

2 Background

This chapter will examine the state of the art in hand gesture recognition from three points of view: the study of hand gestures themselves, recognition of hand gestures, and how hand gestures have been applied to various interface tasks.

2.1 Hand Gesture Theory

Any serious attempt to interpret hand gestures should begin with an understanding of natural human gesticulation. In the course of this work the references discussed below have proven valuable.

The physical structure and limitations of the hand form the ultimate limits of what can be done with gesture. While this work does not make explicit use of hand models, it has been useful to have some knowledge of the physical level. In [LK95], Lee and Kunii have looked at the physiology of hand movement from the point of view of computer modeling. They analyze the range of motions of individual joints and other characteristics of the physiology of hands. Their goal is to build an accurate internal model where the parameters can be set from imagery.

In Sturman's PhD dissertation [St92] he investigates the use of hands for computer interaction independent of an application or sensing modality. He discusses characteristics of the hands that might help determine which tasks are suitable for hand gestures, and which are not, and develops a design method that can be used to develop and evaluate a particular application.

Francis Quek of the University of Michigan has studied natural hand gestures in order to gain insights that may be useful in gesture recognition ([Qu93]). So far applications

are somewhat limited [Qu95], but many of the observations are both interesting and useful. For example, he makes the distinction between inherently obvious (transparent) gestures that need little or no pre-agreed on interpretation, and iconic (opaque) gestures. He observes most intentional gesticulation generally is, or soon becomes, iconic. Other classifications are whether a gesture is indicating a spatial concept or not, and whether it is intentional or unconscious. He indicates that humans typically use whole hand *or* digit motion, but seldom both at once. He suggests that if a hand moves from a location, gesticulates, then returns to the original position, the gesticulation was likely to be a gesture rather than an incidental movement of the hand. This work will make use of several of these observations. The concepts of the exclusivity of digit versus whole hand motion, and of a home base for movement, played a large roll in the basic design of the interaction language.

McNeill's interest in [Mc92] is mostly spontaneous gesticulation that accompanies speech. He discusses how it interacts with speech, and what it can tell us about language. A basic premise is that gestures complement speech, in that they capture the holistic and imagistic aspects of language that are conveyed poorly by words alone. Most applicable here, he sites work that indicates that gestures have three phases: preparation, where the hand rises from its resting position and forms to the shape that will be used; stroke, where the meaning is conveyed; and retraction, where the hand returns to its resting position.

Cassell [Ca94] uses these types of observations about spontaneous gesture to drive the gestures of animated characters in order to test specific theories about gesticulation and communication. She has also been a part of gesture recognition projects at the MIT Media Lab [WBC96].

2.2 Hand Gesture Recognition

Until a few years ago, nearly all work in this area used mechanical devices to sense the shape of the hand. The most common device was a thin glove mounted with bending sensors of various kinds, such as the DataGlove by VPL [ZL87]. Often a magnetic field sensor was used to detect hand position and orientation, but that was notorious for having problems in the electro-magnetically noisy environment of most computer labs. The shape of the hand was most commonly classified using an approach similar to that developed by VPL, where joint angle vectors index into lookup tables.

Researchers reported successful systems in areas such as CAD design [WG89], sign

language recognition [TK91] control of presentations [BB93], and a range of other areas [SZ94], but in spite of readily available sensors, interest in glove-based interaction faded. Some possible reasons for this were discussed in Section 1.2.

Some of the earliest work using visual recognition was done as part of Kreuger's VIDEOPLACE [Kr91]. While the work did not focus on hand gestures, he was able to successfully incorporate basic gesture input into his interactive environment. Using environmental constraints and a strong sense of context he was able to get interesting results with the limited computer power available to him, a theme shared with this work. Unfortunately the unique nature of that environment also means that many of his techniques did not generalize well to other problems. Some details of his approach are discussed in Section 2.3.1.

More recently the increasing computer power available and advances in computer vision have come together with the interest in virtual reality and human centric computing to spark a strong interest in visual recognition of hand gestures. The interest seems to come from many sectors. The difficulty of the problems involved make it a good application domain for vision researchers interested in recognition of complex objects (e.g., [CW95][KH95]). The freedom from the need for interface devices has the interest of those working on virtual environments (e.g., [DP95]). The potential for practical applications is sparking work on various aspects of interface design (e.g., [FW95][Cr95]). The proceedings of the first two International Conferences on Automatic Face and Gesture Recognition ([FG95] and [FG96]) provided some of the first good collections of papers on the topic.

The remainder of this section presents an overview of current research in several key areas pertinent to this work; hand segmentation, pose recognition, and motion interpretation. Section 2.3 will then discuss the ways in which gesture has been put to work for various applications. For additional pointers to other work in hand gesture recognition see Huang and Pavlovic's survey [HP95].

2.2.1 Hand Segmentation

Many researchers have avoided the difficulties of segmenting the hand from an arbitrary background, using either uniform color backgrounds ([Kr91][UV95][NB95][RK93]), marked or colored gloves ([SP95][DS93][WM93]) or devices such as light tables ([Se93]). This approach allows the research to focus on other aspects of the problem but it has a major weakness. Algorithms developed in these circumstances often

come to depend on the high quality segmentation available, or the unique characteristics of the segmentation method used. This will limit their application in domains where such segmentation aids are not practical.

One way of finding a hand in a cluttered image is to use knowledge of the object you are looking for. This can be done by deforming a 2D hand model to fit image data. By limiting deformations to those which correspond to actual hand poses you can get a best fit to edges or blobs in the image within the bounds of a realistic interpretation. The result is a 2D description of the hand in the image that can be used for either segmentation or pose interpretation.

A good example is the work of Heap and Samaria, described in [HS95]. They extract the hand from an environment very similar to that used here, except that the camera looks back at the user from above the screen. Good lighting is assured by additional light sources to the sides of the screen. The hand image is preprocessed to highlight the red band and de-emphasizing the blue and green bands. The hand-outline model is based on control points where the average position and principal modes of variation for each are determined from a set of training outlines. The initial outline is placed in the image by a genetic algorithm, and iteratively deformed within the constraints of the model to lie along image edges. Similar work is described by Kervrann and Heitz in [KH95].

Iteratively deforming a simple outline model works well for situations where the approximate location and expected pose of the hand is known in advance, and what is needed is a detailed description. For example if the palm of the hand is displayed, this method can determine how the fingers are spread. Generally, however, the pose of the hand is unknown, so that the initial shape of the model may be significantly different from the shape of the hand, and the simple approach is not likely to perform well.

Cui and Weng [CW95] avoid needing to know the pose in advance by using a two step process where the first step searches a database of learned hand poses. Training examples are examined to find a set of most discriminating features (MDFs). Each training example then becomes identified with some region of the space defined by these features, and a space-partition tree is defined to allow efficient searching through the regions. A new image is classified by first searching the tree to find the closest training example in MDF space. The outline of this example is then used to initialize a deformable model, and an iterative procedure adjusts the model to edges in the original image. The resulting model can be used to segment the hand. Since the initial database search is based on appearance, it requires a training set that is representative of the expected range of

variation in size, orientation, lighting, etc.

Model-based segmentation has the advantage of extracting an accurate outline and a coherent region without voids. Unfortunately, due to their complexity they tend not to run in real time. They often require a complex hand-built model that includes details such as constraints on deformations. Most current model-based approaches make limiting assumptions on the expected shape of the hand, and rely on knowing in advance exactly where the hand is in the image, and in many cases how it is oriented. So while model-based segmentation is not yet ready for realistic applications, it does provide promise for the future.

The approaches that currently have the most potential for real-time applications use local image clues, such as motion or color. The shortcomings of using such low-level information are generally compensated for by using knowledge about the expected task or imaging environment.

In many applications, the hands are moving against a stable background. As a result, a common technique for segmenting hands is to look for movement. A good example is Brand in [Br96], which looks for changes in a learned background to segment blobs representing the users hands and objects they are manipulating. He uses heuristics on the relative movement of the blobs to derive a series of causal events that describe what the user has done. Freeman in [Fr96] describes an artificial retina chip which can detect moving objects at greater than video rates. He shows how it can be used to allow users to control games with hand and body gestures using very coarse estimates of the box bounding the moving object. Using movement as the first step in a longer process has the potential to give a high quality segmentation. Cui and Weng in [CW96] use motion to give an initial estimate of where the hand may be in an image, then segment in detail using knowledge of the expect poses, as described above.

It is possible to segmentation skin using only color, because its color is often relatively unique in the environment. Many commercial video-effects systems (e.g., [I94]) support basic segmentation by allowing the user to set a bounding range for each dimension of color-space and identifying pixels with colors in that region. This approach can be used to segment skin, but there are two problems that must be addressed. The first is that a cube in color-space does not match the shape of the region defined by the range of colors found in skin. Defining a large enough cube to include the range of colors present on even a single user with even lighting will also include many colors that are never found on skin, leading to false positive regions being segmented. Secondly, the apparent color

of skin is highly variable, depending on lighting, skin tone, color shift during image capture and transmission, and a host of other causes. Pre-defining a static region of color space may work well in controlled conditions, but does not have the flexibility for realistic systems.

The regions of color space where skin colors fall can be identified by training. In [Ma95], Maggioni mentions a color segmentation scheme based on regions in an RGB color lookup table where the table is trained interactively, but few details are provided. Quek [Qu95] also uses a simple RGB lookup table trained from a pre-segmented image of the hand. Both these systems have the hand against simple and controlled backgrounds. Maggioni looks down on the hand from above, so the desk top forms the background, while Quek looks down on the hand with the keyboard below. Quek discusses an improved method based on HSI color space that will be more resistant to changes in lighting.

Schiele and Waibel [SW95] address the variability problem using a probabilistic technique. A probability function over normalized RGB color space is created using data from the faces of several different people. This is used to place a bounding box around the face in a larger image by labeling the pixels with their skin color probability and bounding the resulting region. Within that box the skin colors are intensified, by scaling them by the same probability function. This suppresses the background and highlights skin. In this work, the resulting region is intensity normalized and passed to a neural net to determine the gaze direction. One other interesting note about this work is that the skin color probability function automatically adjusts to the user's skin color as it is used.

Because of the problems using only color, some work combines color with other clues such as edges, outlines and motion to give better results ([Gr96][SF96]).

Some work combines color and change detection to increase the range of circumstances where hands can be detected. In [Wr95] the authors describe a static camera that observes a room and determines the expected range of color values for each pixel over time when it is empty. Any large region where the pixel colors deviate from the expected values is assumed to be a person. Contour analysis is used to identify extended limbs, and color analysis to find potential skin regions. The results are combined to reliably locate the hands, whether they are an extremity of the contour or in front of the body.

Approaches using local image clues like these have the advantage of speed and simplicity. The disadvantage is that they generally do not produce as accurate a

segmentation as model-based approaches. In particular they are sensitive to such “noise” as shadows, color bleeding from a poor video signal, blooming around light sources, colored background objects, etc. As a result, the bulk of the desired object (hand) is usually intact, but it may have holes or attached regions, and the outline is not likely to be accurate. Without some knowledge of the expected shape these local errors can not be corrected. This has implications for any processes using the result of the segmentation. In particular algorithms that rely on an accurate outline are at a disadvantage.

2.2.2 Pose Recognition

One way to dissect components of a gesture is into the 3D path traced out by the hand as it moves, and the poses it forms at various times. The majority of the work up till now in “hand gesture recognition” focused exclusively on the pose.

3D Hand Modeling:

A popular theme for hand pose recognition is to build an internal 3D model of the hand, so that a detailed recognition step can be performed later.

In [RK93] Regh and Kanade describe how they track a hand with an internal model in real-time. Tracking is initialized by having the user place their hand in a known position. As the hand moves, the current model is projected into the image plane and lines in the image perpendicular to the model fingers are searched to find the edges of the real fingers. The model is then moved to a new configuration by minimizing the difference between the model and image points using least squares. The approach relies on fast tracking so that differences between model and image points at any one step are minimized. The method has only been demonstrated with the hand against a black background, and without self-occlusion. The problem of interpreting the model for anything more complex than cursor movement is not addressed. Vaillant and Darmon describe a similar but more limited system in [VD95].

Lee and Kunii [LK95] begin with an anatomically accurate hand model, represented as a tree where fingers are branches off the palm, etc. On the basis of their analysis they suggest that the entire pose of the hand can be uniquely identified using 7 key points. Using a stereo image of a hand wearing a glove marked with colors to identify these 7 points, the orientation and pose of the hand are adjusted using an iterative model based on minimizing torque's produced by spring tension. Each iterative step first adjusts the wrist orientation then moves out toward the fingers. One interesting aspect of this model is

that each point has a measure of importance that is used to weight the effect of its spring during parameter fitting. Again, pose interpretation is not addressed.

Heap [HH96] use a surface-mesh-based deformable model which they call a Point Distribution Model. The model was derived using MRI data. They fit it to image edges in real time using weighted least squares. The result is very good for rendering, but they have not addressed symbolic interpretation.

Using a 3D hand model has two primary advantages. First, the model describes the entire hand, including portions hidden from the view of the camera. The configuration of the hidden portions of the hand are reconstructed using constraints provided by the portions that are visible. Secondly the resulting representation of the hand is at a high level — using shape features such as joint angles that are easy for people to understand. As a result, such models are useful for creating accurate renderings of the hand and for describing the shape of the hand as a set of compact and understandable parameters.

It is interesting, though, that no work seems to have addressed interpretation of the pose of a 3D model in any realistic way. It is not clear that this is going to be a trivial task. Hand gesture is inherently a visual rather than a tactile form of communication. Features that are easy to differentiate visually may be very sensitive to certain parameters of a model. For example whether or not two extended fingers are touching depends on a small difference in the abduction angles between them, and so may be difficult to accurately determine from a joint-angle representation. (For a more complete discussion of these ideas see Section 3.5.1) This argument suggests that pose recognition based on the appearance of the hand in the image may be more reliable as well as being potentially simpler.

Appearance-based pose recognition:

Quite a bit of work has been done which attempts to classify the hand pose using its appearance in the image. Approaches can be classified into three broad categories: 2D hand models, template matching and various techniques that use local image features to derive a feature vector that is then matched.

In [HS95], described above, Heap and Samaria determine the deformation parameters of a 2D hand model as part of the segmentation process. The deformation parameters are examined to classify the pose, but few details are given as to how. The method relies on an initial estimate of hand location and pose, and can only deform to track minor deformations to that pose (the exact position of fingers leaving the palm, for example). It

is not clear how the method would be extended to differentiate fundamentally different poses, such as a palm and a fist.

In [Se92], and [Se93] Segen describes the application to hand pose recognition of the edge based techniques he explored earlier in [Se89]. Starting with a high quality binary hand image, he finds distinctive points on the outline of the hand using local features such as curvature maxima, clusters these points into sub-parts and uses the clusters to index into a database of hand shapes. He has demonstrated the approach discriminating several different poses in real time. Scale provides limited 3D information. The primary weakness of this method is its sensitivity to segmentation quality because of its reliance on local curvature. Interesting results might be obtained if this were combined with a model based segmentation scheme such as Heap and Samaria's [HS95].

The most basic appearance based technique is to match the incoming image against a set of pose templates. The problem with this, of course, is that the training set must be representative of all expected views of all poses, in all possible lighting conditions, making it very large. Matching an incoming image against a large set of templates in image-space is very time consuming. The matching can be made more efficient by reducing the images to a lower dimensional space, but this must be done in a way that preserves the features that best differentiate the poses. This can be done by performing principal component analysis on the training set and mapping the images onto the basis formed by the first n eigenvectors. An image is classified by mapping onto that basis and finding the nearest training image in eigenspace. A good example of this approach is used by Cui and Weng in [CW95] where, as described above, the nearest neighbor is found using a space-decomposition tree. This training image can then be used for both segmentation and pose recognition.

Another slightly different approach is taken by Moghaddam and Pentland in [MP95]. Here pre-segmented training images are mapped into eigenspace and a probability density function is constructed for each hand pose which determines the probability that some point in eigenspace is a member of that class. To find and classify a hand pose in an incoming image, sub-windows of that image at a range of scales are mapped into eigenspace, and the probability that it is a member of each pose class is determined. The maximum probability determines the pose and location of the hand in the image. Since this work uses only edges, it depends on quality edge data, which can be difficult to extract from in realistic scenes. It also assumes that exactly one instance of a hand pose is present in the image.

While these techniques reduce the amount of computation from pure image-space template matching, the computation involved is still significant. Many simpler ways have been developed to map a hand image into a feature vector, where again, the nearest training example in that feature space can be found.

In [UV95], Uras and Verri use a novel “size function” that uses the topology of the edges of the segmented hand to produce a feature vector for each pose. The paper does not suggest how sensitive it might be to the variations in hand shape between people, to errors in segmentation or to changes in apparent edges caused by changes in lighting.

Freeman and Roth compute the histogram of local intensity gradient orientations as the feature vector ([FR95]). This approach has the advantage that it uses the gray scale information on the hand's interior, rather than just the silhouette or outline information. Unfortunately the characteristics of the matching do not seem to be well suited to hand pose classification. In the examples given in the paper, it confused a pose with two fingers extended with one having 4 fingers extended, and it failed to classify two identical poses that had been slightly rotated as the same.

Hunter ([Hu95]) makes a feature vector using “Zernike moments”. This approach shares much with the eigenspace approaches in that the best set of features is determined by examining the training images. These features are used to create a set of images that, when multiplied by an image, produce a value for each dimension in the feature space. One characteristic of these moments is that they are insensitive to rotation of the image. This can be an advantage or disadvantage, as it increases the possibility of misclassification.

Like many visual pose recognition strategies, [UV95] and [Hu95] use only the silhouette of the hand. This limits recognition of gestures where fingers extend toward the camera or have different positions in front of the palm, e.g., a fist with the thumb inside the fingers, versus having the thumb outside the fingers. It also makes the recognition very sensitive to segmentation quality, as errors generally occur on the outline of the object.

All appearance-based approaches rely on the (often implicit) assumption that the appearance of hand in the image stays relatively constant. Variations in appearance occur both because of variations in the actual shape of the hand, and variations in lighting and other environmental conditions. In many applications, the overall shape of the hand and its location in the image are likely to be relatively stable due to a combination of fixed imaging conditions and limitations of human anatomy (e.g., the hand can only form

gestures within a small range of rotations). However, there will always be some variation due to the natural variability of human action and individual differences in how people form hand poses. Different lighting conditions can produce wide variations in the pattern of light and dark on the interior of the hand region, but again, in many domains lighting remains relatively constant. To be useful in any realistic setting, appearance-based approaches to pose recognition must be able to handle the expected range of variability. This will often imply the need to train the system in its target environment rather than relying on preprogrammed algorithms. In this sense, model-based pose recognition is likely to be more robust.

2.2.3 Motion Interpretation

As mentioned above, most “hand gesture recognition” to this point has been concerned with the pose of the hand, not its motion. When the motion of the hand is considered at all, it is most commonly used in an analog fashion to control the movement of some virtual object. Rarely is the motion of the hand in 3-space examined for any symbolic meaning, either alone or in combination with pose information. This is somewhat surprising, considering that motion information seems to play an important role in hand gestures.

The problem of symbolically interpreting a path in 2-space has been addressed to some extent in work on interpreting mouse or pen based gestures for applications such as editing a paper with proofreader symbols (e.g. Lipscomb in [Li91], and Rubine in [Ru91]). This work might logically be applied to paths in 3-space for hand paths, but apparently this has not been done.

One of the few examples in hand gesture recognition that relies on hand motion is by Starner and Pentland. In [SP95] the authors interpret American Sign Language sentences using primarily the path of the two hands. Pose information is limited to the coarse shape information gained from a bounding ellipse. Information about the hand's position and bounding ellipse is encoded into an eight element feature vector. The gestural sentences are interpreted by recognizing sequences of the vectors using Hidden Markov Models.

Charayaphan & Marble [CM90] describe a vision based system that runs on a custom board in a personal computer and recognizes a very limited subset of American Sign Language. It uses the end points of the gross motion of the hand as the principal index feature. Less important are other features of the motion, and least important is a Hough based approach to determine hand shape.

In [KST93], Koons et.al. describe a multi-modal interface using speech, eye tracking and glove-based gesture recognition. Hand motion is analyzed by extracting coarse direction vectors such as *left*, *forward* and *stopped* from the raw glove data. The groups of vectors occurring between hand pauses are classified as higher level features such as *attack* and *sweep*, which the authors refer to as gestlets. Finally these gestlets are combined with inputs from the other modalities into a coherent interpretation using a frame-based representation.

Wilson, Bobick and Cassell ([WBC96]) use the motion of a speakers hands to classify the gestures as bi-phasic — essentially a Preperation/Retract gesture — or a tri-phasic gesture — a Preperation/Stroke/Retract gesture which contains potentially interesting information. Hidden Markov Models are used to make the classification.

Some work makes use of the time-varying nature of gesture, though they do not make explicit use of the path of the hands. These often examine the sequence of hand poses that occur during a gesture.

In [DP95] and [DP93], Darrell and Pentland represent dynamic gestures as the normalized correlation scores over time for a set of view models. Each view model represents a collection of similar images of the hand that vary slightly in position, lighting, by a different user, etc. Gestures are matched using dynamic time warping to handle differences in speed, and given a match score by combining the scores for each time step. The definition of gesture used here is as a sequence of hand poses and orientations. It does not take into account the location or path of the hand in 3-space.

Wilson and Bobick define gestures as a sequence of states in some configuration space ([WB95]). They define methods to determine those states from training data, and recognize them in test data. They show examples of recognizing paths in 3-space using mechanically acquired data, and of recognizing hand gestures in image data using an eigenspace projection as his configuration space. The hand gestures shown are again sequences of poses and orientations without 3-space path information. In theory this technique could be used to address hand gestures consisting of both the motion and pose of a hand, but this has not been demonstrated.

In [DS93] Davis and Shah rely on the motion of the digits but ignore motion of the hand as a whole. Gestures are modeled as the motion of the fingers between a known initial pose and some final pose, and recognized by first indexing based on which fingers have moved, then matching the motion vector of each finger. This approach is based on a very constrained definition of gesture, and it is not obvious how it might be used in a

realistic setting.

2.3 Applications of hand gesture

Gestures have been primarily applied to three types of tasks: interaction with virtual environments, sign language understanding and as part of a more traditional computer interface. Sign language is a very attractive application, but it has a unique set of problems and potentially includes many of the subtleties of natural language understanding and speech recognition. Because of the focus of this thesis, this section will stick to a discussion of applications in virtual reality interfaces and for more traditional user interfaces.

2.3.1 Virtual Environments

Most glove-based gesture recognition work and a large portion of the early vision-based work was modeled as a direct manipulation interface for virtual worlds. Movement of the user's hand is mapped to movement of a "hand" in some 3D environment. Often the user could "fly" around by pointing in the direction he wanted to travel. In theory the user could then manipulate objects by "grasping" them and turning them around, "pushing" virtual buttons, or "throwing" virtual balls. In practice, however, the interface was essentially symbolic. It is very expensive to compute the points of contact between a virtual hand and some object, then model how the object should respond. So instead, desired actions were triggered by hand poses only vaguely reminiscent of the how the action would be performed in the real world. For example to grasp an object the user would fly their hand into it and form a fist.

This style of interaction does not eliminate the need for more traditional commands. The seeds of this thesis were planted by a group at IBM's research lab who wanted to replace their glove-based gesture recognition system for an immersive virtual environment with a vision-based system. It soon became clear that while hand gestures were used for manipulating virtual environments, they were used almost as often to bring up floating menus and perform other command actions.

Krueger developed a very interesting application called VIDEOPLACE [Kr91]. He created a virtual environment designed to encourage people to use their whole body to interact with it in various simple ways. The user stood against a mono-color screen. Their image was extracted and placed in a simulated environment, which was then

projected on a screen for them to see. The style of the interaction was primarily direct manipulation. In one application the user “painted” on a virtual wall with an extended finger, and erased what they had done with their spread open hand. In another application, one user can manipulate the image of another, shrinking their image or moving it from place to place. Krueger describes some 20 such applications, including VIDEODESK, where the system observes the user's hands on a backlit desk and allows them to interact with text and create simple graphics, such as a spline curve fit to control points that are the user's fingers.

Work at MIT has built on this idea by designing an interactive room [DP95] where one wall is a screen. The system watches the room through several cameras and other input devices such as a phased array microphone. A mirror image is put up for the user to see. In a slight twist from Kreuger, rather than an extracted user being placed in a virtual environment, often simulated objects are placed in the “real world” of the user and the room, giving a “magic mirror” effect. For example a user can interact with a semi-autonomous simulated dog — petting it and playing fetch.

2.3.2 Gesture in Traditional User Interfaces

Relatively little work has attempted to use gesture in a traditional user interface. Of that which has been done, the most common approach is to use gesture as a direct mouse replacement. This has most often been done using indirect positioning. In other words the user moves their hand within some control space to move the cursor about the screen in an analogous fashion, rather than pointing directly at the screen to indicate exactly where they want the cursor to go. Both absolute positioning, where the location of the hand or an extended finger within the control space is mapped directly to a screen location, and relative positioning, as is done with a mouse, have been used. Typically some action, such as a change in pose or a key press by the other hand is used to simulate a mouse click.

Nesi and Bimbo use two cameras are used to position the mouse in 3-space ([NB95]). The hand is observed against a black background in a work space, presumably to the side of the keyboard. The motion of the hand is smoothed using a predictive polynomial filter. To take the place of mouse buttons, three hand poses are used: palm down with the fingers extended and together, rotated from that 90 degrees so the palm faces sideways and the thumb is up, and palm down with the fingers curled in a fist. The poses are differentiated by taking the ratio of the sides of the bounding box of the hand when

viewed from above.

In [Qu95] Quek describes a system called FingerMouse. The system is designed to allow the user to switch from typing to moving the mouse simply by assuming a pointing pose with their hand above the keyboard. The camera looks straight down on the hand from above. A finite state machine examines the shape of the segmented hand to determine when a finger is extended. When it is, the mouse is tracked by the location of the fingertip in the plane above the keyboard. Mouse clicks are triggered by pressing the shift key while pointing. The system has been tested by having users fill out on-screen forms, using pointing to select the field to type in.

Some work has been directed at designing a workstation to make greater use of gesture.

In [Ma95] Maggioni describes several additions to a conventional workstation that allow it to use both hand gestures and head movements. One camera images the user's face, another looks down on a region to the side of the keyboard to image the user's hand. When the hand is on the desk, its position is used to position the cursor like a conventional mouse. When it raises off the desk it enters a 3D mode where movement in the center of the imaged volume positions a 3D cursor. When the hand nears the edge of the control volume it moves the observer's viewpoint of the virtual space. The paper describes several hand poses which can be differentiated, but does not suggest how they might be used.

In [We93] Wellner describes his work on an automated desktop that lets the user seamlessly combine physical and digital media. As part of the interface, a camera positioned above the desk observes the user's hands as he interacts with both paper and digital data projected onto the desktop. The system uses motion to find the hand, and segment it from the background. He then can determine the location that the user's finger or a stylus is pointing to. He makes no attempt to classify the pose. Several novel interaction modes are suggested, such as using the finger to draw a circle around a graphic on a sheet of paper, then pointing to where it should be placed in a digital document. The system would then digitize that portion of the paper and place the digital graphic where the user indicated.