Speech/Gesture Interface to a Visual Computing Environment for Molecular Biologists

Rajeev Sharma, Thomas S. Huang, Vladimir I. Pavlović, Yunxin Zhao, Zion Lo, Stephen Chu,
Klaus Schulten, Andrew Dalke, Jim Phillips, Michael Zeller, William Humphrey
The Beckman Institute, University of Illinois at Urbana-Champaign,
405 N. Mathews Avenue, Urbana, IL 61801

Abstract

Recent progress in 3-D, immersive display and virtual reality (VR) technologies has made possible many exciting applications, for example, interactive visualization of complex scientific data. To fully exploit this potential there is a need for "natural" interfaces that allow the manipulation of such displays without cumbersome attachments. In this paper we describe the use of visual hand gesture analysis and speech recognition for developing a speech/gesture interface for controlling a 3-D display. The interface enhances an existing application, VMD, which is a VR visual computing environment for molecular biologists. The free hand gestures are used for manipulating the 3-D graphical display together with a set of speech commands. We describe the visual gesture analysis and the speech analysis techniques used in developing this interface. The dual modality of speech/gesture is found to greatly aid the interaction capability.

1 Introduction

Although there has been tremendous progress in recent years in 3-D, immersive display and virtual reality (VR) technologies, the corresponding human-computer interaction (HCI) technologies have lagged behind. For example, current interfaces involve the use of heavy headsets, datagloves, tethers, and other VR devices which may deter or distract the user of the VR facility. To fully exploit the potential that VR offers as a means of visualizing and interacting with complex information, it is important to develop "natural" means of interacting with the virtual display.

The communication mode that seems most relevant to the manipulation of physical objects is hand motion, also called *hand gestures*. We use it to act on the world, to grasp and explore objects, and to express our ideas. Now virtual objects, unlike physical objects, are under computer control. To manipulate them naturally, humans would prefer to employ hand gestures as well as speech. Psychological experiments, for example, indicate that people prefer to use speech in combination with gestures in a virtual environment, since it allows the user to interact without special training or special apparatus and allows the user to concentrate more on the virtual objects and the tasks at hand [1]. We explore this multimodal nature of HCI involved in manipulating virtual objects using speech and gesture.

To keep the interaction natural, it is desirable to have as few devices attached to the user as possible. Motivated by this, we have been developing techniques that will enable spoken words and simple free-hand gestures to be used while interacting with 3D graphical objects in a virtual environment. The voice commands are monitored through a microphone and recognized using automatic speech recognition (ASR) techniques. The hand gestures are detected through a pair of strategically positioned cameras and interpreted using a set of computer vision techniques that we term automatic gesture recognition (AGR). These computer vision algorithms are able to extract the user hand from the background, extract positions of the fingers, and distinguish a meaningful gesture from unintentional hand movements using the context. We use the context of a particular virtual environment to place the necessary constraints to make the analysis robust and to develop a command language that attempts to optimally combine speech and gesture inputs.

2 A Virtual Environment Testbed

The particular virtual environment that we consider has been built for structural biologists by the Theoretical Biophysics Group at the University of Illinois at Urbana-Champaign. The system, called MDScope, provides an environment for simulation and visualization of biomolecular systems in structural biology; its graphic front-end is called VMD [2]. A 3-D projection system permits multiple users to visualize and interactively manipulate complex molecular structures (Figure 1). This helps in the process of developing an understanding of important properties of the molecules, in viewing simulations of molecular dynamics, and in "playing" with different combinations of molecular structures. One potential benefit of the system is reducing the time to discover new compounds, in research toward new drugs for example.

The older version of this system uses a keyboard and a magnetically tracked pointer as the interface. This is particularly inconvenient since the system is typically used by multiple (6-8) users, and the interface hinders the interactive nature of the visualization system. Thus incorporating voice command control in MDScope would enable the users to be free of keyboards and to interact with the environment in a natural manner. The hand gestures would permit the users to easily manipulate the displayed model and "play"

The financial support of the National Science Foundation (Grant IRI-89-08255,BIR-9318159, IRI-95-02074), Sumitomo Electric Industries, the National Institutes of Health (PHS 5 P41 RR05969-04), the Department of Energy Computational Science Graduate Fellowship Program, and the Roy J. Carver Charitable Trust is gratefully acknowledged.

1015-4651/96 $5.00 © 1996 IEEE
Proceedings of ICPR '96

Figure 1: *A 3D visualization facility for structural biologist; here researchers are seen discussing the structure of a protein-DNA complex.*

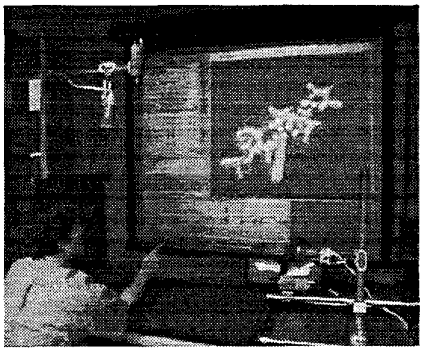

Figure 2: *The experimental setup with with two cameras used for gesture recognition.*

with different spatial combinations of the molecular structures. The integration of speech and hand gestures as a multi-modal interaction mechanism would be more powerful than using either mode alone, motivating the development of the speech/gesture interface. Further, the goal was to minimize the modifications needed to the existing VMD program for incorporating the new interface. The experimental prototypes that we built for both the speech (ASR) and hand gesture analysis (AGR) required the following addition to the VMD environment.

Software. In order to reduce the complexity and increase the flexibility of the program design, a communications layer was added so external programs can be written and maintained independently from the VMD code. These use the VMD text language to query VMD for information or to send new commands. The VMD text language is based on the TCL scripting language. Since all the capabilities of VMD are available at the script level, an external program can control VMD in any way. Both the ASR and AGR programs interact with VMD using this method. For a simple voice command, such as "rotate left 90", the ASR converts the phrase into the VMD text command "rotate y 90" and sends that to VMD. Similarly, when the AGR is being used as a pointing device, it sends the commands to change the current position and vector of VMD's graphical 3D pointers.

Setup for visual gesture analysis. To facilitate the development of AGR algorithms, we designed an experimental platform shown in Figure 2 that was used for gesture recognition experiments. In addition to the uniformly black background, there is a lighting arrangement that shines red light on the hand without distracting the user from the main 3D display. The setup has the additional advantage that it can be transported easily and is relatively unobtrusive.

Setup for speech analysis. A prototype ASR system has been implemented and integrated into VMD. The system consisted of two blocks: a recorder front-end followed by the recognizer unit. The recorder employed a circularly-buffered memory to implement its recording duties, sending its output to the recognizer unit in blocks. A digital volume meter accompanied this to provide feedback to the user by indicating an acceptable range of loudness. The recognizer that followed was developed by modifying HTK software. This unit performed feature extraction and time-synchronous Viterbi decoding on the input blocks, sending the decoded speech directly via Tcl-dp commands to an SGI Onyx workstation where the VMD process resided.

Speech/gesture command language. In order to effectively utilize the information input from the user in the form of spoken words and simple hand gestures, we have designed a command language for MDScope that combines speech with gesture. This command language uses the basic syntax of $< action >< object >< modifier >$. The $< action >$ component is spoken (e.g., "rotate") while the $< object >$ and $< modifier >$ are specified by a combination of speech and gesture. An example is, speaking "this" while pointing, followed by a modifier to clarify what is being pointed to, such as "molecule", "helix", "atom", etc., followed by speaking "done" after moving the hand according to the desired motion. Another example of the desired speech/gesture capability is the voice command "engage" to query VMD for the molecule that is nearest to the tip of the pointer and to make the molecule blink to indicate that it was selected and to save a reference to that molecule for future use. Once engaged, the voice command "rotate" converts the gesture commands into rotations of the chosen molecule, and the command "translate" converts them into translations. When finished, the command "release" deselects the molecule and allows the user to manipulate another molecule. The ASR and AGR techniques that made the above interaction possible are described next.

3 Speech input using ASR

3.1 Background

In the integration of speech and gesture within the MDScope environment, a real-time decoding of the user's commands is required in order to keep pace with the hand gestures. Thus there is a need for "word spotting" which is defined as the task of detecting a given vocabulary of words embedded in unconstrained continuous speech. It differs from conventional large-vocabulary continuous speech recognition (CSR or LVCSR) systems in that the latter seeks to determine an optimal sequence of words from a prescribed vocabulary. A direct mapping between spoken utterances and the recognizer's vocabulary is implied with a CSR, leaving no room for the accommodation of non-vocabulary words in the form of extraneous speech or unintended background noise. The basis for word spotting, also termed keyword spotting (KWS), is dictated by real world applications. Real users of a spoken language system

often embellish their commands with supporting phrases and sometimes even issue conversation absent of valid commands. In response to such natural language dialogue and the implications to robust human-computer interaction, standard CSR systems were converted into spotters by simply adding filler or garbage models to their vocabulary. Recognition output stream would then consist of a sequence of keywords and fillers constrained by a simple syntactical network. In other words, recognizers operated in a "spotter" mode. While early techniques emphasized a template-based dynamic time warping (DTW) slant, current approaches are typically armed with the statistical clout of hidden Markov models (HMMs) [4, 5, 8], and recently with the discriminatory abilities of neural networks (NN). These were typically word-based and used an overall network which placed the keyword models in parallel with the garbage models.

3.2 Prototype and Results

Keywords. Table 1 lists the keywords and their phonetic transcriptions chosen for the experiment. These commands

Keyword	Transcription
translate	t-r-ae-n-s-l-ey-t
rotate	r-ow-t-ey-t
engage	eh-n-g-ey-jh
release	r-ih-l-iy-s
pick	p-ih-k

Table 1: *Keywords.*

allowed the VMD user to manipulate the molecules and polymeric structures selected by hand gestures. In modeling the acoustics of the speech, the HMM system was based on phones rather than words for large vocabulary flexibility in the given biophysical environment. A word-based system, though invariably easier to implement, would be inconvenient to retrain if and when the vocabulary changed.

Fillers. Filler models are more varied. In LVCSR applications, these fillers may be represented explicitly by the non-keyword portion of the vocabulary, as whole words for example. In other tasks, non-keywords are built by a parallel combination of either keyword "pieces" or phonemes whether they be context-independent (CI) monophones or context-dependent (CD) triphones or diphones [5].

Twelve fillers or garbage models were used to model extraneous speech in our experiment. Instead of being monophones or states of keyword models as used in prior experiments in the literature, the models that were used covered broad classes of basic sounds found in American English. These are listed in Table 2. Such models provide good coverage of the English language and are amenable to training. There are several things to note. First, the class of "consonants-africates" was not used due to the brevity of occurrence in both the prescribed vocabulary and training data. As observed by [8] and many other researchers, varying or increasing the number of models does not gain much in spotting performance. Second, a model for background silence was included in addition to the twelve garbage models listed. Such a model removed the need for an explicit endpoint detector by modeling the interword pauses in the incoming signal. Note also that the descriptors for the vowel class correspond to the position of the tongue hump in producing the vowel.

Sound class	Symbol
vowels-front	vf
vowels-mid	vm
vowels-back	vb
dipthongs	dipth
semivowels-liquids	svl
semivowels-glides	svg
consonants-nasals	cn
consonants-stops-voiced	csv
consonants-stops-unvoiced	csu
consonants-fricatives-voiced	cfv
consonants-fricatives-unvoiced	cfu
consonants-whispers	cw

Table 2: *Broad sound classes used as garbage models.*

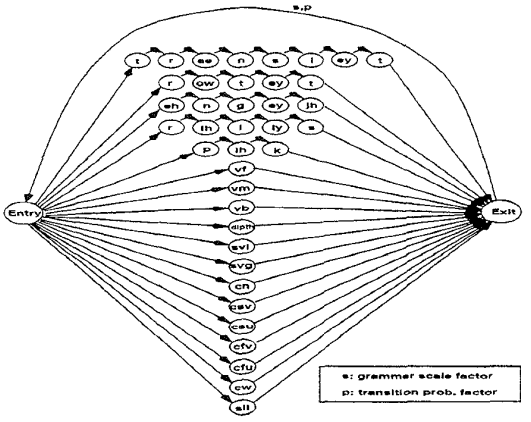

Figure 3: *The recognition network.*

Recognition Network. The recognition syntactical network, as shown in Figure 3, placed the keywords in parallel to a set of garbage models which included a model for silence. These models followed a null grammar, meaning that every model may precede or succeed any other model. As indicated in the figure, a global grammar scale factor (s) and transition probability factor (p) were used to optimize the recognition accuracy and adjust the operating point of the system.

Features and Training. After sampling speech at 16kHz and filtering to prevent aliasing, the speech samples were preemphasized with a first order digital filter using a preemphasis factor of 0.97 and blocked into frames of 25 ms in length with a shift between frames of 10 ms. Each frame of speech was weighted by a Hamming window, and then mel-frequency cepstral coefficients of sixteenth order were derived and weighted by a liftering factor of 22. Cepstral coefficients were chosen as they have been shown to be more robust and discriminative than linear predictive coding coefficients or log area ratio coefficients. Normalized log energy and first order temporal regression coefficients were also included in the feature vector.

The topology of the HMMs for both keyword-phones and garbage models consisted of five states, the three internal states being emitting states. Following a left-to-right traversal, each state was described by a mixture of five continuous density Gaussians with diagonal covariance matrices. Three iterations of the Baum-Welch reestimation procedure were

used in training.

In training the set of fifteen keyword-phones, forty sentences were developed as follows. Each of the five keywords were individually paired with the remaining four. This was then doubled to provide a sufficient number of training tokens. In sum, the sentences were composed of pairs of keywords such as "engage translate" and "rotate pick" which were arranged in such an order as to allow each of the keywords to be spoken sixteen times. Each VMD user proceeded with this short recording session.

In training the garbage models, a much more extensive database of training sentences was required to provide an adequate amount of training data since the twelve broad classes cover nearly the entire spectrum of the standard 48 phones. The TIMIT database was subsequently employed to provide an initial set of bootstrapped models. Retraining was then performed once for one VMD user who had recorded a set of 720 sentences of commonly used VMD commands. These sentences spanned the scope of the VMD diction, including a more detailed set of commands, numbers, and modifiers. This was necessary to provide data normalized to the existing computational environment. Note that the garbage models were trained only once for this experiment. Hence, VMD users only needed to go through the short training procedure detailed above.

Performance. Upon testing the system as a whole with fifty test sentences that embedded the keywords within bodies of non-keywords, wordspotting accuracies ranged to 98% on the trained speaker. The trained speaker refers to each user who trained the keyword-phones regardless of the one who trained the garbage models. This was considered very well by the VMD users for the given biophysics environment, supporting the techniques that were used. In general, false alarms occurred only for those situations where the user embedded a valid keyword within another word. For example, if one says 'translation' instead of 'translate', the spotter will still recognize the command as 'translate'.

3.3 Discussion

In the experiment, a standard CSR was converted to a wordspotter by operating in "spotter mode". This was chosen for its efficiency and experimentally proven success. Had a pure wordspotter been constructed, two passes would be required for the spotting and complete decoding task stretching the envelope of real-time constraints. Also, there is often a tradeoff between the resolution of the models and speed of execution in model-based systems. Furthermore for a wordspotting system, one must balance the false alarm rate against miss rate. In the former, the garbage models provide too much resolution and the opposite occurs in the latter. In wordspotting, one may use transition weights into the individual models to position the operating point at the desired location. As noted earlier, global parameters were used for adjustment; however, later tests will employ a more defined set of weights to control the operating point. In addition to weighting the transitions into the keyword-phone and garbage models, research will be focused on providing a more efficient training procedure and more discriminative models for the keyword-phones. As noted, the confusion that exists when a valid keyword is embedded within a non-keyword must be handled.

4 Hand gesture input using AGR

The general AGR problem is hard, because it involves analyzing the human hand which has a very high degree of freedom and because the use of the hand gesture is not so well understood (See [3] for a survey on vision-based AGR). However, we use the context of the particular virtual environment to develop an appropriate set of gestural "commands". The gesture recognition is done by analyzing the sequence of images from a pair of cameras positioned such that they facilitate robust analysis of the hand images. The background is set to be uniformly black to further help with the real-time analysis without using any specialized image-processing hardware.

Finger as a 3D pointer. The AGR system consists of two levels of subsystems (See Figure 4). First level subsystems are used to extract a 2D pointing direction from single camera images. The second level subsystem combines the information obtained from the outputs of the first level subsystems into a 3D pointing direction. To obtain the 2D pointing direction, the first level subsystems per-

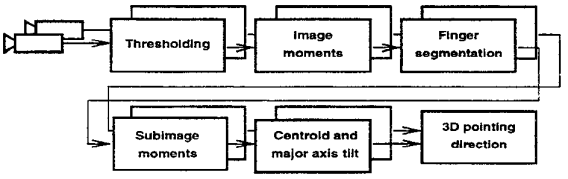

Figure 4: *An overview of the AGR system.*

form a sequence of operations on the input image data. The gray-level image is first thresholded in order to extract a silhouette of the user's lower arm from the background. Next, first and second image moments are calculated and then used to form a bounding box for extraction of the index finger. Once the finger is segmented from the hand, another set of image moments is calculated, this time for the finger itself. Finally, based on these moments, 2D finger centroid and finger direction are found. 3D pointing direction is finally determined in the second level subsystem using the knowledge of the setup geometry and 2D centroids and pointing directions. This information is then forwarded to the central display manager which displays a cursor at an appropriate screen position. Our implementation produced a tracking rate of about 4 frames per second, mainly limited by the inability of the digitization hardware to properly handle multiple video signals. Special purpose hardware can easily improve the performance. However, even with the low sampling rate, the users can achieve a reasonable control of the display.

Gestures for manipulating 3-D display. In addition to recognizing a pointing finger, we have developed a hidden Markov model based AGR system for recognizing a basic set of manipulative hand gestures. Figure 5 gives examples of some of the gestures that were used. We have also developed a gesture command language for MDScope that is mainly concerned with manipulating and controlling the display of the molecular structures. The gesture commands are categorized as being either dynamic (e.g., move back, move forward) or static (e.g., grab, release, stop, up, down).

The system uses image geometry parameters as the features that describe any particular hand posture (static hand image). We use the *radon transform* of an image to extract

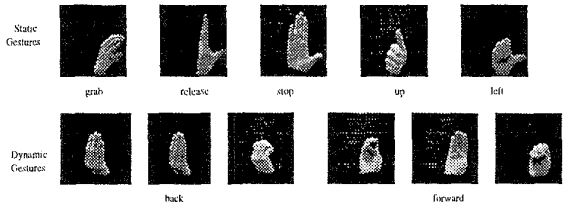

Figure 5: *Examples hand gestures used in manipulating a virtual object and interpreted using AGR.*

these features. The radon transform of the image $I(x,y)$ is defined as follows:

$$R(\theta,t) = \int_{(x,y)} I(t\cos\theta - s\sin\theta, t\sin\theta + s\cos\theta)ds, \quad (1)$$

where $0 \leq \theta \leq \pi/2$. The image geometry moment of the order k is then given by:

$$m^{(k)}(\theta) = \int_t t^k R_0(\theta,t)dt, \quad (2)$$

where R_0 denotes the radon transform normalized with respect to the image "mass":

$$R_0(\theta,t) = \frac{R(\theta,t)}{\int_v R(\theta,v)dv}. \quad (3)$$

The first order moments are what is known as the center of mass of an image. The higher order moments provide additional information on image shape [7]. The set of gestures that were used consisted of both static and dynamic gestures (see Figure 5). The recognition system was built by training hidden Markov models for the specific gestures on example runs. Each gesture in the vocabulary was modeled as a single four-state-HMM. The observations were modeled by a Gaussian mixture of two different sizes (one and three) with a diagonal covariance matrix. Although only 35 training sequences were used the performance of the recognition system was quite good (80% correct recognition rate). The performance is expected to improve with a better model of the hand and by exploiting multimodality as discussed next.

5 Discussion

The speech/gesture interface reported so far could be part of a more general multimodal framework, where other modalities can also be exploited to make the interface more natural and efficient. For example, consider the problem of visual gesture analysis, the following interactions can be exploited to improve the gesture recognition process in the multimodal framework: (a) Interaction of gesture and speech, (b) Interaction of gesture and the virtual scene, (c) Interaction of gesture and gaze direction, and (d) Interaction of gesture and graphical display. These interactions are discussed in more detail in [6] and form a basis of future work.

The experimental results for the gesture recognition show that even with simple image moments the HMM based approach yields a reasonable performance. However, model-based approach could significantly effect the recognition performance. For example, there is a trade-off between the reliability and speed of gesture recognition for different levels of the hand-model used [3]. One approach for hand motion analysis for AGR is to consider the class of motion called articulated motion for analysis and tracking of the hand. Using the prediction based upon articulated motion analysis, we can reliably derive a minimal description of the hand image in real-time. The more detailed the hand model used, the better the prediction that can be made of the hand positions under different gestures. Such models can be used to develop a suitable "feature" vector that can be used in the gesture classification and recognition. The aim would be to replace the simple image moments used in our current implementation with a feature vector that can define a more complicated set of hand gestures needed for manipulating a virtual environment.

6 Conclusions

This paper describes an application where computer vision and speech recognition techniques are used for building a natural human-computer interface for a VR environment, using spoken words and free hand gestures. A VR setup used by structural biologists is considered as a test-bed for developing the multimodal interface and help in defining the gesture recognition (AGR) and speech recognition (ASR) problems. A prototype speech/gesture interface is presented that lets the scientist easily and naturally explore the displayed information. The speech/gesture interface offers a level of interactive visualization that was not possible before. Incorporating voice command control in MDScope enables the users to be free of keyboards and to interact with the environment in a natural manner. The hand gestures permit the users to easily manipulate the displayed model and "play" with different spatial combinations of the molecular structures. The integration of speech and hand gestures as a multi-modal interaction mechanism proves to be more powerful than using either mode alone.

References

[1] A. G. Hauptmann and P. McAvinney. Gesture with speech for graphics manipulation. *Int. Jrn. Man-Machine Studies*, 38(2):231–249, Feb. 1993.

[2] W. F. Humphrey, A. Dalke, and K. Schulten. VMD – visual molecular dynamics. *Journal of Molecular Graphics*, 1995.

[3] V. I. Pavlović, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. Technical Report UIUC-BI-AI-RCV-95-10, Univ. Illinois, Dec 1995.

[4] J.R. Rohlicek, W. Russell, S. Roukos, and H. Gish. Continuous hidden markov modeling for speaker-independent word spotting. In *Proc. ICASSP*, pages 627–630, 1989.

[5] R.C. Rose and D.B. Paul. A hidden markov model based keyword recognition system. In *Proc. ICASSP*, pages 129–132, 1990.

[6] R. Sharma, T. S. Huang, and V. I. Pavlović. A multimodal framework for interacting with virtual environments. In C. A. Ntuen, E. H. Park, and J. H. Kim, editors, *Human Interaction with Complex Systems*. Kluwer Academic Publishers, 1996.

[7] M. Teague. Image analysis via the general theory of moments. *Jrn. Optical Society of America*, 70:920–930, 1980.

[8] J.G. Wilpon, L.R. Rabiner, C. H. Lee, and E. R. Goldman. Automatic recognition of keywords in unconstrained speech using hidden markov models. *IEEE Trans. ASSP*, 38(11):1870–1878, 1990.