Modeling and Classifying Six-Dimensional Trajectories for Tele-operation Under Time Delay

Vytas SunSpiral(1)(3), Mark B. Allan(1), Rodney Martin(2), Kevin R. Wheeler(2)

(1)QSS Group Inc., NASA Ames Research Center, MS 269-3, Moffett Field, CA, USA 94035
(2)Intelligent Systems Division, NASA Ames Research Center, MS 269-1, Moffett Field, CA, USA 94035
(3)Formerly published as Thomas Willeke
Email: Vytas@email.arc.nasa.gov

Abstract

Within the context of tele-operating the JSC Robonaut humanoid robot under 2-10 second time delays, this paper explores modeling and classifying human motions represented as six-dimensional (position and orientation) trajectories. A dual path research agenda is reviewed exploring both deterministic and stochastic approaches using Hidden Markov Models. Finally, recent results are shown from a new model that integrates these two research paths. In future work it will be possible to automatically generate autonomous actions by reusing these same predictive models of human behavior to be the source of autonomous control. This approach may change the role of tele-operation from being a stand-in for autonomy into the first step of mentoring generative models capable of autonomous robotic control.

Introduction

The contextual question for this research has been how to best issue remotely operated commands for the JSC (NASA Johnson Space Center) Robonaut platform given a 2-10 second time delay. Robonaut is a humanoid robot designed to have dexterous manipulation capabilities similar to those of a suited astronaut. This enables it to perform many of the construction and repair functions that would currently require an extra-vehicular activity (EVA) (Bluethmann et al. 2003). Besides reducing the number of dangerous EVA's that must be performed, Robonaut would also be able to perform emergency repairs. Currently it takes an astronaut 4-5 hours to suit up for an EVA. If there is a critical emergency which must be repaired quickly, this delay may doom the astronauts. Thus, if the Robonaut were able to deploy immediately it might be able to handle the emergency repair before a human could even start the EVA. How should this robot be controlled? Fully autonomous control for such a high degree of freedom robot that is both verifiably robust and safe is a long term research goal and is unlikely to be flown soon. Thus, current and mid-range research objectives have been focused on controlling Robonaut through some form of tele-operation. While success has been shown in the lab with fully immersive tele-operation, difficulties emerge when a time delay is introduced. The 2-10 second time delays that this project is concerned with can occur if there is a human controller on the ground and the Robonaut is in LEO (up to two second delay) or on the moon (8-10 second communications delay). While these distances are not so great, it is the nature of the communications networks and topology which introduce such large time delays. When a multi-second time delay is introduced, direct tele-operation slows down tremendously since, for safety reasons, the operator is forced to adapt a "bump and wait" strategy. This entails making a small movement (the "bump") and then waiting to get feedback on the results of that motion. Controlling the robot with a "bump and wait" tele-operation strategy is slow and tedious, and does not optimize the speed advantage of deploying the robot for a critical emergency repair.

Thus, in order to deal with the time delay a sliding scale of autonomy is proposed. This is a hybrid human/robot control scheme that keeps a human in the loop (and ultimately in control for safety reasons), yet allows for and greatly increases operation speeds. This is done by allowing small, well understood and conditioned tasks (which are often the routine tasks) to be performed autonomously. The granularity of these autonomous tasks can vary from no autonomy (full teleoperator control), to simple motions such as positioning the hand close to (but still a safe distance away from) an object of interest, to compound tasks such as reaching towards a known object, and then grabbing and retrieving it.

Once capabilities for these autonomous tasks exist, one must ask how they will be commanded. As will be shown below, the operator engages in supervisory control of the Robonaut by operating in an immersive 3D virtual environment. Thus, the specific goal of this work is to be able to recognize and predict when the operator is performing one of the pre-defined autonomous tasks so that the action can be triggered early and completed by the robot. The control of these autonomous actions is purposefully kept in physical context (as opposed to developing some command grammar) to facilitate the seamless transition between direct tele-operation control and the varying levels of autonomous actions.

Along side this goal of helping control the Robonaut under a time delay, we are also using this work to advance our research into automating the process of creating autonomous robotic actions by modeling human behavior. If we can accurately understand and model the methods humans use to solve certain problems, then it is possible that those very same models can be used to control a humanoid robot like Robonaut to accomplish similar tasks. Thus, we view tele-operation not simply as a means to accomplish a task in the absence of robust autonomy, but rather as the first step in building models of human action for the purpose of developing robust autonomous control (Peters et al. 2003).

The current method of controlling Robonaut involves the operator wearing two data gloves that are used to measure finger joint positions, and two magnetic trackers used to measure the x-y-z position and roll-pitch-yaw orientation of each hand (end effector). The position and orientation information is then transmitted to the robot as end effector position commands. The number of degrees of freedom in the elbow and shoulder are constrained to enable this position while maximizing strength. For safety considerations, the rate of movement of the arms is limited; thus the operators are trained to match or move slower than this rate. Most of the feedback to the operator comes from the stereo cameras mounted in the head of Robonaut and transmitted back to the tele-operator's head mounted display. An operator will reach for an object so that his view from the head mounted cameras is not obscured by the hand. This results in some simple tasks taking a very long time to accomplish. For example, in grasping a hand rail (as a rung of a ladder) the operator must make sure that the fingers can safely wrap around the rail using the stereo visual cues. This action typically takes several seconds for an experienced operator.

Experiments

We chose two basic tasks, retrieving a hand rail mounted vertically and dropping it into a box, and retrieving a hand rail mounted horizontally and dropping it into a box. The hand rails are mounted with Velcro on a board, affixed to a stationary wall. The target box is a flexible cloth box that is open but is not within the same field of view as the hand rails. These tasks were chosen as a first step towards automating climbing on a space habitat or operating a drawer pulling sequence.

The tasks consist of the following steps:

1. Start in initial position/state
2. Look down at hand (substitute for proprio-receptive feedback) and then at hand rails
3. Reach for specified hand rail (either vertical or horizontal according to plan)
4. Grasp hand rail
5. Remove hand rail from wall (pull)
6. Move hand rail over box
7. Drop hand rail into box
8. Return to initial position

The Robonaut can be operated via a simulated environment, so that the operators can perform tasks without regard for the time-delay normally associated with long distance operations. The motion commands generated in the simulated environment are then sent to the actual robot for execution. For this experiment, inexperienced operators tended to have greatly varying behaviors, whereas the variance in the data was negligible for the most experienced tele-operator. Fig. 1 shows the simulated environment in which the experiments discussed in this paper were conducted. These experiments were conducted on many different days over six months. Initial conditions varied noticeably from day to day.

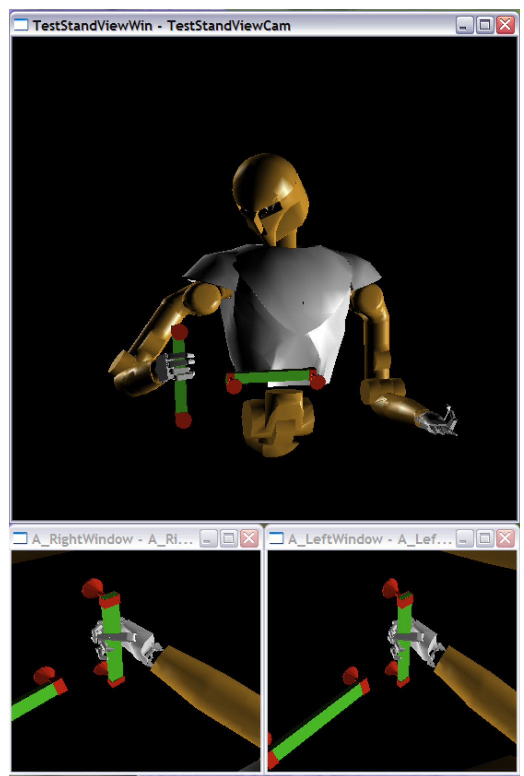

Figure 1 – Top: Simulation based experiment. Bottom: The operator's view from the left and right cameras of the simulated hand rails.

Technical Problem

While there are a number of technical and design challenges to the larger problem we are solving, the core problem is modeling and classification of the six dimensional trajectories.

Each trial can be cut into a sequence of trajectory segments. In this work we focus on the first segment, which goes from the start position to one of the possible graspable objects. These trajectories are a time-stamped sequence of the X, Y, Z, location of the operator's hand and the associated rotation matrix, which we cast into a roll, pitch, yaw representation. Thus, the trajectories are a time sequence of six-dimensional data points with messy initial conditions. Since the operator starts in approximately the same location for every motion, the initial segments of all the trajectories are heavily overlapped, making it difficult to clearly separate them in the early part of the motion (Fig. 2). Furthermore, this is compounded by a large day-to-day variance on the operator's start position. (Within a single day's trials the trajectories tend to cluster fairly nicely). This non-stationarity means that there is a wide variance on the motion and location of the trajectories, and that in some cases the initial motions towards different objects end up looking very similar (Fig. 3).

Our task is: given a set of example six-dimensional trajectories to certain known points in space, develop meaningful models of the different trajectories, and then use those models to classify a new trajectory in real time, as soon as possible (*i.e.* while having seen only some initial segment of the trajectory), and with no false alarms (*i.e.* it is better to not classify than to give the wrong classification).

Notice that the solution to this problem, which was motivated by tele-operation and user intent prediction, can be used in other ways. Specifically, it really is about classifying and predicting human gestures. Thus, if a robot were working side by side with a human, and assuming there was a good solution for visually capturing the motion of the human's hand or arm, this trajectory classification method could be used to comprehend the intent behind the human's motions, such as for which object the human was reaching. And, as mentioned earlier, we hope that the same models could be reused to generate autonomous control.

Development Approach

We explored in parallel different approaches to solving this problem. One track continued with previous work by applying stochastic methods, such as using Hidden Markov Models to classify the trajectories. The other track looked at more deterministic methods and ended up exploring different spatial representations of the data. This research approach has been very successful, with the two different tracks starting to converge and borrow techniques from each other.

In the following section we will try to present the evolution of our thoughts on these two different tracks, and show how they have influenced each other.

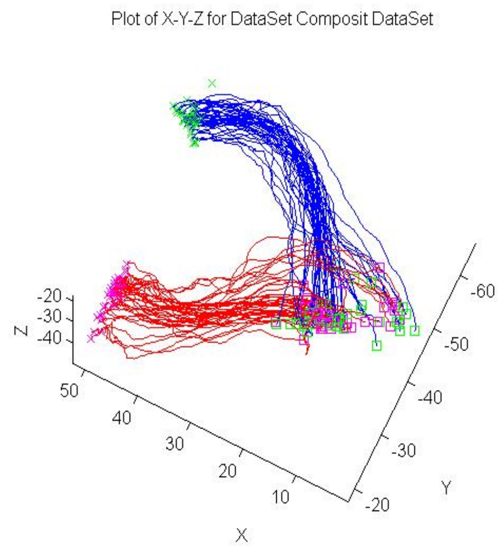

Figure 2 - This is a plot of the XYZ position information of many trajectories to two different handles. The start location on the right shows how the initial conditions for the different trajectories are heavily overlapped.

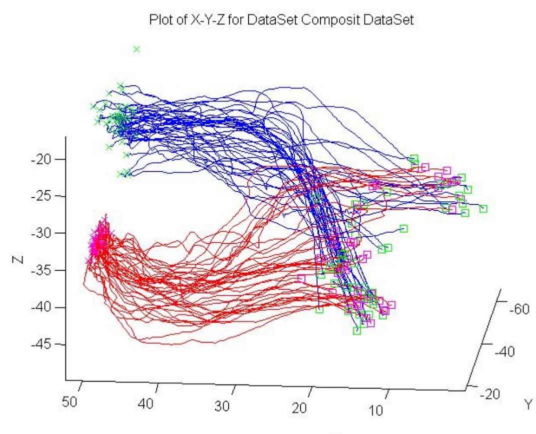

Figure 3 - This is the same data as figure 2, but viewed at a different angle to highlight the spatial variance of the start positions.

Deterministic Approach

At first we thought that predicting which object an operator was reaching for would be easy to solve using some simple heuristic. It turned out to be a much more interesting modeling problem than we anticipated. This section will give a high-level outline of our exploration of the solution space, and will show how we developed our newest algorithms.

Our first thought was to calculate which object the operator was "moving towards the most" -- i.e. find the first derivative of the motion, using position information exclusively. The major lesson from this approach was the importance of orientation. We found examples in which the operator was positioning the hand to be in the correct orientation to grab the target handrail, yet the motion to do this involved moving towards some other object.

Our next step was to combine instantaneous orientation and motion vectors, but we still ran into problems. Looking at instantaneous minimum values is similar to a minimal path approach (greedy algorithm). It turns out that operators do not follow a minimum-distance path in either position or orientation space. The operators have been trained to move in such a way as to avoid singularities, self contact and other hazards for the robot. Also, since control is purely visual (they use no force-feedback information), they always move so as to maintain visual contact with both the object and the hand. This leads to a preference for underhand grasps so that the operator can see when the object is properly grasped. An overhand grasp would occlude the object being manipulated. All these constraints result in motion paths that are not necessarily the shortest distance or most direct motion towards the goal.

While it might be possible to enumerate all the competing optimization constraints that the operator is solving (occlusions, singularities, collision avoidance, etc), we started modeling the trajectories following our belief that good models can be used to create autonomy. We started by explicitly modeling the gesture trajectories in six-dimensional space (X,Y,Z, Roll, Pitch, Yaw). To make this a proper spatial model that was independent of time (and thus independent of the speed at which the operator moved), we normalized along the length of the trajectory itself. In other words, we recast the data into a new set of data points which were a known distance apart along the piecewise linear direction of travel. The overall shape of the plotted trajectory stayed the same, but the actual data points which represent the trajectory were now equidistant along the trajectory itself.

Using this new representation an "average" trajectory was computed for each graspable object. A method of comparing two trajectories was developed by calculating the average distance error and then novel trajectories were compared to these canonical averages. The weakness of this is that the high degree of variance and overlap, especially in the early part of the trajectories, caused many false alarms (i.e. matching to the wrong trajectory).

In order to capture this variance a memory based approach was experimented with where all the trajectories in the training set were kept in memory and the novel trajectory was compared to all of them to find the best match. A variety of different distance metrics were experimented with. Under certain conditions this was found to work fairly well, but was fragile to the inclusion of outlier examples in the training set.

We decided that for a general solution involving models of many different grasp trajectories, this memory based approach would bog down. Thus we started looking at statistical methods to represent clusters of trajectories. This is where we really started to converge with the stochastic methods which were being explored in parallel. We will discuss that line of research next, and then discuss the new converged algorithms in the last section.

Stochastic Approach

The research group has a strong body of experience in applying Bayesian techniques to gesture recognition problems (Wheeler and Jorgensen, 2003). The techniques used for this work have its roots solidly grounded in formal probabilistic graphical modeling. Using the junction tree algorithm, DBN's (dynamic bayes nets) can easily meet the needs of any number of inferential or parameter learning problems. In this case, the parameters we would like to learn are those of a Gaussian mixture model, or multi-modal Gaussian distribution which would describe a cluster of example trajectories. In this approach we assume that human motion, at a coarse level, is Markovian in nature and can be broken down into discrete chunks – such as a "start phase," a "reaching phase," and a "grasping phase." In turn, we hypothesize that each of these chunks can be characterized probabilistically by the aforementioned Gaussian mixture model. This is very similar to what is done in voice/speech recognition, where words are broken down into phonemes. The use of the Hidden Markov Model (HMM) encapsulates all of these ideas (DBN's, Markov chains, Gaussian mixture emissions), and is ubiquitous in the voice/speech recognition community. However, these techniques have also been applied by other researchers interested in the field of motion classification and detection for video sequences. This field is related closely to our problem of motion trajectory classification, and they use similar intuitive arguments for applying formal probabilistic methods (Porikli 2004), (Bregler 1997), (Zhou, Gong, and Tao 2005), (Makris and Ellis 2002), (Fablet and Bouthemy 2001).

Much of the Bayesian approach applied to the current problem of modeling and classifying 6-D trajectories for tele-operation under a time delay has been documented in previous work (Wheeler et al. 2005). The common theme has been the use of the HMM as the fundamental modeling paradigm.

Feature selection has proven to be a very important factor in how well the models characterize the experimental motion trajectory-based data. Additionally, it has played an important role in how consistently the models and real-time recall thresholds can be optimized to achieve the goals of no false alarms, minimum missed detections and time to prediction. Feature selection refers to the choice of several different combinations of feature vectors that can be used (i.e. what comprises y_t - the observation vector at time t having dimension n). These feature vectors act as templates for the observation sequences used to train and recall the hidden Markov models. “Recall” is a term that often refers to the use of the Viterbi algorithm, but can also be used to describe any algorithm that is used during the model testing phase, after the models have already been trained. During real-time recall of the models, HMMs trained on all tasks of interest (reaching for a particular object) are arbitrated based upon an algorithm to determine the “winning model,” or which model best describes the streaming data.

Example feature vectors include subsets of the pose vector, which provides point of resolution (POR) data, a 4x4 homogeneous transform matrix representing the commanded position and orientation of the back of the robot’s hand decomposed into position (x-y-z) and orientation (roll-pitch-yaw). Euclidean distances to the objects of interest being reached for can be used to form the feature vector as well. In addition to feature selection, the tradeoffs, advantages, and disadvantages of applying different recall methods were studied in detail, including their optimizations. Concerning the feature vectors, initially we considered looking only at Y and Yaw, which were discovered to provide good discrimination between the different types of trajectories to distinct objects. However, there were some problems with inconsistent initial conditions across multiple tele-operation sessions and multiple operators that biased the final error statistics. Optimization has not completely resolved this problem, which may in part be due to the small size of the validation sets.

We’d also like to determine whether we can maximize the robustness of our final error statistics (% false alarms and missed detections) to minor variations in the experimental setup. In doing so, we’ve found that processing and testing new datasets based upon HMM prediction models trained on previous sessions are not sufficiently robust to changes in the experimental setup to yield reasonable error statistics. As a result, this gives us incentive to converge to a hybrid solution between the Bayesian approach and the deterministic approach that incorporates the best features of both. The first step in this process is to study and test a new approach that takes advantage of the spatial characterization of the trajectories rather than their time dependence. As such, we become much more reliant on distance to the objects of interest as not only an element within the feature vector, but as a method for discretizing the trajectory space, as one would discretize pixels in a video sequence.

Converged Method

Our previous efforts to model tele-operator movements directly resulted in fragile prediction systems that were sensitive to relatively minor changes in the experimental setup. The models performed well for known experimental configurations but did not generalize to new situations. Modifications to initial conditions, target object position and orientation, or speed of tele-operation would result in a significant increase in missed detections or false positives. It became clear that massive amounts of data would have to be collected in order to train enough models to sufficiently cover the tele-operator's working area. Additionally, it was unclear how well the models would adapt to new tele-operators and how the models could be extended to deal with non-stationary coordinate frames. With this in mind, we have started to explore an alternate approach which attempts to model manipulator approach trajectories in a manner which is invariant to tele-operator speed as well as object position and orientation.

The new method, which we call the approach manifold method, takes an object-centric view of an approaching manipulator. Whereas previous approaches attempted to answer the question "Which object is the tele-operator reaching for?” this approach asks each object in the scene "Do you think the tele-operator is reaching for you?" Central to this approach is the idea that there are a limited number of ways an object can be grasped and that as the manipulator moves towards a target object the trajectory will eventually fall into the approach manifold. Intuitively, the manifold is shaped somewhat like a funnel with the narrow end at the object's grasp site. As the distance between manipulator and object decreases, the range of possible position and orientation values collapses. The approach manifolds are object dependent and further constraints can be added depending on scene construction. As an example, a sphere floating in mid-air can be approached from any direction, so it is possible that any spherical object is the target as long as the manipulator maintains an open pose and is moving in the general direction towards the sphere. The approach trajectory for a drill lying on a table would be considerably more constrained. At a distance of 30cm, one would expect the manipulator to be above the table, moving toward the object, and be open or opening. At a distance of 10cm, one would expect the manipulator to be directly behind the drill handle, hand open, with index finger extended. Because hand pose values are parameterized on distance rather than time, tele-operator speed is irrelevant.

Because the full manifold is defined over many dimensions (X, Y, Z, Roll, Pitch, Yaw – and could include things like hand shape), it is somewhat difficult to visualize. In figures 4 and 5 we show a histogram based distribution on two of the axis, Roll and X, for the horizontal handrail target. In the Roll distribution there is a multi-modal distribution at greater distance, which collapses to a single clean distribution as the hand gets closer to the object. The X distribution is interesting because is stays tightly focused over all distances, which is not surprising because X is the major contributor to the distance metric.

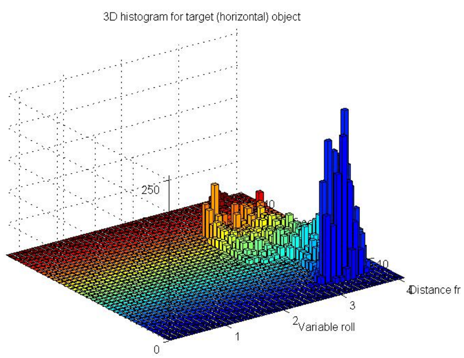

Figure 4 - Distribution of Roll values versus distance.

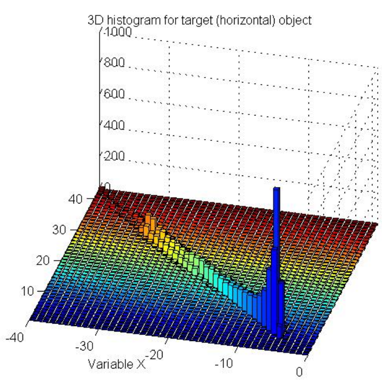

Figure 5 - X versus distance

This approach is still under development, but has already shown robust results. In its current implementation the manifolds are built by grouping together all trial data for a certain target object. Then, for discretized distances from the hand to the object, the mean and variance of each variable of the hand pose is calculated. During runtime recall, object models are selected based upon the object types recognized by the computer vision system. Then, the hand pose is transformed into each object's local coordinate system and compared to the statistics for the object model. Each object is scored based how well the trajectory matches the object manifold. Scores above a heuristically determined threshold are considered candidates for grasping. All candidate objects then calculate a confidence score based on their raw score and the stability of the prediction over time, weighted by linear distance to the manipulator.

We have started to explore replacing this binning approach with a more general mixture of gaussians to describe the manifold. Yet, early experiments indicate that performance may actually degrade in this more general approach. As is often the case, it may be that the simple approach is superior. The current approach is producing good results with no false positives. It has also shown itself to be robust to changes in handrail position and novel experimental setups not encountered in the training data.

Results

As noted above, we have not yet fully implemented the Approach Manifold Method (AMM), and already we are getting some good results. Since this is still under development, we have not yet performed rigorous variational testing. However, not only are the results good on their own terms (100% correct predictions), they also compare favorably against previous results using the optimized HMM's.

Table 1 shows the results of three different tests with collection dates of 6/16 and 10/7. In all these results, the models were trained on other data sets and then validated with either the 6/16 or the 10/7 data set. The Avg. Time Delta metric is the number of seconds from when the classification was made until the handrail was grasped. Thus, the larger the value the better the algorithm did. For example, if the Delta were zero that would mean the classification was made only after the operator had made the entire gesture and was grabbing the handrail. The tests shown in Table 1 are as follows:

HMM: The Optimized Hidden Markov Models described in the Stochastic section, and in earlier papers, is tested against the 6/16 data set.

AMM: The Approach Manifold Method, using binning to approximate a distribution.

	HMM 6/16	AMM 6/16	AMM 10/7
Total Trials	28	28	20
Classified Correct	26	28	20
Failed to Classify	2	0	0
Classified Wrong	0	0	0
Avg. Time Delta	6.9	7.2	5

Table 1: Results Metrics

As can be seen from this table, the new Approach Manifold Method does a better job of classifying the trials, and does so faster than the previous HMM method. What the chart doesn't show is even more impressive. A special dataset was collected where the handrail orientations were reversed: the vertical and horizontal handrails positions were swapped. The models were trained on earlier data, and then shown this reversed data and they were still able to correctly classify all the trials. Since one of the goals of the Approach Manifold Method was to let the models be more independent of position and orientation, these early results show that we are working in the right direction.

Future Work & Discussion

One feature that all these approaches share is that a single model is not general for any arbitrary location of the object. This makes sense since a trajectory is fundamentally the path *from* some known point in space, *to* some known point in space. That being said, it is clear that the different models being produced are valid for some amount of deviation from the points they were trained on. The approach manifold method explicitly attempts to get away from this limitation and encode the approach manifold of a hand towards the object from any arbitrary location. Yet even in this case there are limitations to how general a single model can be because unusual approaches, while valid, may also look like outlier data and will be given low score. Thus, for all these approaches it can be said that, at some granularity, there is a region for which a specific model is valid. Thus, in order to have a complete solution which could recognize grasp attempts anywhere in the workspace of the robot, one would need a number of models to account for all the possible positions of objects. This is a reasonable approach, though it comes at the expense of requiring large amounts of example data. One thing we would like to look at in the future is how to automatically generate appropriate models for any arbitrary object location by having the models be parameterizable by the object location. The approach manifold approach is already taking steps in the direction of location independence by looking at relative positions between the object and the hand.

This greater independence from initial conditions is important for the next phase of our project. Until now the overall experimental setup has been fairly static – the robot has been mounted to the floor, so the objects were generally in well known location relative to it. Next we will be moving to a mobile base for the Robonaut, so we will not be able to make a strong assumption about the object positions relative to the robot.

Another common first thought for achieving position independence is to create a path planner which can generate paths to any arbitrary location. The problem with this approach is that it is not clear that a path planner will generate a trajectory that looks like what a human would decide to do. The goal here is not optimal path generation, but rather the goal is interpreting human gesture. Thus, what we would like to do is to develop enough models of human motion that we can start automatically generating motion models that mimic human gestures.

One of the philosophical bases represented within this work is the notion that we should bootstrap autonomous behaviors through a mentoring relationship between humans and robots. In the case of humanoid robotics, we are capable of symbolically wearing robots by tele-operating them as a means to exemplify desired behaviors. To this end, we are endorsing that one of the goals of researchers in humanoid robotic automation should be to develop systems that automate the process of automation, rather than developing the automation itself. This does not mean that we should not use available prior information (such as inverse kinematic models), but that whenever we are faced with a decision as to how to proceed, we take our cues from human behavior.

This leads us back to one of our original questions: Is it possible to automatically generate autonomous behavior by modeling human action? An important, and unanswered, question is: given two different models which both predict the human action equally well, will one of these models be better at generating autonomous control? What are the properties of a good generative model? And, finally, how can you verify that a model is robust enough that its autonomous control will be safe and reliable? This is a long term research agenda, which, if successful, could have a huge impact on lowering the barrier to the creation of autonomous control.

Our current and future activities are focused upon principles consumed and derived from research in neurophysiology. This does not mean that we need a computer or algorithm based upon sodium ion channels or spiking neurons, but rather that we are interested in the underlying information theoretic principles involved in, among other things, robust pattern recognition and pattern generation. Currently we are developing the methods necessary for closing the loop between sensory and motor control within a Bayesian framework.

We would like to close by thanking the members of the Dexterous Robotics Laboratory at Johnson Space Center for their hard work and ongoing support of this research. Without them, we would have no data.

References

Bluethmann, W.; Ambrose, R.; Diftler, M.; Askew, S.; Huber, E.; Goza, M.; Rehnmark, F.; Lovchik, C.; and Magruder, D. “Robonaut: A robot designed to work with humans in space.” *Autonomous Robots*, 14(2-3):179–198, 2003.

Bregler, C. “Learning and Recognizing Human Dynamics in Video Sequences” *In Proceedings of IEEE Conf. on Computer Vision and Pattern Recognition,* San Juan, Puerto Rico, June 1997.

Fablet, R., and Bouthemy, P. “Motion recognition using spatio-temporal Random Walks in Sequence of 2D Motion-Related Measurements.” *In Proceedings of IEEE Int. Conf. on Image Processing*, Pages 652-655, Thessalonique, Greece, October 2001.

Makris, D., and Ellis, T. “Spatial and probabilistic modeling of pedestrian behavior,” *In Proceedings of British Machine Vision Conference.* Cardiff, U.K., vol. 2, Sep. 2002, pp. 557-566

Peters, R.; Campbell, C.; Bluethmann, W.; and Huber, E. “Robonaut task learning through teleoperation.” *In Proceedings of IEEE International Conference on Robotics and Automation*, pages 2806–2811. Taipei, Taiwan, September 2003.

Porikli, F., “Trajectory Distance Metric Using Hidden Markov Model Based Representation”, *In Proceedings of European Conference on Computer Vision (ECCV)*, Prague, Czech Republic, May 2004.

Wheeler, K.; Martin, R.; SunSpiral, V.; and Allan, M. “Predictive interfaces for long-distance tele-operations”, *In Proceedings of the 8th International Symposium on Artificial Intelligence, Robotics and Automation in Space*, Munich, Germany, September 2005.

Wheeler K., and Jorgensen C., “Gestures as Input: Neuroelectric Joysticks and Keyboards”, *In Proceedings of IEEE Conf. on Pervasive Computing*, Vol. 2, No. 2, April-June, 2003.

Zhou, Y.; Gong, Y.; and Tao, H. “Background modeling using time dependent Markov random field with image pyramid,” in *Proc. IEEE Motion’05*, Breckenridge, Colorado, January 2005.