Gesture-Based Control of Spaces and Objects in Augmented Reality

Yaser Yacoob and Larry Davis

Computer Vision Laboratory

University of Maryland

College Park, MD 20742

Abstract

A multi-modal system integrating computer vision and speech recognition to enable interaction with virtual spaces/objects by natural gestures and speech is described. Computer vision algorithms are employed to measure and interpret hand/finger movement of the user. Our research focuses on detection, tracking, recognition and visual feedback of the hand and finger movements in a cooperative user environment, and the integration of gesture and speech recognition for man/machine communication.

1 Introduction

1.1 Background

Humans regularly specify and manipulate real physical space and objects by controlled movements of their arms, hands and fingers; they communicate their intentions and goals most naturally through speech. The arm, hand and body movements range from simple actions, such as pointing at an object in space, to expressing meaning through sign language or to assembling parts into a complex object. Our research focuses on a user's hand gestures and actions to express intentions with respect to an object or part of a

scene in a graphic environment. The environment is rendered to the user through stereoscopic head-mounted-display (HMD), (Sony LDI-D100). A video camera mounted on the user's HMD is employed to recognize the hand/finger gestures in the context of speech and cause modifications to the virtual world.

Human gestures, in conjunction with a limited speech vocabulary, can be categorized into

1. Identification gestures, like pointing, that identify locations/objects in space. So, a person might

   - place an object,
   - identify goals for an object's movement,
   - indicate that a collection of objects be treated as a group,
   - segment groups into subgroups, or
   - change an object's internal state.

   A person might first use speech to indicate the type of identification action to be performed (e.g., group), and then use a hand gesture to control the application of the action (e.g., outline the set of elements to be grouped).

2. Action gestures that specify the movement of an object or a group so that a person could

   - Specify a translation or a rotation of an object, etc.
   - Control a virtual tool to "reshape" the virtual world he inhabits, or
   - Directly apply forces that deform and alter the shape of an object

Our research builds on our prior work for detection of people and their body parts from color video [2], motion estimation of rigid, articulated and deformable motions [1,3,4] and recognition of facial and body activities [1,3] to develop real-time hand gesture recognition.

There is relatively little reported research on un-encumbered interfaces for human-computer interaction focusing on hand and finger movements. For a review of research aspects of gesture use in human-computer interaction see [18]. Sato et al.[21] proposed an augmented reality system that uses an infrared video camera

to increase the robustness of the system for detection of the user's hand and fingers over a horizontal desktop. The augmented reality system was used in manipulating WWW pages and a text book by determining the hand and finger pose in the image. Wu et al. [29] describe a system for tracking and recognizing hand posture with respect to a virtual blackboard. The distance between the person and the camera is relatively large so that a fine analysis of finger gestures was not carried out. Derivation of 3D information of the arm's location was pursued to enable recognition of the intended hand movements. Kang-Hyun et al. [13] describe a system for recognizing hand gestures for human-computer interaction. The user is assumed to wear a green glove which is tracked and analyzed to determine the intention behind the gesture. A state space is maintained to constrain the recognition of user intentions. Lee at al. [16] describe an interactive system for determining the location (on a 2D board) at which a finger is pointing from 2D information by employing homography and morphing. The design utilizes two camera to enable determining the coordinate pointed at on the screen.

1.2 Vision

Our vision for future multimodal interfaces is motivated by the following factors:

- The centrality of a multi-modal interface (e.g., vision and speech). Integration of speech with visual information is needed for achieving effective human-computer interaction in virtual environments.
- The importance of passive vision techniques instead of wearable sensors to avoid imposing constraints on user's use of hands and fingers. Specifically, we design representations for gestures, learning algorithms that can recover these representations from examples and real-time computer vision algorithms that can recognize these gestures from (possibly multi-perspective) video.
- The utility of a closed-loop visual feedback framework to enhance the immersion of the user. The is achieved by texture mapping the virtual scene with the user's hands, achieving a blending of real/virtual scenes.

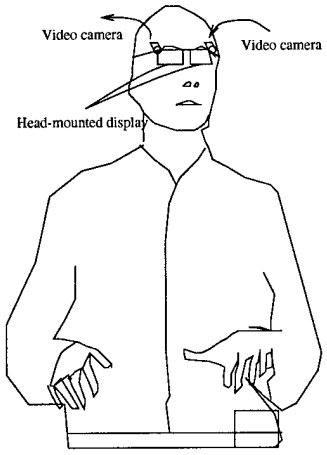


Figure 1: Sketch of Wearable System

Figure 1 sketches the system. One or more video cameras are mounted or embedded on the head mounted display. These cameras generate video streams that capture hand and finger movements within a narrow field of view. The video data is overlayed on a graphical scene that is shown to the user as feedback.

2 System Description

2.1 Overview

Figure 2 shows a high level overview of the system. There are two input streams, an acoustic source from a microphone and a video camera (we currently use a single camera due to the real-time constraint). Both are mounted close to the user to capture the speech and hand movements that occur. The output is a graphic scene that shows the user's hand overlayed onto a synthesized scene, as well as graphical content to reflect the user's actions when those are recognized. The acoustic stream is fed to an off-the-shelf speech recognition engine which in turn passes recognition results to our system. The video consists of a small digital video camera streaming at 30 frames/second.

In the first image the hand is detected using a skin-color classification algorithm [11]. The detected region is analyzed to determine if it corresponds to one of the known hand postures. Currently, it is assumed that the hand can be either in an open hand posture or a grab posture (see Figures 16 and 13). Once a posture

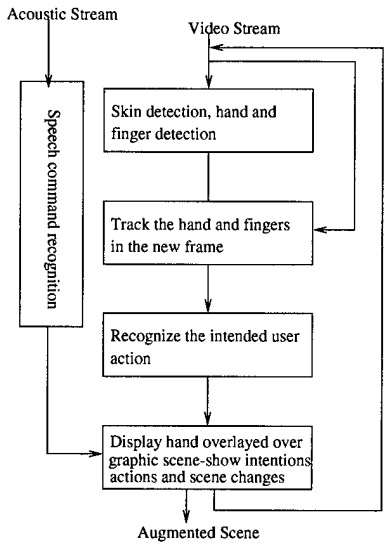


Figure 2: Software Architecture

has been determined, the system detects the back of the hand as well as the fingers and tracks them in subsequent images.

The tracking of hand and finger motions extends the work of Black and Yacoob [5] to estimate the motions of fingers using a 2D motion formulation. The result is a set of parameters that describes the instantaneous motion of the hand and the fingers. These parameters are in turn used by a rule-based system to recognize the occurrence of a limited set of gestures. Currently, the system focuses on Click and Grab actions. Each action is divided into begin-hold-end segments.

The tracked hand and fingers as well as any interpreted actions of the hand are overlayed onto graphical scene such as a map or a 3D scene. The user, accordingly, can move their hand and fingers to affect the synthesized scene and manipulate its contents.

2.2 Speech Recognition

We employ speech recognition software (IBM-Via Voice) to provide text-to-speech and speech-to-text processing. We designed grammars that support understanding the user's spoken sentences and executing them. Currently, speech is used to instruct the system to move the hand to specific locations in the scene since actual movement over large physical space is limited by the narrow field of view of the camera. Also, speech

Figure 3: Illustration of image processing leading to hand and finger detection and labeling, skin color detection, skin region after expansion and contraction and labeling using the convex hull of the region as well as contour depth analysis.

is used to introduce graphic objects into the scene.

Command category	Example
Move Absolute Location	Move to Washington Move to center Move to top-left corner
Move Relative	Move left Move up
Import Object	Import cube Import pen
Remove Object	Remove cube Remove pen
Grouping Command	Begin grouping End grouping
Manipulate Object	Flip object vertically Flip object horizontally
Draw	Draw with index finger Draw with hand

Table 1: The speech input used in the system

2.3 Hand and Finger Detection

Hand and finger detection are critical first steps to enable hand tracking and gesture recognition. Several approaches have recently been reported [8, 20, 26, 32]. Zhu et al. [32] employ Bayes decision theory to develop a hand color model and segment the region assuming that the hand color is consistent within the region. The hand model is computed based on a set of training images pre-labeled manually with hand

and non-hand pixels. A mixture of Gaussians is used to represent the foreground and background colors. Delamarre and Faugeras [8] proposed a stereo-based approach for finding the hand pose in video sequences. A detailed 3D hand and finger model is used to fit the depth information computed from a stereo system. Triesch [26] proposed to recognize hand postures by matching features computed using Gabor filters and Elastic Graph matching. The model graph is derived from a training set of images. Rosales et al. [20] proposed an architecture that computes moment-based features of the hand image regions and matches them to synthesized image regions of hand poses captured using a data glove. This representation allows rendering the hand pose from any view on a viewing sphere. The matching considers a many-to-many mapping that is handled by a specialized mapping architecture.

Skin detection is a straightforward way to detect the hand and finger regions. For a comparative study of skin classification algorithms from color imagery see [12, 23, 25]. Most algorithms perform a statistical analysis of sample skin regions during a modeling stage and then use the model for classification. Models include single Gaussian, Gaussian mixture density model or histograms. Jones and Rehg [12] suggest that histograms have a slight advantage over Gaussian models while Terrillon et al. [25] focused on Gaussian models. Comparative performance involved assessing when a 3D RGB representation is better than a normalized 2D chrominance space normalization. The authors evaluated nine normalized color spaces with respect to a single Gaussian and mixture of Gaussians models of skin. Overall it was concluded that for a small training sample a single Gaussian performs well while mixture models and histograms perform better with un-normalized color spaces and larger training sets (see citations for a complete discussion). Sigal et al. [23] proposed an evolving color distribution model for skin detection. A second order Markov model is used to predict the evolution of the skin color histogram over time. The evolution is parameterized by translation, scaling and rotation in color space.

Our system employs a skin detection algorithm for hand detection and initialization of hand and finger regions for the tracking algorithm (see [11]). The basic assumption is made that for a Lambertian surface the measured color results from brightness and surface spectral reflectance. A model that separates these two components can facilitate color constancy detection by focusing on the surface spectral reflectance

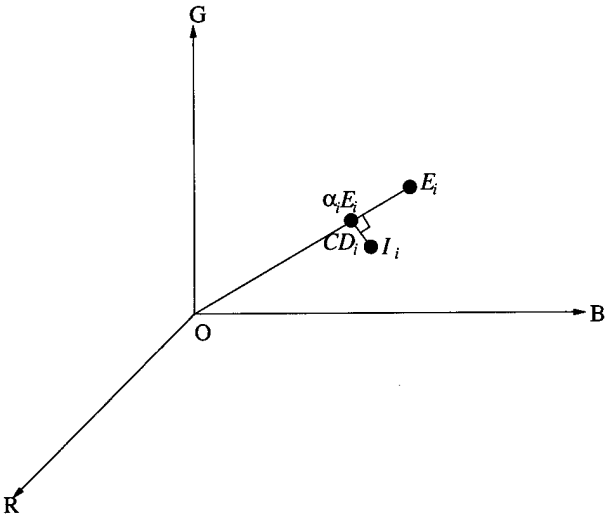


Figure 4: Illustration of brightness and chromaticity color model

component. Figure 4 illustrates the color model proposed in [11]. In the figure $E_i = (E_r(i), E_g(i), E_b(i))$ represents the expected (modeled) color R, G, B values at pixel i and $I_i = (I_r(i), I_g(i), I_b(i))$ is the actual R, G, B color at pixel i. The line OE_i is called the expected chromaticity line. The distortion between I_i and E_i can be attributed to brightness and chromaticity by observing that brightness difference is equivalent to bringing the point I_i to the line OE_i and can be posed as minimization of the error function

$$f(\alpha_i) = (I_i - \alpha_i E_i)^2 \tag{1}$$

where α_i represents the current brightness with respect to the brightness of the model (being greater than 1 if it is more bright and less than 1 if it is less bright). Color distortion, CD_i, is defined as the distance between I_i and α_i

$$CD_i = ||I_i - \alpha_i E_i|| \tag{2}$$

Computing a skin model involves using one or more sample images of skin (manually segmented) and computing a 4-tuple $< E_s, d_s, a_s, b_s >$ where E_s is the mean (R, G, B) value for the skin color, d_s is the standard deviation of the skin color training set, a_s is the variation of the brightness distortion among the points in the training set and b_s is the variation in the chromaticity among these points. This model

is employed to classify skin or non-skin colors while accounting for shadows and highlights as particular brightness values with respect to the model. Once a skin model is computed it does not change unless the illumination in the scene has changed considerably. A pixel i is classified as skin or non-skin based on the observed distortion value α_i that is computed with respect to the color model. A threshold value is used to accept the brightness and chromaticity deviation from the skin color model. Specifically, a pixel is classified by first computing the distortion α_i

$$\alpha_i = \frac{(I_r(i)\mu_r/\sigma_r)^2 + (I_g(i)\mu_g/\sigma_g)^2 + (I_b(i)\mu_b/\sigma_b)^2}{(\mu_r/\sigma_r)^2 + (\mu_g/\sigma_g)^2 + (\mu_b/\sigma_b)^2} \tag{3}$$

and CD_i

$$CD_i = \sqrt{(\frac{I_r(i) - \alpha_i\mu_r)}{\sigma_r})^2 + (\frac{(I_g(i) - \alpha_i\mu_g)}{\sigma_g})^2 + (\frac{(I_b(i) - \alpha_i\mu_b)}{\sigma_b})^2} \tag{4}$$

where (μ_r, μ_g, μ_b) is the mean of the skin color in the training set and $(\sigma_r, \sigma_g, \sigma_b)$ is the standard deviation of this set. A point is classified as skin if the computed CD_i is smaller than a preset threshold and α_i falls in value between two thresholds that are determined automatically based on the desirable detection rate of the skin color in the training sequence (since the skin region is presegmented manually a detection of 100% is used).

The skin region is further consolidated using a standard binary connected component analysis that employs an expand/contract iterative process to construct a region without holes and minimal noise. Figure 3 illustrates the stages of this process.

The candidate region is analyzed to determine if it fits one of the known posture of the hand and fingers. Currently, two postures are represented, an open hand and a grab-ready hand (see Figures 5, 16, 13). The matching of a region to a posture involves (1) computing the convex hull of the region (2) determining the maximum distance points on the hand silhouette from the convex hull (i.e., the deepest concavity distances) (3) hypothesizing finger locations (4) verifying that the back of the hand and finger regions are mostly skin color and that overall spatial properties are plausible (constraints on sizes of the region of the fingers and their distances). Figure 5 illustrates the posture analysis. The convex hull of the open-hand region (left) is

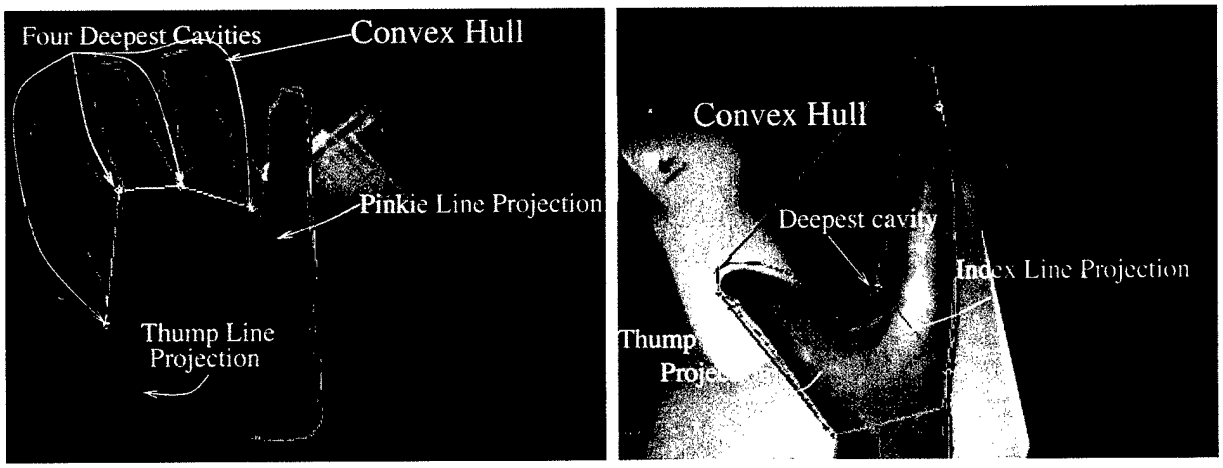


Figure 5: Illustration of hand posture analysis for an open hand and a pre-grabbing posture

computed and the four deepest concavities in the region are chosen as the candidate points for between-finger locations. Then, lines are projected from the right and left-most of these four points and hypotheses for finger and backhand regions are created. The verification step determines whether each of the regions is mostly skin, that the relative sizes are within thresholds (e.g., the pinkie is smaller than the other fingers except the thumb). The picture on the right shows a pre-grab pose of the hand. Here the thumb and the index finger are detected. Only the deepest concavity point is sought and used for the initialization process.

The hand initialization approach has been effective in the environment we have been developing. Note also that since the user is cooperative, he/she can change her hand posture until the system provides a graphic feedback indicating it has been initialized correctly. There are two cases where the labeling of hand and finger regions is not successful, if (1) the hand posture does not conform to one of the known ones, or (2) the background includes large regions with skin-like colors. In these cases, the user continues to guide the system to initialize until its criteria are met. A different type of failure occurs when the system mislabels the regions. When this occurs the user observes that the system was not successful and he/she changes the posture of her hand (by moving it out of the field of view of the camera) to force the system to re-initialize.

2.4 Hand and Finger Tracking

Algorithms for estimating hand and finger motion have been described in [31, 19, 22]. Yang and Ahuja [31] used motion segmentation, color and geometric information to detect the hand palm region in image sequences and build a trajectory graph. Shimada et al. [22] employed a 3D detailed hand and finger model to track the movement between frames. They use an estimation by image synthesis from an initialization that is close to the hand posture, and refine the estimate by incorporating silhouette features and prior probability models. Stenger et al. [24] describe a method that employs an anatomic model of the hand to estimate the joint angles from the projected contours of the hand regions into the image. The results are made more robust by a variation on Kalman filtering. In [10] a deformable model that employs a surface mesh that is constructed by PCA from training examples is used. Wu et al. [28] used a data glove device to develop a model of constraints on finger articulation and then use a Monte Carlo tracking algorithm to estimate angles in the scene. Rehg and Kanade [19] developed a model-based approach for handling occlusions that occur during finger movements. A 3D model of the hand was used to anticipate and account for visibility of finger regions during motion.

We propose a tracking algorithm that estimates rigid and deformable motions of hands and fingers in the 2D image space. The tracking estimates the motion of the hand between two consecutive frames. The hand is divided into the backhand and the fingers. The former is assumed to move as a rigid planar object and therefore can be tracked using a planar or affine parameterized motion model [5]. The latter are tracked using a parameterized motion model that is inspired by a deformable model for tracking eyebrow motion [5].

Tracking proceeds in three stages. In the first the backhand region motion is estimated. In the second stage the estimated backhand motion is used to register the full region of the hand (i.e., backhand and fingers) of the image at time $t+1$ to the hand region in frame t. This cancels the rigid motion of the backhand but leaves out the finger articulation/deformation. The third stage estimates the deformation of each finger between frames from its known region at time t.

Before the proposed model is discussed we review basic concepts related to optical flow. Let $I(x, y, t)$

be the image brightness at a point (x, y) at time t. Since changes between consecutive images are generally small, it is typical to employ the brightness constancy assumption for a point in motion which is given by

$$I(x, y, t) = I(x + u, y + v, t + 1) \tag{5}$$

where (u, v) is the horizontal and vertical image velocity at location (x, y). Applying a Taylor Series approximation (assuming locally constant flow) and dropping terms leads to

$$0 = I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t) \tag{6}$$

where I_x, I_y and I_t are the spatial and temporal derivatives of the intensity image $I(t)$ relative to $I(t + 1)$. Since Equation 6 is underconstrained for the computation of (u, v) at a single pixel, it is usually reposed as a minimization of a least squares error of the flow over a very small neighborhood, R, of (x, y) in which the velocity is constant over R (more complex models are possible as discussed below). In this case we choose (u, v) to minimize

$$E(u, v) = \sum_{(x,y)\in R} (I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t))^2. \tag{7}$$

Instead of the classical least squares minimization in Equation 7 it is preferable to employ a robust estimation approach as proposed in [3], resulting in

$$E(u, v) = \sum_{(x,y)\in R} \rho(I_x(x, y, t)u + I_y(x, y, t)v + I_t(x, y, t), \sigma_e) \tag{8}$$

where ρ is a robust error norm that is a function of a scale parameter σ_e. The computational aspects of the multi-scale model follow, generally, the approach proposed by Black and Anandan [3, 4]. The minimization is carried out using a descent method, Simultaneous Over-Relaxation (SOR). The minimization of $E(u, v)$

with respect to u and v is achieved using an iterative update equation, so at step $n+1$

$$u_{x,y}{}^{(n+1)} = u_{x,y}{}^{(n)} - \omega \frac{1}{T(u_{x,y})} \frac{\partial E}{\partial u_{x,y}} \tag{9}$$

where $0 < \omega < 2$ is an overrelaxation parameter which is used to overcorrect the estimate of $u_{x,y}{}^{(n+1)}$ at stage $n+1$ (a similar treatment is given for v). The value of ω determines the rate of convergence. The term $T(u_{x,y})$ is an upper bound on the second partial derivative of E

$$T(u_{x,y}) \geq \frac{\partial^2 E}{\partial^2 u_{x,y}} \tag{10}$$

To achieve a globally optimal solution the robust error norm ρ is started with a large enough scale parameter σ_e to find a solution using the SOR technique, then iteratively repeating this process while decreasing σ_e and starting with the last estimate. The choice of a large enough σ_e guarantees convexity of the error function at the beginning of the process which is followed by the use of the Graduated Non-Convexity method developed by [6]. The iterated decrease in σ_e reduces the influence of the outlier measurements and thereby refines the estimates.

The robust error norm chosen is the one proposed by Geman-McClure [9] (see Figure 6)

$$\rho(x, \sigma_e) = \frac{x^2}{\sigma_e + x^2} \tag{11}$$

and its derivative

$$\psi(x, \sigma_e) = \frac{2x\sigma_e}{(\sigma_e + x^2)^2} \tag{12}$$

Figures 7 and 8 illustrate the flow results computed at each point of the mouth during a speech image sequence. The images in Figure 7 show frames from a long sequence of mouth motion during speech. The dense flow shown in Figure 8 shows examples for optical flow fields computed from the long image sequence and not just the images shown in Figure 7.

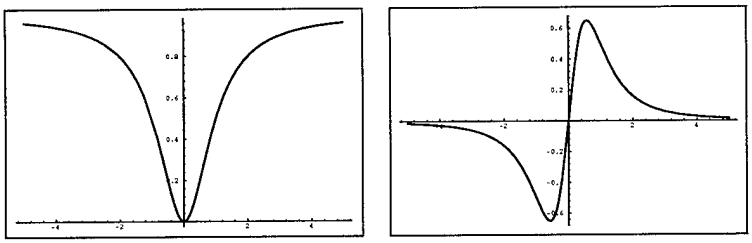


Figure 6: The robust error norm ρ (left) and its derivative (right) taken from Geman-McClure

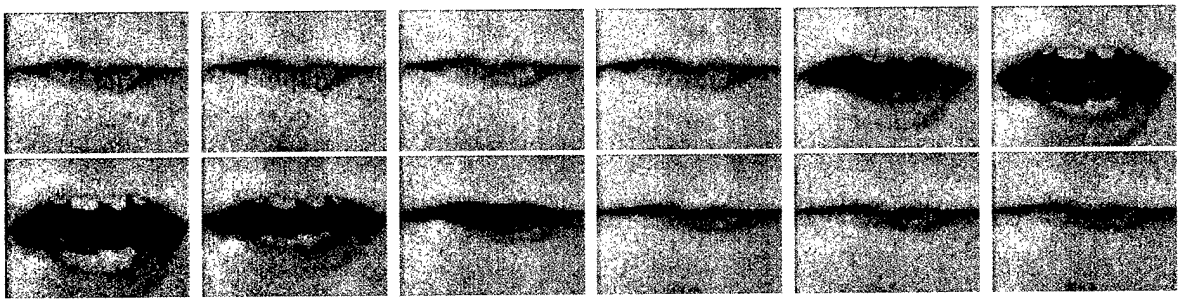


Figure 7: Example frames for one letter in the training set.

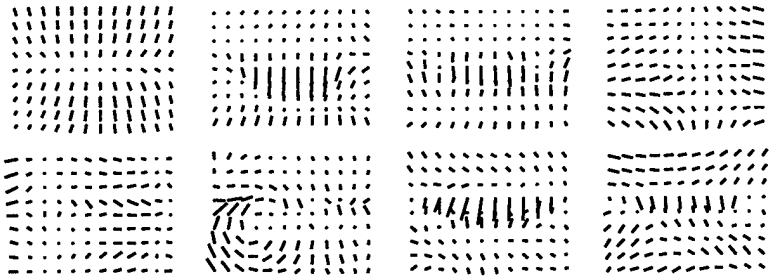


Figure 8: Sample flow computed from an image sequence of mouth motion during speech. The vector at each point shows the magnitude and direction of the (u, v) values computed at the point.

The local constant-flow model can be replaced by a model that integrates information over a larger neighborhood of the image under more general assumptions about the variation of the motion. Parameterized models of image motion make explicit the assumptions about the motion and typically assume that the image flow can be represented by a low-order polynomial [2]. Within small image regions, the following affine model of image motion is often sufficient [14] to represent the motion of a plane under orthographic projection:

$$u(x,y) = a_0 + a_1 x + a_2 y \tag{13}$$

$$v(x,y) = a_3 + a_4 x + a_5 y \tag{14}$$

where the a_i are constants, $(u(x,y), v(x,y))$ are the horizontal and vertical components of the flow at the image point (x,y), and the spatial positions (x,y) are defined with respect to some image point (typically the center of the region).

The parameters a_i have qualitative interpretations in terms of image motion (see Figure 9). For example, a_0 and a_3 represent horizontal and vertical translation respectively. Additionally, we can express *divergence* (isotropic expansion), *curl* (rotation about the viewing direction), and *deformation* (squashing or stretching) as combinations of the a_i [7, 14]:

$$\text{divergence} = a_1 + a_5 = (u_x + v_y), \tag{15}$$

$$\text{curl} = -a_2 + a_4 = -(u_y - v_x), \tag{16}$$

$$\text{deformation} = a_1 - a_5 = (u_x - v_y) \tag{17}$$

where the subscripts x and y indicate partial derivatives of the image velocity. Divergence, curl and the magnitude of the deformation have the convenient property of being invariant to rotations of the image coordinate frame [7].

The motion of the backhand as a planar object under perspective projection is captured by an eight-

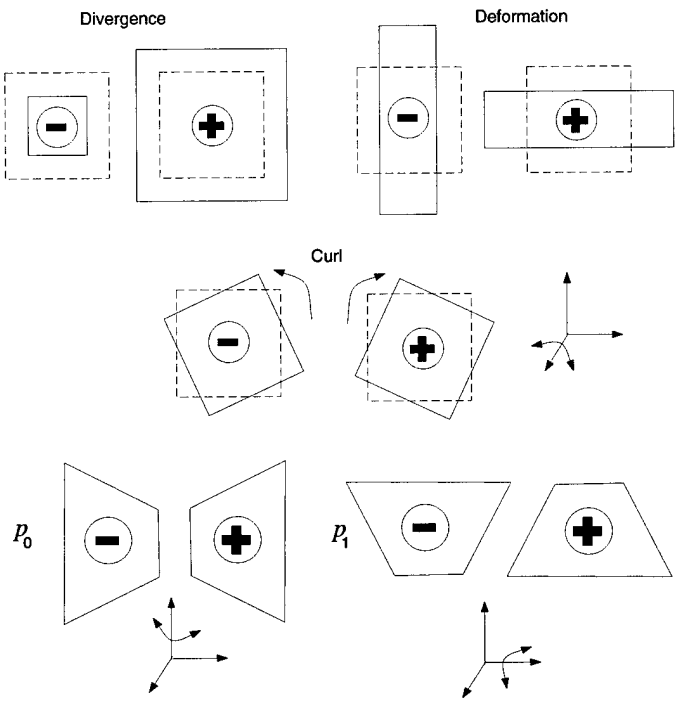


Figure 9: Interpretation of parameters for affine and planar motion.

parameter model [1, 27]:

$$u(x,y) = a_0 + a_1 x + a_2 y + p_0 x^2 + p_1 xy, \tag{18}$$

$$v(x,y) = a_3 + a_4 x + a_5 y + p_0 xy + p_1 y^2. \tag{19}$$

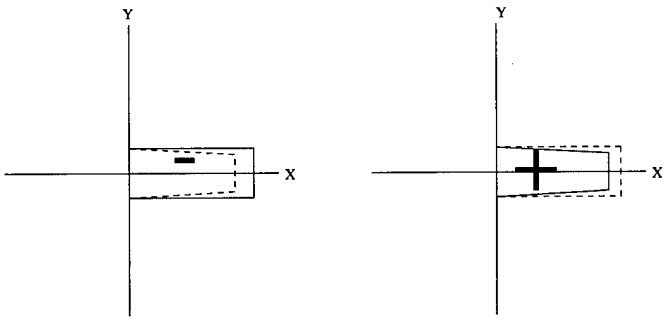
These parameters roughly represent "yaw" and "pitch" deformations in the image plane respectively and are illustrated in Figure 9.

The affine model can be expanded to a general model for finger deformation:

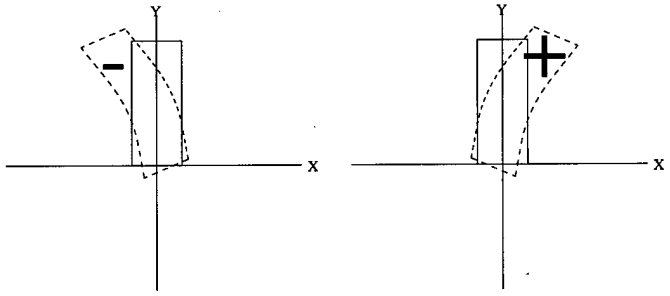
$$u(x,y) = a_0 + a_1 x + a_2 y + a_6 x^2 + a_7 y^2, \tag{20}$$

$$v(x,y) = a_3 + a_4 x + a_5 y + a_8 y^2 + a_9 x^2. \tag{21}$$

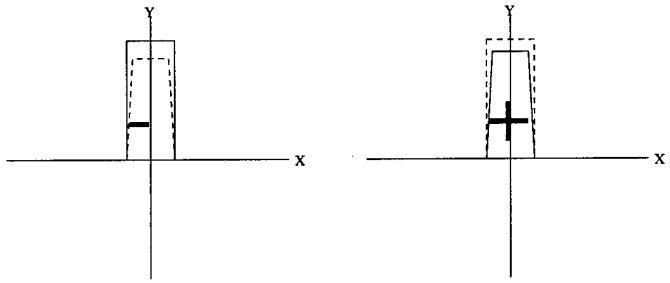
This model combines the affine transformation as well as more specialized transformations shown in Figure



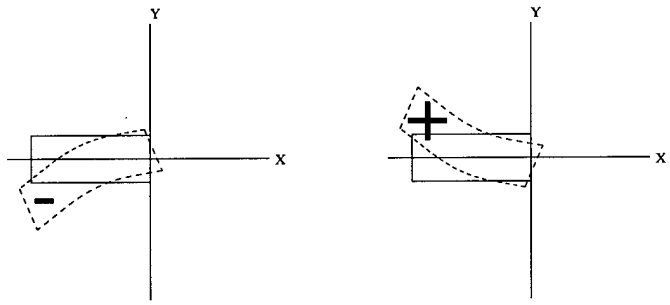
Positive and negative a_6 capture these motions



Positive and negative a_7 capture these motions



Positive and negative a_8 capture these motions



Positive and negative a_9 capture these motions

Figure 10: Interpretation of motion parameters. Solid lines reflect hand region at time t while dashed lines reflect the region at $t+1$.

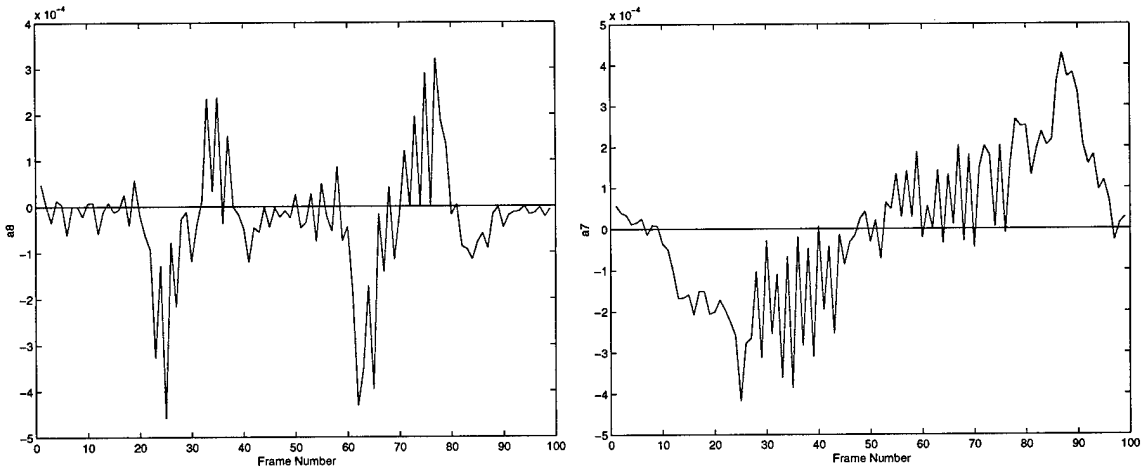


Figure 11: Examples for a_7 and a_8 from a real sequence of finger clicking and grabbing (left and right, respectively).

10. The figure shows the motion that is captured by the quadratic terms a_6, a_7, a_8 and a_9 assuming that the region of the finger is placed at the center of the coordinate system. While this motion model is rich in capturing various finger deformations it can be simplified by taking into account that the gestures we are most interested can be captured by a subset of these parameters, specifically a_7, a_8 and a_9. This motion model approximates the finger region appearance under motions such as clicking and grabbing. The estimation of the models can thus be done on a subset of the 10 parameters. Note that the region's location is not invariant to the coordinate system. In practice, each finger region is defined with respect to a local coordinate system shown in Figure 10 and it is done automatically for each frame.

Figure 11 shows examples of the recovered motion parameters for tracking of finger motions during a click and grab actions. The left graph shows the estimated value of a_7 for a click finger motion repeated two times. As shown in Figure 10. as the finger is depressed the parameter a_7 assumes a negative value and as it is released a positive value is recovered. The repeated gesture shows similar parameter values. The graph on the right shows the value of parameter a_8 during a grab-release sequence. The value is negative for grab closure and positive for grab opening.

2.5 Finger gesture/motion recognition

Gesture hand and finger movement recognition have been approached using Hidden Markov Models [15, 17] and state transition diagrams [13]. Jo et al. [13] proposed a state space diagram that constrains the recognition of user intentions by explicitly marking allowable transitions. Image features obtained from extracted hand regions are used to control state transitions. While other recognition tools are generally available for recognition (e.g., neural networks, Principal Component Analysis, Nearest Neighbor) overall only a few algorithms have been reported for recognition of hand and finger movements.

Recognition of hand and finger gestures and motions is central to our system's interface between the user and the synthesized scene. There are multiple levels of recognition:

- General hand and finger movements with intention to reach a particular location or object in the scene. The system simply reflects the user's hand motion in the scene as the user executes the motions. As a result, it relies on the user to place her hand and fingers in the scene and when she is satisfied she then executes an intended action.

- Single finger movements intended to affect part of the scene in simple ways such as clicking or dragging. The system detects the begin-hold-end of a click action of the fingers by employing a rule-based system that uses the parameter values recovered at each frame in addition to those in a preceding temporal window. The beginning and ending of a clicking finger action are reflected in the synthesized scene by a visual feedback such as a color change of the active area or object.

- Multiple finger movements employing several fingers to accomplish a coordinated task such as grabbing or releasing of an object. In this case, a rule-based system is used to recognize if the simultaneously beginning and ending on the respective finger's actions reflect the intended action.

Action recognition is done directly from the motion parameters computed during tracking the finger movements. As shown in Figure 11, the parameters directly reflect the deformation of the tracked finger region and thus are indicative of the finger motion between frames. In earlier work [5] a rule-based system

was used to interpret facial actions from similar parameters and in another approach [30] a Principal Component Analysis based approach was developed to model and recognize actions in high-dimensional space of parameters (40D space). Here, we employ a simple rule-based recognition system since the number of parameters is small and the model of finger actions is relatively simple. Each parameter is thresholded and turned into a label: POSITIVE ACTIVE or NEGATIVE ACTIVE or NEUTRAL. If five consecutive frames are of the same label then a respectively higher level action is recognized. For example, in Figure 11, at the top, the five consecutive instances of NEGATIVE ACTIVE trigger a BEGIN CLICK, the five consecutive instances of POSITIVE ACTIVE trigger END CLICK and overall they recognize the finger gesture of clicking. Similarly in the graph on the right, the NEGATIVE ACTIVE is followed by POSITIVE ACTIVE for one finger. However, for a grab recognition the thumb must perform the reverse activity simultaneously.

Table 2 describes the conversion of parameter values into the symbols used for recognition of finger actions.

Command category	Example
POSITIVE CLICK	$a_8 > 0.00005$
NEGATIVE CLICK	$a_8 < -0.00005$
NEUTRAL	$-0.00005 < a_8 < 0.00005$
POSITIVE INDEX GRAB	$a_7 < -0.0001$
NEGATIVE INDEX GRAB	$a_7 > 0.0001$
NEUTRAL INDEX GRAB	$-0.0001 < a_7 > 0.0001$
POSITIVE THUMP GRAB	$a_7 < -0.0001$
NEGATIVE THUMP GRAB	$a_7 > 0.0001$
NEUTRAL THUMP GRAB	$-0.0001 < a_7 > 0.0001$

Table 2: The recognition rule for Click and Grip from finger motion parameters

2.6 Graphic rendering

We developed stereo modeling and rendering software that allows us to render graphic scenes on the Sony HMD. It allows the user to tune several parameters (e.g., parallax, eye distance) to control the viewing of the 3D scene. The software employs Open GL to create the scene and texture map image regions (such as user hands) taken from the video camera onto synthetic objects in real-time and render the scene from stereo

vantage points.

2.7 Implementation

Real-time performance is critical since feedback from the system reinforces and corrects gestures that are performed by the user. The algorithms have been implemented to achieve a frame rate of between 20-30HZ. The implementation employs a dual processor PC and an efficient implementation that takes advantage of the computational architecture of the Intel Pentium III processors.

3 Experiments

Figures 16, 13 and 15 show the movement of the hand and fingers as they are tracked and overlayed over a map conveying a clicking and grabbing gestures. As the user moves his hand, the overlayed scene is updated. When the user executes a gesture the scene is changed to reflect the intent of the user. In the case of clicking, the active link changes color and a message is shown on the screen indicating what was selected, while in the case of grabbing and displacing an object the color of the object changes from red to green and it is moved with the hand until it is released. The figures show a 2D view of the scene; we also implemented 3D rendering where stereoscopy is used to enhance the depth effect in the graphic scene.

In Figure 13 the hand is shown grabbing an object (introduced verbally to the map) and moving it to another location then releasing it. The change in the color of the object provides the user a feedback that the object is being held or released as the actions are recognized. Figure 12 show the images captures by the camera.

In Figure 16 the index finger is shown clicking on an image label (69) placed onto an augmented map of the Washington DC Metro area. The top left images show the beginning of the click while the recognition of the click occurs as soon as the finger bends back to its normal position (bottom two images). The label Clicked 69 shows the user that the action has been recognized. In this map the user can select points marked with black square and a number in it.

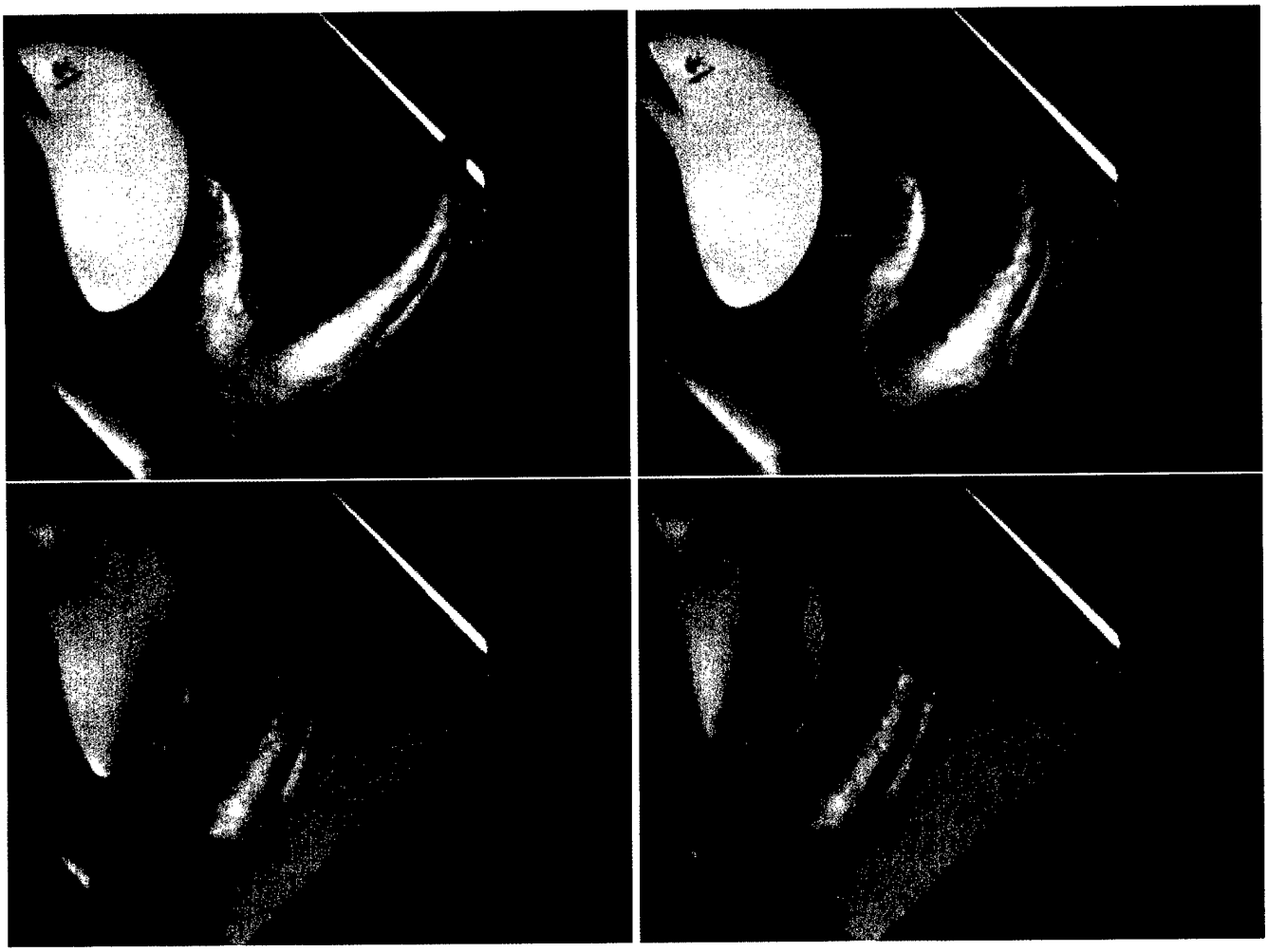


Figure 12: Images of hand movement during a grabbing sequence (top left, right and bottom left and right, respectively

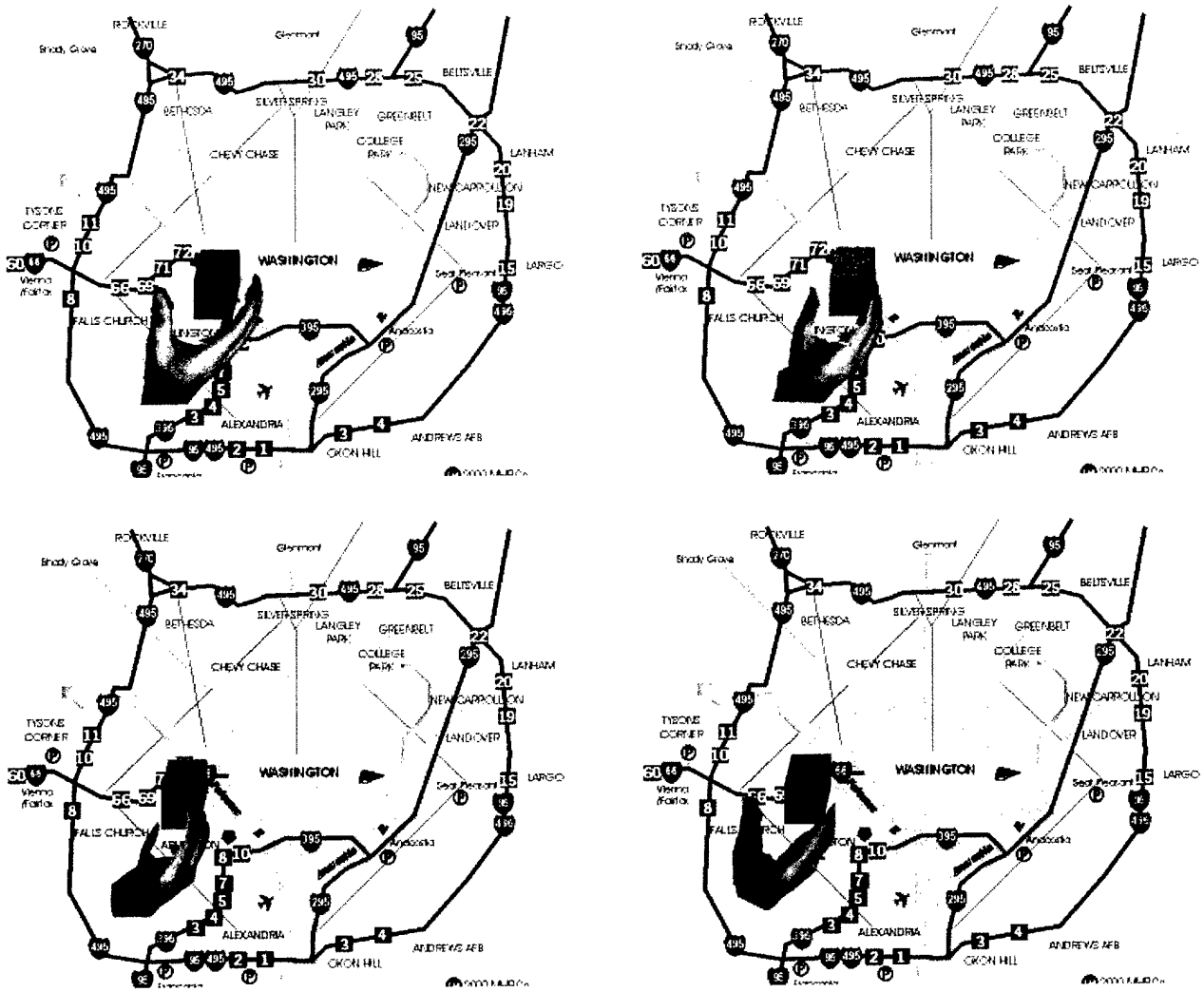


Figure 13: Images of hand tracking and overlay on a map for a object Grip and Move action

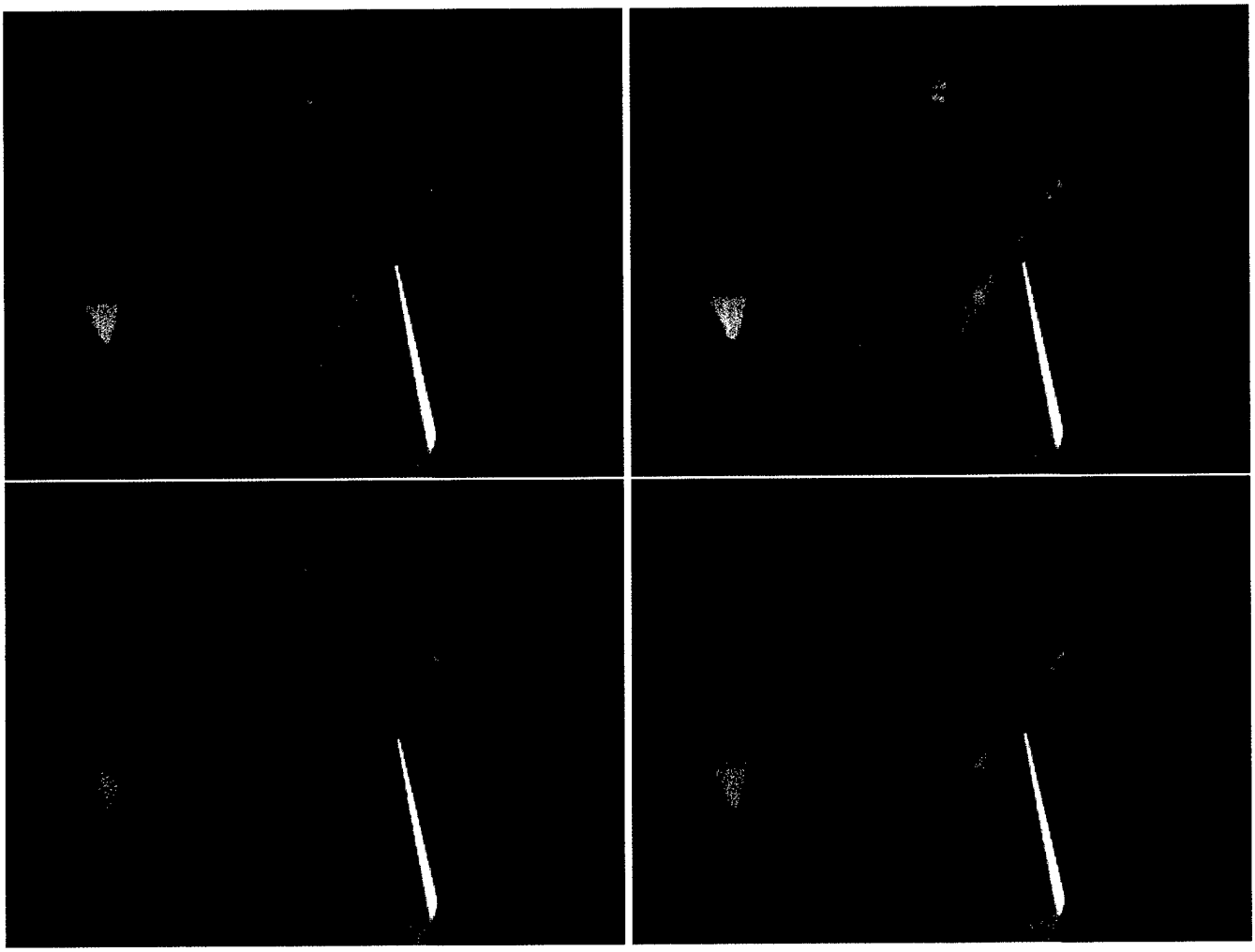


Figure 14: Images of hand movement during a click (top left, right and bottom left and right, respectively

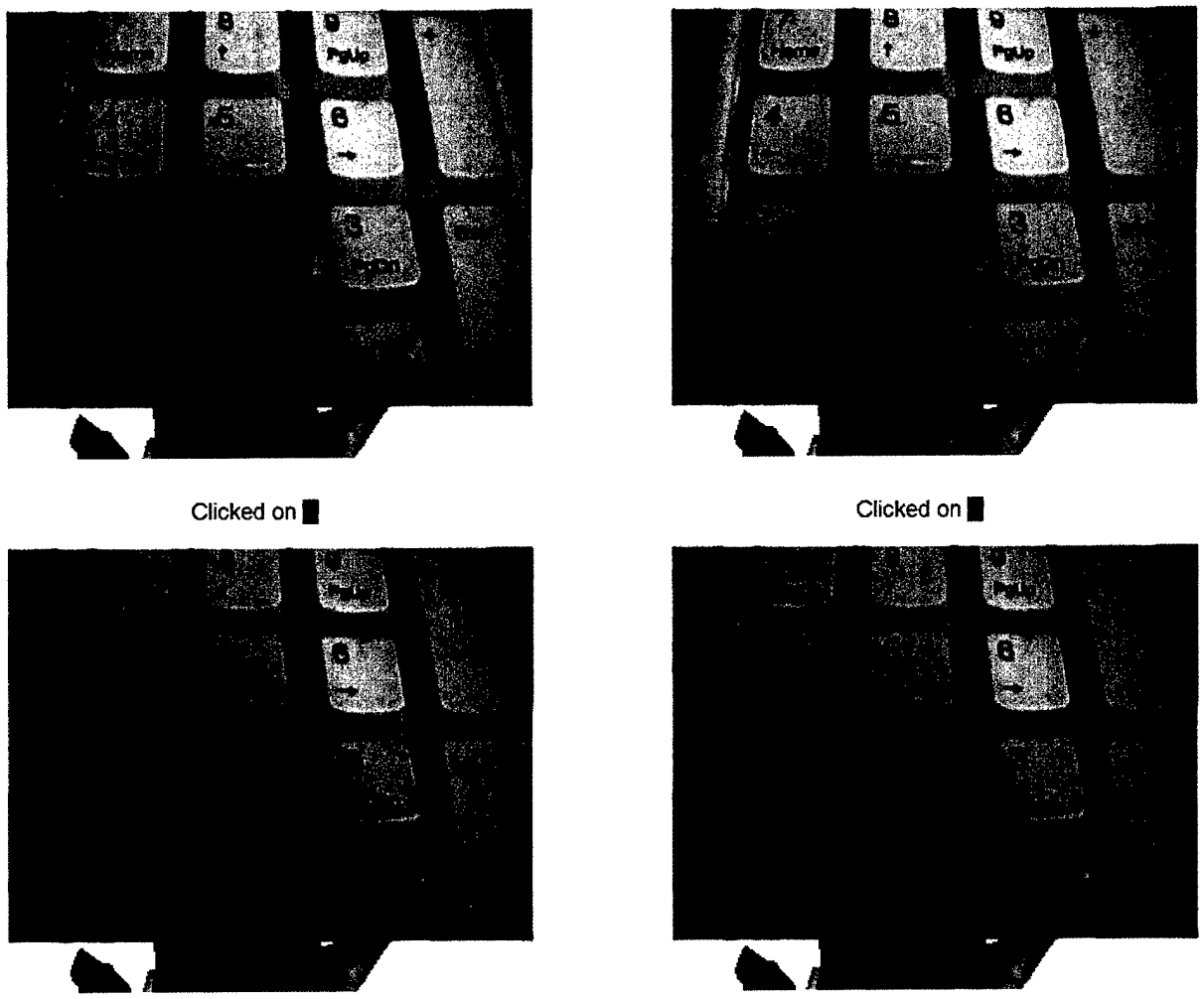


Figure 15: Images of hand tracking and overlay during a clicl action

Finally, Figure 15 shows a clicking on a keypad similar to Figure 16. Here the key 2 has been recognized as the object of the action. Figure 14 show the images captures by the camera.

4 Summary and Future Work

The reported research considered an end-to-end system for multimodal interface to manipulate a virtual scene. Speech recognition was used to alter the scene composition in situations where vision of the user gesture is of limited utility as well as cases in which the change is best affected by speech. Visual analysis focused on a closed-loop feedback to the cooperative user. This reduced the challenge to the vision algorithms and enhanced overall performance. Both fundamental as well as system implementation challenges were encountered and addressed.

We are currently developing an algorithm for hand depth estimation to support accurate rendering of the user's hand within the geometry of the synthetic scene. Computational power should in the near future allow higher frame-rate processing and as a result accuracy of tracking and 3D depth estimation will improve.

Expanding the vision analysis to handle more hand postures and gestures as well as more complex movement scenarios is planned and remains challenging. Richer interactions with an object that involve hand-object occlusions remain a serious obstacle since they require complex 3D analysis in both the graphical and visuals scenes.

References

[1] G. Adiv. Determining three-dimensional motion and structure from optical flow generated by several moving objects. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, PAMI-7(4), 1985, 384-401.

[2] J. R. Bergen, P. Anandan, K. J. Hanna, and R. Hingorani. Hierarchical model-based motion estimation. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, 237-252. Springer-Verlag, May 1992.

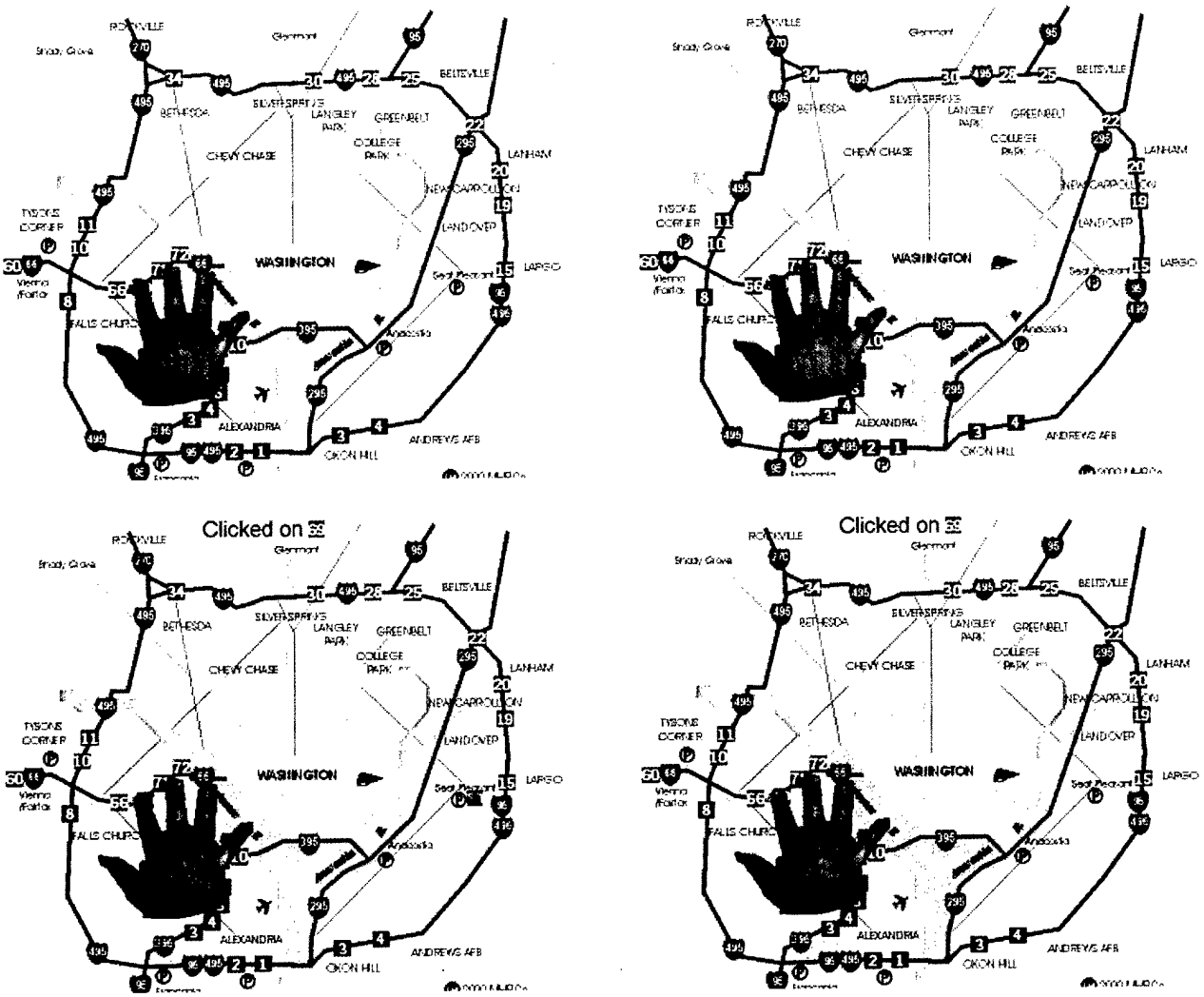


Figure 16: Images of the index finger clicking on an image label (label 69) onto an augmented map of the Washington DC Metro area. The top left images show the beginning of the click while the recognition of the click occurs as soon as the finger bends back to its normal position (bottom two images). The label Clicked 69 (top of the two images) shows the user that the action has been recognized. In this map the user can select points marked with black square and a number in it.

[3] M.J. Black and P. Anandan. *A Frame-work for Robust Estimation of Optical Flow.* ICCV 1993, Berlin, 231-236.

[4] M.J. Black and P. Anandan. *The Robust Estimation of Multiple Motions: Parametric and Piecewise-Smooth Flow Fields.* 1994 Revision of Technical Report P93-00104, Xerox PARC, December 1993.

[5] M. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motions. *IJCV*, 25(1), 1997, 23-48.

[6] A. Blake and A. Zisserman. *Visual Reconstruction* The MIT Press, Cambridge, Massachusetts, 1987.

[7] R. Cipolla and A. Blake. Surface orientation and time to contact from image divergence and deformation. In G. Sandini, editor, *Proc. of Second European Conference on Computer Vision, ECCV-92*, volume 588 of *LNCS-Series*, 187-202. Springer-Verlag, May 1992.

[8] Q. Delamarre, and O. Faugeras, Finding pose of hand in video images: a stereo-based approach. Proceedings of the International Conference on Automatic Face and Gesture Recognition, 1998, 585-590.

[9] S. Geman and D.E. McClure. *Statistical Methods for Tomographic Image Reconstruction.* Bulletin of the International Statistical Institute, LII-4:5-21, 1987.

[10] T. Heap and D. Hogg. Towards 3D Hand Tracking using a Deformable Model. In 2nd International Face and Gesture Recognition, 1996, 140-145.

[11] T. Horprasert, D. Harwood, and L.S. Davis, A Statistical Approach for Real-time Robust Background Subtraction and Shadow Detection, Proc. ICCV FRAME-RATE Workshop, Greece, September 1999.

[12] M.J. Jones and J.M. Rehg, Statistical color models with application to skin detection. Proceedings of the Conference on Computer Visio and Pattern Recognition, 1999, Vol. 1, 274-280.

[13] J. Kang-Hyun, Y. Kuno, and Y. Shirai, Manipulative hand gesture recognition using task knowledge for human computer interaction. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 1998, 468-473.

[14] J. J. Koenderink and A. J. van Doorn. Invariant properties of the motion parallax field due to the movement of rigid bodies relative to an observer. *Optica Acta*, 22(9):773–791, 1975.

[15] H.K. Lee and J.H. Kim, An HMM-based threshold model approach for gesture recognition. IEEE Trans. on Pattern Analysis and Machine Intelligence, (21)10, 1999, 961-973.

[16] M.S. Lee, D. Weinshall, E. Cohn-Solel; A. Colmenarez and D. Lyons. A computer vision system for on-screen item selection by finger pointing. Proceedings of the Conference on Computer Vision and Pattern Recognition, 2001, Volume: 1, 2001. 1026-1033.

[17] S. Marcel, O. Bernier, J.-E. Viallet and D. Collobert, Hand gesture recogntion using input-output hidden Markov models. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 2000, 456-461.

[18] V.I. Pavlovic, R. Sharma and T.S. Huang, Visual interpretation of hand gestures for human-computer interaction: a review. IEEE Trans. on Pattern Analysis and Machine Intelligence, (19)7, 1997, 677-695.

[19] J. Rehg and T. Kanade. Model-Based Tracking of Self-Occluding Articulated Objects ICCV 1995, 612-617.

[20] R. Rosales, V. thitsos, L. Sigal, L. and S. Sclaroff, 3D hand pose reconstruction uspecialized mappings. Proceedings International Conference on Computer Vision, 2001, Volume: 1, 378-385.

[21] Y. Sato, Y. Kobayashi and H. Koike, Fast tracking of hands and fingertips in infrared images for augmented desk interface. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 2000, 462-467.

[22] N. Shimada, Y. Shirai, Y. Kuno, and J. Miura, Hand gesture estimation and model refinement using monocular camera-ambiguity limitation by inequality constraints. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 1998, 268-273.

[23] L. Sigal, S. Sclaroff, and V. Athitsos, Estimation and prediction of evolving color distributions for skin segmentation under varying illumination. Proceedings of the Conference on Computer Vision and Pattern Recognition, 2000, Volume: 2, 152-159.

[24] B. Stenger, P.R.S. Mendonca and R. Cipolla, Model-based 3D tracking of an articulated hand. Proceedings of the Conference on Computer Vision and Pattern Recognition, 2001, Volume: 2, 310-315.

[25] J.-C. Terrillon, M.N. Shirazi, H. Fukamachi and S. Akamatsu, Comparative performance of different skin chrominance models and chrominance spaces for the automation of human faces in color images. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 2000, 54-61.

[26] J. Triesch, J. and C. von der Malsburg, A system for person-independent hand posture recognition against complex backgrounds. IEEE Trans. on Pattern Analysis and Machine Intelligence, (23)12, 2001, 1449-1453.

[27] A. M. Waxman, B. Kamgar-Parsi, and M. Subbarao. Close-form solutions to image flow equations. In *Proc. Int. Conf. on Computer Vision, ICCV-87,* pages 12–24, London, England, June 1987.

[28] Y. Wu; J.Y. Lin and T.S. Huang, Capturing natural hand articulation. Proceedings International Conference on Computer Vision, 2001, Volume: 2, 426-432.

[29] A. Wu, M. Shah and N. Da Vitoria Lobo, A virtual 3D blackboard: 3D finger tracking using a single camera. Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 2000, 536-543

[30] Y. Yacoob and M.J. Black, Parameterized Modeling and Recognition of Activities. *Journal of Computer Vision and Image Understanding,* 73(2), 1999, 232-247.

[31] M. Yang and N. Ahuja. Extracting gestural motion trajectories.

[32] X. Zhu, J. Yang and A. Waibel, Segmenting hands of arbitrary color Proceedings of the International Conference on Automatic Face and Gesture Recogntion, 2000, 446-453.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. AGENCY USE ONLY (*Leave Blank*)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	10/1/02	Final 12/1/99-11/30/02

4. TITLE AND SUBTITLE
Gesture-Based Control of Spaces and Objects in Augmented Reality

5. FUNDING NUMBERS
N000140010061

6. AUTHORS
Yaser Yacoob and Larry Davis

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
UMIACS
University of Maryland, CP
College Park, MD 20742

8. PERFORMING ORGANIZATION REPORT NUMBER
n/a

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)
Dr. Behzard Kamgar-Parsi
ONR
800 N. Quincy St, Arlington, VA 22217

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
n/a

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Public availability

12b. DISTRIBUTION CODE

13. ABSTRACT *(Maximum 200 words)*

A multi-modal system integrating computer vision and speech recognition to enable interaction with virtual spaces/objects by natural gestures and speech is described. Computer vision algorithms are employed to measure and interpret hand/finger movement of the user. Our research focuses on detection, tracking, recognition and visual feedback of the hand and finger movements in a cooperative user environmnet and the integration of gesture and speech recognition for man/machine communication.

14. SUBJECT TERMS
augmented reality, gesture analysis, finger tracking, gesture recognition

15. NUMBER OF PAGES
30

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
unclassified	unclassified	unclassified	unlimited