This means that our system will be deployable in real operation with a two order increase in porcessing speed. We expect that the interface methodology will be testable at between 10 to 30 frames per second. We believe both of these goals are reachable through parallelization and faster computation.

8 Acknowledgement

This research is supported by the National Science Foundation, Grant Number IRI-9396329.

References

[1] M.A. Clarkson, "An easier interface", *Byte*, vol. 16, pp. 277–282, Feb. 1991.

[2] S. Akamatsu, T. Sasaki, H. Fukamachi, and Y. Suenaga, "Automatic extraction of target images for face identification using the sub-space classification method", *IEICE Transactions on Information and Systems*, vol. E76D, pp. 1190–1198, Oct. 1993.

[3] Roberto Brunelli and Tomaso Poggio, "Face recognition - features versus templates", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 1042–1052, Oct. 1993.

[4] K. Mase, "Recognition of facial expression from optical flow", *IECE Transactions*, vol. E 74, pp. 3474–3483, 1991.

[5] M.A. Turk and A.P. Pentland, "Eigenfaces for recognition", *Journal of Cognitive Neural Science*, vol. 3, pp. 71–86, 1991.

[6] B.A. Golomb, D.T. Lawrence, and T.J. Sejnowski, "Sexnet: A neural network identifies sex from human faces", in D.S. Touretzky and R. Lippman, editors, *Advances in Neural Information Processing Systems 3*. Morgan Kaufman, San Mateo, CA, 1991.

[7] H. Li, P. Roivainen, and R. Forcheimer, "3-D motion estimation in model-based facial image coding", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 545–555, June 1993.

[8] D. Terzopoulos and K. Waters, "Analysis and synthesis of facial image sequences using physical and anatomical models", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, pp. 569–579, June 1993.

[9] K. Mase and A. Pentland, "Automatic lipreading by optical flow analysis", *Systems and Computers in Japan*, vol. 22, pp. 67–78, 1991.

[10] K. Mase and A. Pentland, "Lip reading by optical flow", *Transactions of the Institute of Information and Communications Engineering*, vol. J73-D-II, pp. 796–803, 1990.

[11] S. Nishida, "Speech recognition enhancement by lip information", *ACM SIGCHI Bulletin*, vol. 17, pp. 198–204, 1986.

[12] A. Pentland and K. Mase, "Lip reading: Automatic visual recognition of spoken words", Technical Report 117, M.I.T. Media Lab Vision Science, 1989.

[13] K. Mase, Y. Watanabe, and Y. Suenaga, "A realtime head motion detection system", *in Proceedings of the SPIE, 1260*, pp. 262–269, 1990.

[14] K.P. White, T.E. Hutchinson, and J.M. Carley, "Spatially dynamic calibration of an eye-tracking system", *IEEE Transaction on Systems, Man, and Cybernetics*, vol. 23, pp. 1162–1168, 1993.

[15] Robert J.K. Jacob, "Eye-gaze computer interfaces:What you look at is what you get", *IEEE Computer*, vol. 26, pp. 65–67, July 1993.

[16] Robert J.K. Jacob, "Eye movement-based human-computer interaction techniques", in H.R. Hartson and D. Hix, editors, *Advances in Human-Computer Interaction, Volume 4*, pp. 151–190. Ablex Publishing Company, 1993.

[17] Masaaki Fukumoto, Kenji Mase, and Yasuhito Suenaga, "Real-time detection of pointing actions for a glove-free interface", *in Proceedings of IAPR Workshop on Machine Vision Applications*, Tokyo, Japan, Dec. 1992.

[18] T. Baudel and M. Beaudouin-Lafon, "Charade – Rmote control of objects using free-hand gestures", *Communications of the ACM*, vol. 36, pp. 28–35, July 1993.

[19] Brigitte Wirtz and Christoph Maggioni, "Imageglove: A novel way to control virtual environments", *in Virtual Reality Systems'93*, pp. 7–12, 1993.

[20] Jakub Segen, "Controlling computers with gloveless gestures", *in Virtual Reality Systems'93*, pp. 2–6, 1993.

[21] Francis Quek, "Hand gesture interface for human-machine interaction", *in Virtual Reality Systems'93 Spring Conference*, pp. 13–19, New York, Mar. 15-17 1993.

[22] Francis Quek, "Vision-based gesture interpretation", *in Virtual Reality Systems'93 Fall Conference*, New York, 1993.

[23] Francis Quek, "Toward a vision-based hand gesture interface", *in Virtual Reality Software and Technology Conference*, pp. 17–29, Singapore, Aug. 23-26 1994.

[24] A. Azarbayejani, T. Starner, B. Horowitz, and A. Pentland, "Visually controlled graphics", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 16, pp. 602–605, June 1993.

[25] F. Elsner, K. Hacine, K. Naceur, J-C. Angue, and M. Bourton, "Proposition of a human motion tracking method by temporal-spatial segmentation in an image sequence", *in Proceedings of the IAPR Workshop on Machine Vision Applications*, pp. 119–122, Tokyo, 1992.

[26] J.Z.C Lai, "Tracking multiple features using relaxation", *Pattern Recognition*, vol. 26, pp. 1827–1837, Dec. 1993.

[27] Peter C. Lombrozo, Ronald E. Barr, and Lawrence D. Abraham, "Smoothing of noisy human motion data using digital filtering and spline curves", *in IEEE Engineering in Medicine and Biology Society 10th Annual International Conference*, pp. 653–654, 1988.

[28] P. Ekman, T.S. Huang, T.J. Sejnowski, and J.C. Hager, "Final report to NSF of the planning workshop on Facial Expression Understanding", Report, NSF, July 30 to August 1 1992.

[29] N. Badler, "Final report to nsf of the standards for facial animation workshop", Electronic communication of report draft, 1994.

[30] M.L.A. Sternberg, *American Sign Language, A Comprehensive Dictionary*, Harper and Row Publishers, New York, 1981.

[31] Paul McAvinney, "Telltale gestures", *Byte*, vol. 15, pp. 237–240, July 1990.

[32] James D. Foley, "Interfaces for advanced computing", *Scientific American*, vol. 257, pp. 127–135, Oct. 1987.

[33] S.S. Fisher and J.M. Tazelaar, "Living in a virtual world", *Byte*, vol. 15, pp. 215–221, July 1990.

[34] S.S. Fisher, "Virtual interface environments", in B.Laurel, editor, *The Art of Human-Computer Interface Design.* Addison-Wesley, Reading, MA, 1990.

[35] S.S. Fisher, C.C. Wenzel, and M.W. McGreevy, "Virtual interface environment workstations", *in Proceedings of the Human Factors Society, 32nd Annual Meeting*, 1988.

[36] S.S. Fisher, C.C. Wenzel, and M.W. McGreevy, "Virtual workstation: A multimodal, stereoscopic display environment", *in Proceedings of the SPIE Conference of Intelligent Robots and Computer Vision*, vol. 726, pp. 517–522, 1986.

[37] Steve Ditlea, "Another world: Inside artificial reality", *PC/Computing*, vol. 2, pp. 91–102, Nov. 1989.

[38] Robert Wright, "Virtual reality", *The Sciences*, vol. 27, pp. 8–10, Nov./Dec. 1987.

[39] J.C. Chung et al., "Exploring virtual worlds with head-mounted displays", *in Proceedings of the SPIE Conference on Three-Dimensional Visualization and Display Technologies*, vol. 1083, pp. 42–52, 1989.

[40] J. Sibert et al., "Issues limiting the acceptance of user interfaces using gesture input and handwriting character recognition", *in Proceedings of CHI & GI*, pp. 155–158, 1987.

[41] E. Sachs, "Coming soon to a CAD lab near you", *Byte*, vol. 15, pp. 238–238, July 1990.

[42] Richard A. Bolt, "Conversing with computers", *Technology Review*, vol. 88, pp. 34–43, Feb./Mar. 1985.

[43] Mark B. Friedman, "Gestural control of robot end effectors", *in Proceedings of the SPIE Conference of Intelligent Robots and Computer Vision*, vol. 726, 1986.

[44] R. Pausch and R.D. Williams, "Giving candy to children – user-tailored gesture input driving an articulator-based speech synthesizer", *Communications of the ACM*, vol. 35, pp. 58–66, May 1992.

[45] Pierre Wellner, "Interacting with paper on the DigitalDesk", *Communications of the ACM*, vol. 36, pp. 87–95, July 1993.

[46] Howard Eglowstein, "Reach out and touch your data", *Byte*, vol. 15, pp. 283–290, July 1990.

[47] T.G. Zimmerman et al., "A hand gesture interface device", *in Proceedings of CHI & GI*, pp. 189–192, 1987.

[48] C. Blanchard et al., "Reality built for two: A virtual reality tool", *in VPL Research, Inc.*, Redwood City, CA 94063, 1989.

[49] S.S. Fels and G.E. Hinton, "Glove-talk - A neural network interface between a data-glove and a speech synthesizer", *IEEE Transactions on Neural Networks*, vol. 4, pp. 2–8, Jan. 1993.

[50] Roberto Cipolla, Yasukazu Okamoto, and Yoshinori Kuno, "Qualitative visual interpretation of 3d hand gestures using motion parallax", *in IAPR Workshop on Machine Vision Applications*, Dec. 1992.

[51] J. Bulwer, *Chirologia: Or the natural language of hand and Chironomia: Or the manual art of rhetoric*, Southern Illinois University Press, Carbondale, IL, [1644] 1973.

[52] J.-L. Nespoulous, P. Peron, and A.R. Lecours, *The Biological Foundations of Gestures:Motor and Semiotic Aspects*, Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[53] David McNeill, *Hand and Mind*, University of Chicago Press, Chicago, 1992.

[54] J.-L. Nespoulous and A.R. Lecours, "Gestures: Nature and function", in J-L Nespoulous, P. Peron, and A.R. Lecours, editors, *The Biological Foundations of Gestures:Motor and Semiotic Aspects*, pp. 49–62. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[55] Richard A. Bolt, "The integrated multi-modal interface", *The Transactions of the Institute of Electronics, Information and Communication Engineers (Japan)*, vol. J70-D, pp. 2017–2025, 1987.

[56] E.S. Klima and U. Bellugi, "Language in another mode", *Language and Brain: Developmental Aspects, Neurosciences Research Program Bulletin*, vol. 12, pp. 539–550, 1974.

[57] E.S. Klima and U. Bellugi, *The Signs of Language*, Harvard University Press, Cambridge, MA, 1979.

[58] Adam Kendon, “Current issues in the study of gesture”, in J-L Nespoulous, P. Peron, and A.R. Lecours, editors, *The Biological Foundations of Gestures:Motor and Semiotic Aspects*, pp. 23–47. Lawrence Erlbaum Associates, Hillsdale, NJ, 1986.

[59] Adam Kendon, “Gesticulation and speech: Two apsects of the process of utterance”, in M.R. Key, editor, *Relationship Between Verbal and Nonverbal Communication*, pp. 207–227. Mouton, The Hague, 1980.

[60] D. McNeill and E. Levy, “Conceptual representations in language activity and gesture”, in R.J. Jarvella and W. Klein, editors, *Speech, Place, and Action*, pp. 271–295. John Wiley & Sons Ltd., 1982.

[61] S.M. Haynes and R. Jain, “Time-varying edge detection”, *Computer Graphics and Image Processing*, vol. 21, pp. 345–393, 1983.

[62] Ellen C. Hildreth, “Computations underlying the measurement of visual motion”, *Artificial Intelligence*, vol. 23, pp. 309–354, 1984.

[63] R. Agarwal and J. Sklansky, “Estimating optical flow from clustered trajectories in velocity-time”, *in Proceedings of the 1992 International Conference on Pattern Recognition(1992)*, Sep. 1992.