A Visual Computing Environment for Very Large Scale Biomolecular Modeling

M. Zeller, J.C. Phillips, A. Dalke, W. Humphrey, K. Schulten
R. Sharma†, T.S. Huang, V.I. Pavlović, Y. Zhao, Z. Lo, S. Chu

Beckman Institute for Advanced Science and Technology
University of Illinois at Urbana-Champaign
405 N. Mathews Avenue, Urbana, IL 61801

†Department of Computer Science and Engineering
Pennsylvania State University
University Park, PA 16802-6106

Abstract

Knowledge of the complex molecular structures of living cells is being accumulated at a tremendous rate. Key technologies enabling this success have been high performance computing and powerful molecular graphics applications, but the technology is beginning to seriously lag behind challenges posed by the size and number of new structures and by the emerging opportunities in drug design and genetic engineering. A visual computing environment is being developed which permits interactive modeling of biopolymers by linking a 3D molecular graphics program with an efficient molecular dynamics simulation program executed on remote high-performance parallel computers. The system will be ideally suited for distributed computing environments, by utilizing both local 3D graphics facilities and the peak capacity of high-performance computers for the purpose of interactive biomolecular modeling. To create an interactive 3D environment three input methods will be explored: (1) a six degree of freedom "mouse" for controlling the space shared by the model and the user; (2) voice commands monitored through a microphone and recognized by a speech recognition interface; (3) hand gestures, detected through cameras and interpreted using computer vision techniques.

Controlling 3D graphics connected to real time simulations and the use of voice with suitable language semantics, as well as hand gestures, promise great benefits for many types of problem solving environments. Our focus on structural biology takes advantage of existing sophisticated software, provides concrete objectives, defines a well-posed domain of tasks and offers a well-developed vocabulary for spoken communication.

1: Introduction

The biomedical sciences are presently undergoing a revolutionary development following the determination of a rapidly increasing number of complex molecular structures of the living cell. Structures encompassing many thousands of atoms, with integral biological functions and of great medical significance, are being discovered at an increasing pace. This development came about only with the advent of high performance computers, powerful molecular graphics software, and a host of program packages regularly used by thousands of researchers. Computational methods play an essential role in structure determination and structure refinement, in graphical interpretations of complex structures, in modeling of biopolymers for drug design and genetic engineering of proteins, and in the physical, mechanistic and dynamic analysis of resolved structures with the goal for understanding the principles underlying biopolymer architecture and function.

1063-6862/97 $10.00 © 1997 IEEE

Computational technology, unfortunately, lags now behind the rapid progress of structural biology and cannot adequately handle the size and number of structures discovered today. Furthermore, current user interfaces suffer from numerous shortcomings, e.g., a poor adaptation to the three-dimensional character of biopolymer structures and a complexity that requires long training periods. Also, gross limitations in locally available computer power restricts the interactive modeling of biopolymers and source codes are, with rare exceptions, inaccessible or poorly documented. Finally, the present paradigm for the rendering of biomolecular structures simply displays positions and types of atoms and bonds, neglecting topological and physical characteristics such as surfaces, voids, pockets, or electrostatic potentials.

We have taken an innovative, collaborative approach to address the above impediments. A visual computing environment, MDScope, has been developed which permits interactive modeling [1, 4, 6]. MDScope connects a powerful molecular graphics program, VMD, with a fast and efficient modeling program, NAMD, specifically designed for parallel computers. MDScope is ideally suited for high performance, distributed computing environments. The program presently links VMD to NAMD, the latter running on a remote high performance parallel computer. Three input methods have been explored in the framework of the VMD program to create an interactive 3D environment: a six (translation and rotation) degrees of freedom, electromagnetically tracked "mouse" has been integrated into VMD for manipulations of 3D objects; voice commands monitored through a microphone and recognized by a speech recognition interface have been introduced to replace cumbersome keyboard commands; hand gestures to manipulate the displayed model are detected through stereo cameras and interpreted using computer vision techniques. A large screen stereo graphics facility, expected to become the standard for biomolecular graphics, has been implemented and is frequently used by researchers.

A long-term goal of VMD is to enable multiple users to interact with the model simultaneously. This interaction and collaboration is expected to significantly shorten the problem-solving cycle in biomolecular modeling since users can guide searches, e.g., for optimal designs, without resorting to a time consuming, non-intuitive cycle of batch jobs carrying out automated searches. Incorporating voice commands will allow the user to be free of the keyboard, and hand gestures will permit the user to easily manipulate the displayed model and to explore different molecular configurations. The combination of both speech and hand gestures will be far more powerful than either individually, and its success will apply to interactive environments for structural biologists as well as to a wide range of other scientific and engineering applications. To accomplish this interaction, highly robust automatic speech recognition (ASR) and automatic gesture recognition (AGR) techniques are necessary. These techniques will be required for such problems as differentiation between commands and casual speech, and distinction between meaningful and meaningless gestures and hand movements.

The primary infrastructure for our project is a large-screen stereographic projection facility, developed in the Theoretical Biophysics Group and shared by a large group of biomedical researchers. The facility employs cost effective and space saving display hardware which is added to a high end graphics workstation and can be easily duplicated at other sites. It produces $8' \times 6' \times 6'$ 3D models in a 120 square foot area. The facility consists of a projector which displays alternating left- and right-eye views onto the screen at nearly twice the rate of ordinary projectors. The images, when viewed through special eyewear, produce a stereo display. In addition, a spatial tracker is used as a 3D input device for the development of a three-dimensional user interface. The program VMD has been designed for this environment as well as for standard monitors. Figure 1 attempts to convey an impression of the system through a photomontage of an actual image (nuclear hormone receptor–DNA complex) and of the actual space.

The key goal of our work is to simplify model manipulation and rendering to such a degree that biomolecular modeling assumes a playful character; this will allow the researcher to explore variations of their model and concentrate on biomolecular aspects of their task without undue distraction by computational aspects. The ultimate goal, illustrated in Fig. 1, will focus on wide ranging improvements of the graphical user interface of the existing programs MDScope which will become possible through the combined expertise of researchers in computational structural biology, parallel computing, numerical algorithm, and intelligent human-computer interaction technology.

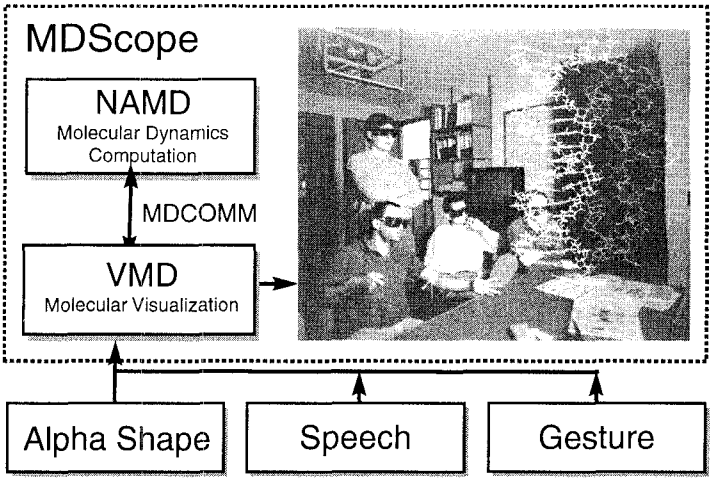

Figure 1. The program VMD, coupled with the program NAMD, and speech and gesture user-interface components. These facilities comprise MDScope, a problem-solving environment for structural biology.

2: MDScope, a Visual Environment for Structural Biology

MDScope is a set of integrated software components which provides an environment for simulation and visualization of biomolecular systems in structural biology. It consists of three separate packages which may be used individually, or together to constitute the MDScope environment [5]. These packages are:

1. The program NAMD, a molecular dynamics program which runs in parallel on a wide variety of architectures and operating systems.
2. The program VMD, a molecular visualization program which displays both static molecular structures and dynamic molecular motion as computed by programs such as NAMD.
3. The MDCOMM software, which provides an efficient means of communication between VMD and NAMD, and allows VMD to act as a graphical user interface to NAMD. Using MDCOMM, VMD provides an interface for interactive setup and display of a molecular dynamics simulation on a remote supercomputer or high-performance workstation, using NAMD as a computational "engine".

Molecular dynamics calculations are computationally very expensive, and require large amounts of memory to store the molecular structure, coordinates, and atom-atom interaction lists. The challenge of efficient calculation of the inter-atomic forces by using high performance computing is addressed in NAMD through the use of parallel computation and incorporation of the Distributed Parallel Multipole Tree Algorithm (DPMTA) [2]. NAMD uses a *spatial decomposition* algorithm to partition the task of computing the force on each atom among several processors. This algorithm subdivides the volume of space occupied by the molecule into uniform cubes (or *patches*), as shown in Figure 2, which are distributed among the processors in a parallel computer. The motions of the atoms in each patch are computed by the processor to which each patch is assigned; as atoms move they are transferred between the patches, and patches are reassigned to different processors in order to maintain a uniform computational load.

The key functions of the program VMD are to visualize biomolecular systems, to allow direct interaction between a user and a molecule being simulated on another computer, and to provide an intuitive user interface for controlling the visual display and remote simulation. VMD uses the

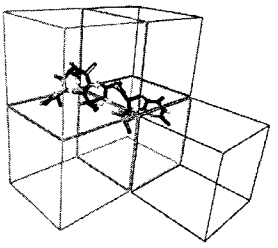

Figure 2. Spatial decomposition of a small polypeptide. Each cube represents a patch in NAMD.

MDCOMM software mentioned above to enable it to initiate, display, and control a simulation using NAMD. As the trajectory of a molecular system is calculated, the coordinates of each atom are sent from NAMD to VMD. Current network technology provides the necessary bandwidth to communicate the atomic coordinate data; the use of a high-performance dynamics program is crucial in order to furnish new data at the speed required for interactive display.

VMD implements many different forms of user interfaces – users may control the program through keyboard commands, a mouse, and a graphical user interface. VMD also implements a mechanism for external programs to serve as user interface components, by allowing them to communicate with VMD through standard network communication channels. This makes it possible for new user interface methods, such as the speech- and gesture-recognition systems discussed in Section 3, to be developed in parallel with VMD.

Development of MDScope is an ongoing project in the Theoretical Biophysics Group at the University of Illinois [1]. All three components of MDScope (NAMD, VMD, and MDCOMM) may be obtained via anonymous ftp (`ftp.ks.uiuc.edu`), or World Wide Web (`http://www.ks.uiuc.edu`). The components may be used individually or in concert, and include the complete source code for the packages as well as extensive documentation describing how to use and modify the programs.

Currently, NAMD is available for a wide variety of architectures and operating systems, including clusters of high-performance Hewlett-Packard, Silicon Graphics, and IBM RS/6000 Unix workstations, as well as the Cray T3D and Convex Exemplar parallel systems. We are in the process of making NAMD available on the IBM SP-2 and SGI Power Challenge architectures. NAMD should be compilable on any system with a C++ compiler and PVM version 3.3.11.

In addition to the full source and SGI binary distributions (IRIX 5.x and IRIX 6.x), the program VMD is already available for Hewlett-Packard (HP-UX 9 and HP-UX 10 using Mesa emulated OpenGL) and Linux workstations. Ports to other platforms, most notably IBM RS/6000 (AIX), will be available soon. VMD displays a graphical rendering of the molecular systems under study, and provides both a text-based console interface and a complete graphical interface for the user. These controls are used to modify the appearance of the molecules and display, to control the display of structural features of the molecules, and to access remote computers running molecular dynamics simulations. Multiple structures may be viewed simultaneously, and a flexible atom selection mechanism enables the user to easily select subsets of atoms for display. VMD includes an extensive text-command processing capability through the use of the Tcl library, a popular and widely available package for script parsing and interpreting. The use of Tcl makes it possible for users to write scripts including such features as variable substitution, control loops, and function calls. The current rendering of a molecule may also be saved in an image file or in a format suitable for use by several image-processing packages. Also, by connecting directly to a remote computer running a molecular dynamics simulation, VMD offers users the capability to interactively participate in an

ongoing simulation, e.g. the option to apply perturbative forces to individual atoms.

Speech and hand gestures are fundamental methods of human communication, and their use for interaction with and control of the display of VMD will greatly improve the utility of the program. Speech- and gesture-recognition user interfaces are being added to VMD to provide a "natural" working environment for researchers.

3: Speech and Gesture Interface

To fully exploit the potential that visual computing environments offer, there is a need for a "natural" interface that allows the manipulation of such displays without cumbersome attachments. In this section we describe the use of visual hand gesture analysis and speech recognition for developing a speech/gesture interface to VMD. 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.

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 [3]. 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 the VMD 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.

VMD 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" 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 3 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.

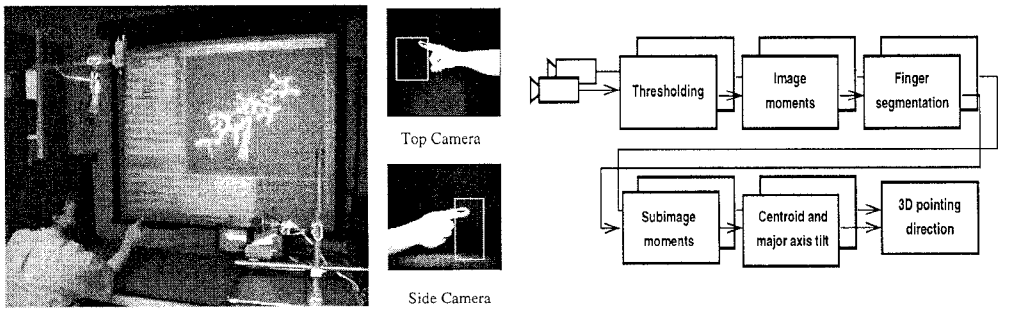

Figure 3. The experimental setup with two cameras used for gesture recognition (left) and overview of the AGR subsystems (right).

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.

3.1: Speech input using ASR

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) [8, 9, 12], 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.

Keywords. Table 1 lists the keywords and their phonetic transcriptions chosen for the experiment. These commands allowed the VMD user to manipulate the molecules and polymeric

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.

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 [9].

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. 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 [12] 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.

Recognition Network. The recognition syntactical network 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. 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'.

3.2: 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 [7] 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 3). 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 perform 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 4 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).

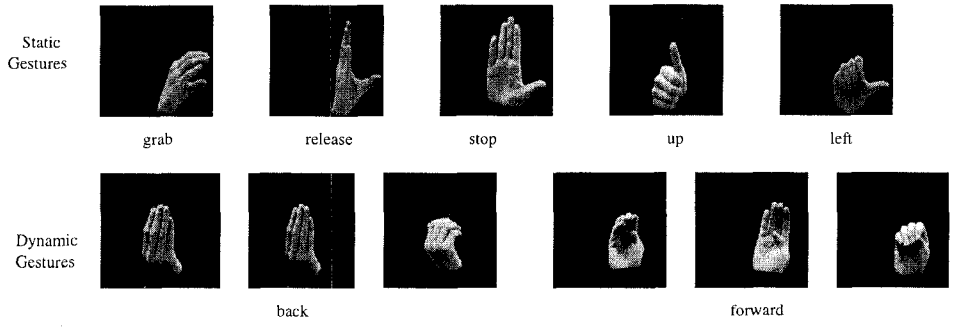

Figure 4. Examples hand gestures used in manipulating a virtual object and interpreted using AGR.

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 these features [11]. The set of gestures that were used consisted of both static and dynamic gestures (see Figure 4). The recognition system was built by training hidden Markov models for the specific gestures on sample 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.

The experimental 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 Ref. [10].

4: Discussion

Combining high performance parallel computing and high end graphics for research in structural biology will open new avenues for very large scale biomolecular modeling. Efficient parallel algorithms and powerful molecular graphics software provide crucial tools to address the challenges ahead in computational biology. MDScope, as a scalable, parallel and object oriented software package, will contribute to the ongoing effort of biomedical researchers to expand system size and time scales of the simulations. Furtheremore, MDScope provides an excellent testbed where computer vision and speech recognition techniques are used for building a natural human-computer interface, using spoken words and free hand gestures. The prototype speech/gesture interface presented 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 and forms a basis of future work

Acknowledgments

This work was supported by the Roy J. Carver Charitable Trust, by the U.S. Army Research Laboratory under Cooperative Agreement No. DAAL01-96-2-0003 and in part by the National Science Foundation Grant IRI-96-34618. Simulations were carried out at the Resource for Concurrent Biological Computing at the University of Illinois, funded by the National Institute of Health (Grant P41RR05969) and by the National Science Foundation (Grants BIR-9318159 and ASC-8902829). J.C.P. was supported by a Computational Science Fellowship from the United States Department of Energy.

References

[1] Source code and documentation for the MDScope applications are available via anonymous ftp (`ftp.ks.uiuc.edu`), or via the World Wide Web (`http://www.ks.uiuc.edu`).

[2] J. A. Board, Jr., R. R. Batchelor, and J. F. Leathrum, Jr. In *Proceedings of the 5th AIAA/ASME Thermophysics and Heat Transfer Conference*, pages 27–34, 1990.

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

[4] W. F. Humphrey, A. Dalke, and K. Schulten. VMD – Visual molecular dynamics. *J. Mol. Graphics*, 14:33–38, 1996.

[5] M. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. Kalé, R. Skeel, K. Schulten, and R. Kufrin. MDScope – a visual computing environment for structural biology. *Comput. Phys. Commun.*, 91(1, 2 and 3):111–134, 1995.

[6] M. Nelson, W. Humphrey, A. Gursoy, A. Dalke, L. Kalé, R. D. Skeel, and K. Schulten. NAMD— A parallel, object-oriented molecular dynamics program. *J. Supercomputing App.*, 10:251–268, 1996.

[7] V. I. Pavlović, R. Sharma, and T. S. Huang. Visual interpretation of hand gestures for human-computer interaction: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7), July 1997.

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

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

[10] R. Sharma, T. S. Huang, and V. I. Pavlović. A multimodal framework for interacting with virtual environments. In C. A. Ntuen and E. H. Park, editors, *Human Interaction with Complex Systems*, pages 53–71. Kluwer Academic Publishers, 1996.

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

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