Presented at the 10th International Symposium on Haptic Interfaces for Virtual Environment and Teleoperator Systems (2002 IEEE Virtual Reality Conference), pp. 225-230.

Building a Task Language for Segmentation and Recognition of User Input to Cooperative Manipulation Systems

C. Sean Hundtofte, Gregory D. Hager and Allison M. Okamura
Engineering Research Center for
Computer Integrated Surgical Systems and Technology
Johns Hopkins University, Baltimore, MD 21218
csean@cs.jhu.edu, hager@cs.jhu.edu, aokamura@jhu.edu

Abstract

We present the results of using Hidden Markov Models (HMMs) for automatic segmentation and recognition of user motions. Previous work on recognition of user intent with man/machine interfaces has used task-level HMMs with a single hidden state for each sub-task. In contrast, many speech recognition systems employ HMMs at the phoneme level, and use a network of HMMs to model words. We analogously make use of multi-state, continuous HMMs to model action at the "gesteme" level, and a network of HMMs to describe a task or activity. As a result, we are able to create a "task language" that is used to model and segment two different tasks performed with a human-machine cooperative manipulation system. Tests were performed using force and position data recorded from an instrument held simultaneously by a robot and human operator. Experimental results show a recognition accuracy exceeding 85%. The resulting information could be used for intelligent command of virtual and teleoperated environments, and implementation of contextually appropriate virtual fixtures for dynamic operator assistance while executing complex tasks.

1. Introduction

Our work is directed at the problem of developing effective *Human-Machine Cooperative Systems* (HMCS), which combine human decision-making with sensory-robotic enhancement to accomplish tasks that are otherwise difficult or impossible to perform. In the past, much of the work related to this problem has taken place in the context of telemanipulation. Telemanipulation systems have been used in remote and hazardous environments, where local human control is infeasible [1], and also as a way for learning and analyzing human manipulation [2].

In high-precision, non-hazardous environments such as microsurgery, there exists another paradigm of control: cooperative manipulation [3]. Here, the user directly manipulates the tool with his or her hand, but the tool is also attached to a robot. The robot's behavior can now be made compliant to user input and/or designed to enhance human performance. Examples of enhancements include tremor dampening, targeting assistance, force scaling, and restrictions on instrument movement, termed "virtual fixtures" [4]. In particular, virtual fixtures are paradigmatic of the HMCS approach: the user maintains primary control over task execution, while receiving guidance in the form of haptic feedback through the robot.

One problem that arises in HMCS is that of providing the *appropriate* assistance to a user, based on the intent or context of his or her actions. In order to do so, the system must have a model of the task being performed, and a means for relating the actions of the user to that model.

This paper examines the use of continuous Hidden Markov Models (HMMs) as a means for solving this problem. HMMs provide both an effective means of training (using the Baum-Welch Algorithm) and an effective means of data segmentation (using the Viterbi algorithm). More specifically, we examine whether it is possible to develop models of multi-step tasks or activities by training appropriately structured HMMs on raw data acquired from users.

Most of the previous work on HMM-based task segmentation and gesture recognition has been directed at telemanipulation systems [5,6]. The input data is force/torque and velocity input from the user to a master device. However, this previous work has used HMMs to model this data at the task level (Figure 1). By way of comparison, the equivalent speech recognition system would use an HMM for a complete word with each state being used to explain a specific phoneme. In general, this is not the case. In fact, speech systems use networks of HMMs at the phoneme level, and words are assembled from these networks.

This paper examines a similar approach to activity recognition. More specifically, we are interested in two questions:

1. Is it possible to model small portions of a task, which we term *gestemes*, using continuous HMMs applied to raw sensor data?
2. Can gestemes be generalized across activities? That is, is it possible to build a small vocabulary of gestemes that can be assembled to describe a much larger range of activities?

We will show that even complex, repeating gestures can

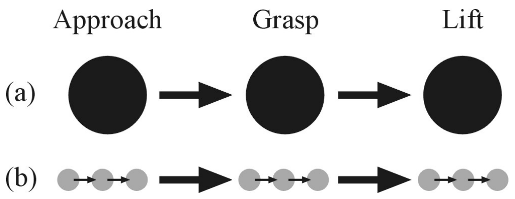


Figure 1. A simplified example of states and transitions of HMMs at the (a) task/word level and (b) gesteme/phoneme level for a “pick up object” task. Start/Finish null states and cyclic edges have been avoided.

be well explained using networks of continuous HMMs and standard training and segmentation algorithms.

The remainder of this paper is structured as follows. In the next section, we provide a brief introduction to HMMs. In Section 3, we describe our system and training methods. In Section 4, we present experimental results, and in Section 5 we summarize our results and discuss future research directions.

2. Background

HMMs are based on the notion of a Finite Markov Chain. Specifically, consider a sequence of random variables S_0, S_1,… taking values in a finite state space *S*. Intuitively, S_i represents the state of some underlying process at time i. Such a chain is a *Markov chain* if the following property holds:

Markov Property: $(\forall k > 0)\ P[S_k|S_0,\ldots S_{k-1}] = P[S_k|S_{k-1}]$,
where S_k is the state occupied at time k

A *Hidden Markov Model* (HMM) is a Markov chain with the following additional attributes:

1. An output alphabet *Y.*
2. A unique starting state s_0.
3. A matrix of transition probabilities P(s’ | s) for all s,s’$\in$ *S.*
4. An output probability distribution P(y | s, s’) for all y $\in$ *Y* associated with transitions between s and s’.

An HMM is “hidden” because the state changes in the Finite Markov Chain that make up an HMM are unobservable. However, there is output associated with occupation of those states (or the transitions between them) and that output is observable. In analysis, HMMs are given initial parameters: their structure, number of states, connectivity, and characteristics of the output from each state. Using segmented training data (i.e. data labeled with the corresponding HMM state), one can train an HMM that is the most likely set of transition probabilities, means and variances that give rise to the observed data. Once trained, it can then be applied to previously unseen test data for the purposes of classifying the data according to the output alphabet.

The theory of both training and applying HMMs is highly developed, particularly in the area of speech recognition. Two important algorithms are the Baum-Welch algorithm, which can be used to compute HMM parameters during training, and the Viterbi algorithm, which seeks to give the likeliest HMM state changes to explain observed data. Readers interested in a more thorough explanation of Hidden Markov modeling and related algorithms are recommended to read [10].

Most recent efforts [7, 8] have focused on HMMs as a stochastic similarity measure for different users and different levels of expertise in performance of certain procedures. Earlier work was done on the use of HMMs for telemanipulation [5] and gesture recognition [6], but relied from the outset on some task knowledge. This task knowledge was used in the segmentation of training data (by experts) or in first pass classification of the data, discriminating the observed data into one of several discrete gestures with a classifier, and using the HMM as a regularization method in a second pass.

In contrast, our work shows that the data can be segmented and states recognized using HMM methods without user intervention or other heuristic segmentation means. Our approach is to first develop multi-state HMMs that correspond to component actions, or “gesmes,” that make up a task. We then create a task-level system in which each task state corresponds to a gesteme in a one-to-one mapping (Figure 1b). For example, instead of recognizing a 5-second pushing motion with occupation of a single state in an HMM, our system might better recognize it with occupation of 5 sequentially connected states. We show that continuous “gesteme-level” HMMs trained on the raw input data with user segmentation are able to capture transitory periods (e.g., when a push or a pull begins) of user activity. By connecting

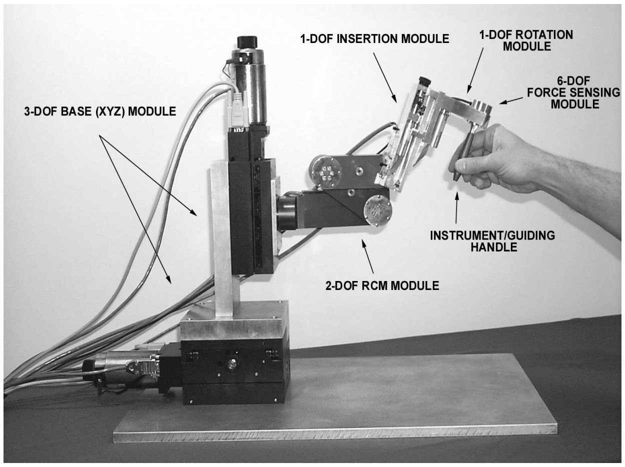


Figure 2. The JHU Steady Hand Robot.

these into a network, a complete task trace can be segmented.

Moreover, highly or fully connected networks of gestemes are capable of expressing many different styles of procedure, or different procedures themselves, using a simple, adaptable language. Such networks can be used in outcomes analysis and examination of procedural styles (looking for correlations between methods of achieving a goal and the success rate in achieving the goal). Our system can also capture different ordering and repetition of the individual task states while still performing well as a recognition system. To our knowledge, such fully connected networks have only been used in systems for measuring user expertise.

In general, the HMM techniques described in this paper could be applied to any system that tracks a user's input to a virtual, teleoperated, or cooperative manipulation environment. Our work uses a cooperative manipulation system: the Steady Hand Robot (Figure 2) [3]. Developed for microsurgical procedures, the Steady Hand Robot has been proven to reduce the effects of tremor on accuracy, and allows the surgeon to feel amplified contact forces [3]. The robot has an instrument as its end-effector, which the user grasps and uses to perform a task as if it was not attached to a robot. Thus, the robot functions as an admittance control haptic interface to a real environment.

One problem associated with this system is the magnitude of resistance it inserts between user input and robot movement. At fine scales, this decrease in bandwidth is generally not viewed as a problem --- in fact, it is extremely effective at reducing tremor and other undesired motion. However, during large-scale motion it can become cumbersome.

Thus, we are faced with a tradeoff: use high force gains and/or low control stiffness to allow for large freedom of motion, or use low gains and/or high stiffness to enhance precision. If separate modes of operation were available (e.g., one for large scale motion and another for fine positioning), the system would be more responsive to user intent and thereby increase HMCS performance. Previous work has shown this to be the case, but required user input via foot-pedal to modify the behavior of the robot [9]. Our goal is to show that a supervisory/look-over-the-shoulder system could be used for the same purpose: to identify the operator's intent.

3. Data Acquisition and System Training

Our experimental system consists of modeling and recognition software [11], machine-level robot control software accessed through an interface to the JHU Modular Robot Control (MRC) library [9], and the JHU Steady Hand Robot (SHR) [11]. The JHU SHR is a 7-degree-of freedom manipulator with XYZ translation at the base for coarse positioning, two rotational degrees of freedom at the shoulder, and instrument insertion and rotation stages. A force sensor is integrated into the end effector. This robot has a remote center of motion and an overall positional resolution of less than 10 microns. In these experiments, 6 degrees of freedom of motion were used, with z movement allowed in the instrument insertion stage and Z base translation inactive.

The input data acquired from the robot consisted of 7 variables: Cartesian force and torque expressed in robot end-effector coordinates, and the magnitude of translation between the last reading and the current one. These state variables were chosen so that any system built from them could be applied to free-hand instruments with Force/Torque sensors and accelerometers (whose positional information would suffer from drift in error across integration). Also, contextually dependant variables, such as absolute position information, were avoided to eliminate any dependence on specific task context (e.g. the absolute or relative location of task elements).

3.1 Data Acquisition

We investigated two tasks: (a) constrained peg-in-hole and (b) a painting-scheme task. Both tasks share certain gestures (described more fully below) including *approach*, *position instrument*, and *withdraw*. In the microsurgery domain, these tasks are analogous to (a) retinal vessel cannulation, where the tip of a needle is inserted into a blood vessel, and (b) retinal-membrane peeling, where micro-forceps or a vitrector is used to remove membranes that have formed at the back of the retina.

For the peg-in-hole task, the instrument (a needle of diameter 1mm) and robot were placed to the right of the operator. A target consisting of a 1cm hole on the round surface of a metal cylinder with a diameter of 2cm was placed directly before them. The position of the cylinder was changed slightly for each experiment, but remained roughly 30 cm to the left of the robot's starting position.

Before beginning the procedure, the users were given a short description of the task, including a list of high-level task states (roughly corresponding to the gestemes used in later analysis). They were instructed to press a foot-pedal during the procedure to signify when they felt they were making transitions between task states. The task states were:

- *PLACE*: move the instrument from a starting position to some point close to the hole
- *POSITION*: align the instrument parallel to the hole
- *INSERT*: move the instrument tip through until touching the opposite inner cylinder wall
- *WITHDRAW*: carefully withdraw the instrument tip
- *REMOVE*: move instrument roughly back to the starting position

For the painting scheme, the user was instructed to approach a flat surface with the needle tip, and to draw 4 parallel lines, 2 in each direction (Figure 3). The sequence of actions given to users in this case was:

- *PLACE*: move the instrument from a starting position to some point close to the painting surface
- *POSITION*: align the instrument to be perpendicular to the surface
- *PAINT*: draw lines with the instrument tip
- *REMOVE*: move the instrument tip away from the surface

As noted above, the *PLACE, POSITION*, and *REMOVE* states were effectively the same for both tasks.

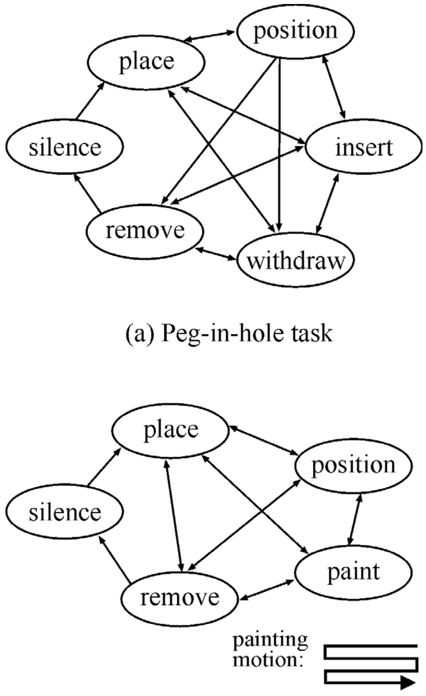
5 users performed the peg-in-hole task approximately 10 times. 3 of the same five users performed the paint task 5 times. Users were allowed to practice the task and foot pedal switching before data was collected. For the peg-in-hole task, the average time varied between 18.83 and 42.27s, demonstrating a large variation in user style and skill level.

3.2 HMM training and system design

We gave a linearly sequential, left-to-right (SLR) structure to the HMMs of each gesteme. In this model, there are distinguished start and end states, and a fixed number of *transmitting* states that are connected in a linear, directed chain from start to end. The number of transmitting states per gesteme was varied to explore the effect on accuracy. Networks of these HMMs, without probabilities assigned to transitions, were then used in subsequent task recognition tests. In most cases, these networks had transitions from any state to any other state, the only requirement being that they start and finish with the quiescent (silence) state (Figure 3). This implies, in particular, that the HMMs had no a priori *sequence* information about either task.

The observable data used as input to train the system was a sequence of 7 dimensional data vectors and corresponding foot-pedal presses for user segmentation of the data into user action states (*PUSH, PULL, PLACE* etc). Quiescence (a period lacking robot motion) was also segmented from the input data as any period with an L2 norm less than a threshold.

The Hidden Markov Model Toolkit (HTK) [12] was used to provide initialization of HMMs, Baum-Welch reestimation, and later to perform Viterbi segmentation. Accuracy of the segmentation was measured in two different ways: *sample accuracy* and *sequence accuracy*. Sample accuracy is the percentage of samples in the output that share the same label as the user-assigned label. Sequence accuracy measures whether the sequence of high-level labels output from the system are the same as user-labels without insertion or deletion. Sequence accuracy is a more linguistic measure of accuracy, as it focuses on whether the correct states have been recognized in the right order, rather than whether the HMM is occupying the "correct" state at any given time.



(a) Peg-in-hole task

(b) Painting scheme

Figure 3. Network diagrams for the two tasks. Nodes represent separate HMMs.

The system was also used to perform alignment: i.e., given a correctly ordered sequence of labels, find the best alignment of the state boundaries (in which case only sample accuracy would be of interest, as sequence accuracy would be 100%). In the case of alignment, a linearly ordered sequence of gestemes was given to the Viterbi algorithm, rather than the fully connected network used for segmentation.

For performance, the leave-one-out method for training/testing was used. The system was trained on all but one of the possible input sequences for a task/user and tested on the one left out. Averages were taken of performance over the entire set of training/testing combinations.

4. Results and Discussion

A typical task recognition result is shown in Figure 4. Observing the 7 input data vectors, it is clear that it would not be straightforward to perform the segmentation manually from the data. Yet, the HMM approach generally segments the task with the correct gesteme ordering and good temporal accuracy, when compared with the operator's segmentation.

Table 1 shows the individual user results of peg-in-hole task segmentation when using a fully connected net-

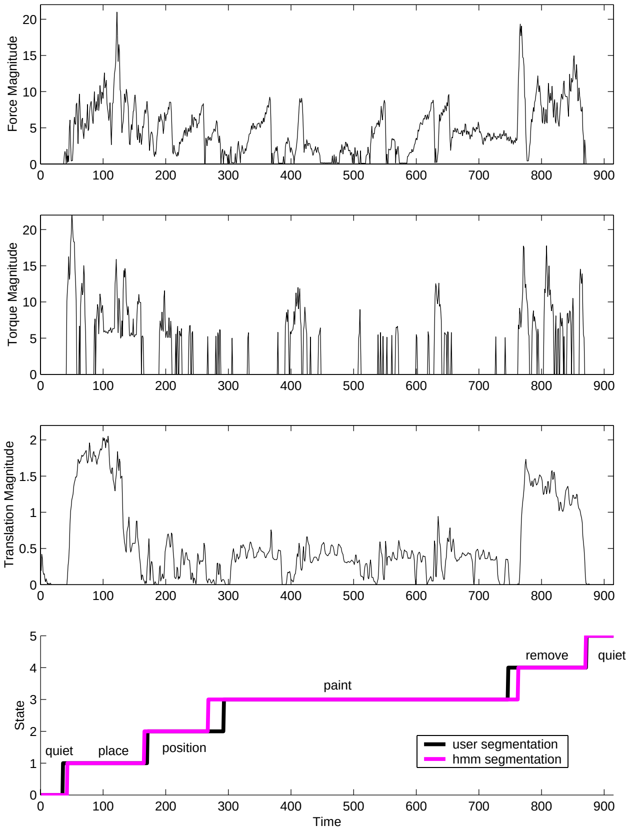


Figure 4. Typical recognition results for the painting scheme task with a fully connected network. The top 3 plots are the force, torque, and translation data acquired during task execution, and the bottom plot shows the user and HMM segmentations.

Table 1. Individual user results from a 7-state HMM performing task segmentation on the peg-in-hole task. Reported values are percentage correct, mean and variance.

User #	Alignment (μ, σ)	Recognition (μ, σ)
1	91.5 3.4	89.33 6.0
2	91.27 3.0	88.7 5.7
3	88.58 6.3	78.66 7.6
4	84.95 6.3	76.17 8.6
5	90.62 2.9	87.04 5.3

Table 2. Obtained accuracies using 3, 5, and 7 non-null (transmitting) states for the peg-in-hole task. Values are mean percentage correct.

Number of States	Alignment	Recognition
3	87.6	83.2
5	89	83.9
7	88.9	84.3

work of 7-state gesteme models. These results are averaged over the 10 runs for each user. Table 2 demonstrates that the accuracy of alignment and recognition are affected by the number of non-null transmitting states used in each gesteme, although there is no significant trend.

For the paint task, we computed the alignment and recognition accuracies using 7 state gestemes. The average sample accuracies of the system over all users were 90.72% for alignment and 90.74% for recognition.

For both tasks, the average *sequence* accuracy when all gestemes were present in a fully connected network was 97.62%. This high sequence accuracy result (recognizing the correct sequence of gestemes rather than the right one at the same time) is very promising for higher-level task recognition. This could be used to answer the question: Is the user performing task X (with goal x) or task Y (with goal y)?

We also considered the role of silence in the segmentation. We found that silence modeling had a two-way effect on results. If we allowed short pauses in the middle of user execution of our abstracted task steps (that is, during gestemes), not just silence as begin and end states, the labeling of the *training* data became more accurate. However, when the same data was used as test data, the system ignored such small pockets of inactivity and instead recognized them as products from the initial and end states of other gestemes. This led to a *reduction* in accuracy on the test data. More precisely, when "short pauses" during task execution were modeled, the alignment accuracy was reduced by 1.08% and the recognition accuracy was reduced by 2.5%.

It is worth noting that a large amount of the error in these results arises from the alignment of gesteme transition boundaries (cf. ~97% sequence accuracy to ~89% sample accuracy). This raises the obvious question as to whether user segmentation is always more valid than automated segmentation. Using a foot-pedal at the same time as performing a task tends to be somewhat distracting, and, in some cases, clearly introduces unnatural pauses into the data.

As noted at the outset, one of the principal goals of this work was to demonstrate that the gestemes developed and trained for one task are applicable to other tasks in the same domain. To this end, we trained the *PLACE*, *POSITION*, and *REMOVE* gestemes on all user's data from the paint task. We then used those models to perform recognition across all users on the peg-in-hole task. The resulting decrease in recognition performance was only 1.08% compared to the leave-one-out method. Thus, it seems fair to conclude that these gestemes can be considered to be "task independent" for the limited situations we have considered.

5. Conclusion

We have examined the problem of modeling multiple human-machine cooperative tasks using a single task language. The use of HMMs to model the components of this task language, and to parse user data has given promising initial results. Most importantly, we were able to achieve these results without using any prior segmentation or labeling of the data, other than that acquired from the users during task execution.

Starting with these initial results, additional tuning of the system could be done to improve alignment and recognition performance. All the parameters in the Hidden Markov Modeling (pruning, insertion penalties, etc.) should be more thoroughly investigated. Projection of the data or classification based on task knowledge could also be used to better compare this system with previous results using discrete HMMs.

The effect of expertise on system performance would be interesting to study further. In these experiments, the user with the largest amount of procedures completed also had the most easily parsed data, but this cannot be statistically verified in this work due to the low number of users.

There are also many obvious extensions to our current system. For example, the vocabulary should now be expanded to handle more (and more complicated) procedures. For a given task, it would also make sense to create the equivalent of phonetic bigram or trigram models. This would allow us to include some "soft" notion of sequencing information in the task structure.

As noted previously, the use of the foot pedal to provide ground truth is often artificial, and may be contaminating the data. Synchronized visual footage of the procedures (with segmentation being done offline) would be helpful for examining the effects of this on results.

The eventual goal of this research is provide real-time assistance to users in teleoperated and cooperative manipulation environments. The work described in this paper uses force and translation magnitude information as input to a recognition, but other inputs will also be useful. Visual cues (video taken from a camera for tracking and targeting purposes), acceleration, and absolute position information are candidate inputs. Contact force sensing at the instrument tip could measure interactions with the environment and help express more varied tasks.

Our long-term goal is to use this system in realistic scenarios, including microsurgical procedures such as vessel cannulation and retinal montage building.

Acknowledgements

The authors would like to acknowledge authors of the Hidden Markov Model Toolkit (HTK) [11], Dr. Asela Gundwarna for his help with HTK, and Dr. Rajesh Kumar for assistance with the Steady Hand Robot. This work was supported by the National Science Foundation grants ERC-9731748 and RHA-099770.

References

[1] T. B. Sheridan, Telerobotics, Automation and Human Supervisory Control, MIT Press: Cambridge MA, 1992.

[2] T. Debus, P. Dupont, R. Howe, "Automatic identification of local geometric properties during teleoperation," *Proceedings of the International Conference on Robotics and Automation*, Vol. 4, 2000, pp. 3428-3434.

[3] R. Kumar, P. Berkelman, P. Gupta, A. Barnes, P. Jensen, L. L. Whitcomb, and R. H. Taylor, "Preliminary experiments in cooperative human/robot force control for robot assisted microsurgical manipulation," *Proceedings of the IEEE International Conference on Robotics and Automation,* Vol. 1, 2000, pp. 610-617.

[4] A. Bettini, S. Lang, A. Okamura and G. Hager, "Vision Assisted Control for Manipulation using Virtual Fixtures," *Proceedings of the International Conference on Intelligent Robots and Systems,* 2001, pp. 1171-1176.

[5] B. Hannaford and P. Lee, "Multi-Dimensional Hidden Markov Model of Telemanipulation Tasks with Varying Outcomes," *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics,* 1990, pp. 127 -133.

[6] P. K. Pook, "Teleassistance: Using Deictic Gestures to Control Robot Action," PhD thesis, Department of Computer Science, University of Rochester, 1995.

[7] J. Rosen, M. Solazzo, B. Hannaford and M. Sinanan, "Objective Laparoscopic Skills Assessments of Surgical Residents Using Hidden Markov Models Based on Haptic Information and Tool/Tissue Interactions," *Medicine Meets Virtual Reality,* Vol. 81, 2001, pp. 417-423.

[8] M. C. Nechyba. "Learning and Validation of Human Control Strategies," PhD thesis, The Robotics Institute, Carnegie Mellon University, 1998.

[9] R. Kumar, "An Augmented Steady Hand System For Precise Micromanipulation," Ph.D. Thesis, Department of Computer Science, The Johns Hopkins University, 2001.

[10] F. Jelinek, Statistical Methods for Speech Recognition, MIT Press: Cambridge, MA, 1998.

[11] R. Taylor. P. Jensen, L. Whitcomb, A. Barnes, R. Kumar, D. Stoianovici, P. Gupta, Z. Wang, E. deJuan, and L. Kavoussi, "Steady-hand robotic system for microsurgical augmentation," *International Journal of Robotics Research*, 18(12), 1999, pp. 1201-1210.

[12] The Hidden Markov Model Toolkit (HTK), http://htk.eng.cam.ac.uk