Toward Robot Learning of Tool Manipulation from Human Demonstration

Aaron Edsinger and Charles C. Kemp [1]
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts
{*edsinger, cckemp*}*@csail.mit.edu*

***Abstract*— Robots that manipulate everyday tools in unstructured, human settings could more easily work with people and perform tasks that are important to people. Task demonstration could serve as an intuitive way for people to program robots to perform tasks. By focusing on task-relevant features during both the demonstration and the execution of a task, a robot could more robustly emulate the important characteristics of the task and generalize what it has learned. In this paper we describe a method for robot task learning that makes use of the perception and control of the tip of a tool. For this approach, the robot monitors the tool's tip during human use, extracts the trajectory of this task relevant feature, and then manipulates the tool by controlling this feature. We present preliminary results where a humanoid robot learns to clean a flexible hose with a brush. This task is accomplished in an unstructured environment without prior models of the objects or task.**

I. Introduction

Robots that manipulate everyday tools in unstructured, human settings could more easily work with people and perform tasks that are important to people. Task demonstration could serve as an intuitive way for people to program robots to perform tasks. By focusing on task-relevant features during both the demonstration and the execution of a task, a robot could more robustly emulate the important characteristics of the task and generalize what it has learned.

An important type of task relevant feature is the tip of a tool. For a wide variety of human tools, control of the tool's endpoint is sufficient for its use. For example, use of a screwdriver requires precise control of the position and force of the tool blade relative to a screw head but depends little on the details of the tool handle and shaft. Radwin and Haney [21] describe 19 categories of common power and hand tools. Approximately 13 of these tool types include a distal point which can be considered the primary interface between the tool and the world.

Focusing on a task relevant feature, such as the tip of a tool, is advantageous for task learning. In the case of tool use, it emphasizes control of the tool rather than control of the body. This could allow the system to generalize what it has learned across unexpected constraints such as obstacles, since it does not needlessly restrict the robot's posture. It also presents the possibility of generalizing what it has learned across manipulators. For example, a tool could be held by the hand, the foot, or the elbow and still be used to achieve the same task by controlling the tip in the same way.

[1]authors ordered alphabetically

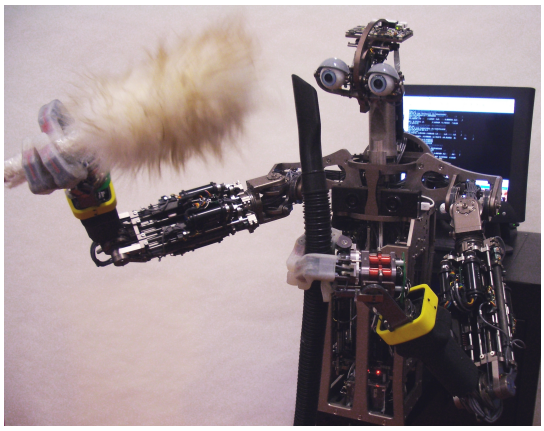

Fig. 1. Domo, the robot with which we obtained our results.

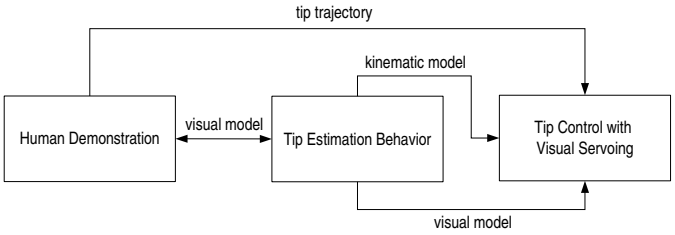

Fig. 2. Overview of the robot task learning framework. The robot watches a human demonstration from which it extracts a tool tip trajectory and potentially a visual model of the tool tip. Once the tools are in the robot's hands, the robot uses its tip estimation behavior to create a visual model of the tip and extend its kinematic model to include the tool. Finally the robot controls the tip to follow the learned trajectory using visual servoing in conjunction with its kinematic model.

We have previously presented a method that combines edge motion and shape to detect the tip of an unmodeled tool and estimate its 3D position with respect to the robot's hand [11] and [12]. In this approach, the robot rotates the tool while using optical flow to find rapidly moving edges that form an approximately semi-circular shape at some scale and position. At each time step, the scale and position with the strongest response serves as a 2D tool tip detection. The robot then finds the 3D position with respect to its hand that best explains these

Fig. 3. We previously demonstrated a method for tool tip detection and control on these tools (hot-glue gun, screwdriver, bottle, electrical plug, paint brush, robot finger, pen, pliers, hammer, and scissors).

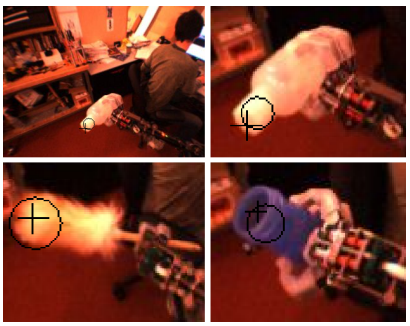

Fig. 4. Our method also works with these tool-like objects that have tips that do not come to a sharp point. The upper left image gives an example of the type of images that were used during detection and estimation. In the other three images the black cross marks the hand annotated object tip location and has a size equivalent to the mean pixel error for prediction of the tips location over a set of images. The black circle is at the tip prediction with a size equal to the average feature.

noisy 2D detections. This method was shown to perform well on the wide variety of tools pictured in Figure 3 and Figure 4.

In this paper, we use this method for tip detection and control as part of a framework for robot task learning, see Figure 2. For this framework, the robot first detects and tracks the tip of a tool while a person demonstrates its use. Next, while moving the tool, the robot detects the tip of the tool, estimates its 3D position with respect to the hand, and builds a visual model of the tip. Finally, the robot controls the tip of the tool to follow a learned trajectory using visual servoing and a kinematic model. Since the trajectory should be relative to the object being acted upon by the tool's tip, the robot also holds an object in its other hand that mostly remains stationary, but has a tip that allows it to be controlled with the same method. We show preliminary results for this framework using the humanoid robot (Figure 1) described in [3].

II. Related Work

Work involving manipulation of task relevant features typically involves fiducial markers or simple objects. Jagersand and Nelson [8] have demonstrated that many tasks can be visually planned and executed using sparse, task relevant, fiducial markers placed on objects. Piater and Grupen [19] showed that task relevant visual features can be learned to assist with grasp preshaping. The work was conducted largely in simulation using planar objects, such as a square and triangle. Pollard and Hodgins [20] have used visual estimates of an object's center of mass and point of contact with a table as task relevant features for object tumbling. While these features allowed a robot to generalize learning across objects, the perception of these features required complex fiducial markers.

Research involving robot tool use often assumes a prior model of the tool or constructs a model using complex perceptual processing. A recent review of robot tool use finds few examples of robots using human tools [22]. NASA has explored the use of human tools with the Robonaut platform, which has used detailed tool templates to successfully guide a standard power drill to fasten a series of lugnuts [7]. Approaches that rely on the registration of detailed models are not likely to efficiently scale to the wide variety of human tools. Williamson [24] demonstrated robot tool use in rhythmic activities such as drumming, sawing, and hammering by exploiting the natural dynamics of the tool and arm. This work required careful setup and tools that were rigidly fixed to the hand.

The robot hand can be thought of as a specialized type of tool, and many researchers have created autonomous methods of visual hand detection through motion including [4] and [18]. These methods localize the hand or arm, but do not select the endpoint of the manipulator in a robust way.

Many researchers have used human demonstration to program robots such as [10], [1], [23], [17], [15], and [7]. These approaches typically use predefined models for the objects or tasks, simplified perception through motion capture, or simplified worlds. Our framework is designed to work with unmodeled objects in unstructured environments using the robot's sensors.

III. Review of Tip Detection and Estimation

In this section we summarize our tool tip detection method, which we describe in detail within [11] and [12]. Our approach consists of two components. First, a tool tip detector finds candidate 2D tool tip positions and sizes within the image while the robot rotates the tool within its grasp. Second, a generative probabilistic model is used to estimate the 3D position of the tool tip within the hand's coordinate system that best accounts for these 2D detections.

A. Tip Detection

We wish to detect the 2D image position and size of the end point of a tool in a general way. This 2D detection can be noisy since the 3D position estimation that follows uses the kinematic model to filter out noise and combine detections from multiple 2D views of the tool.

The 2D tip detector looks for shapes that are moving rapidly while the hand is moving. This ignores points that are not controlled by the hand and highlights points under the hand's control that are far from the hand's center of rotation. Typically tool tips are the most distal component of the tool

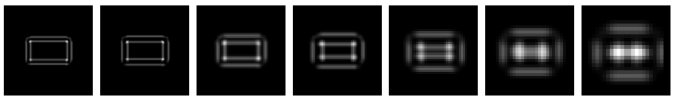

Fig. 5. An example of the raw interest point detector scale-space produced from a rectangle of edges weighted equally with unit motion. Strong responses in the planes correspond with corners, parallel lines, and the ends of the rectangle. The scale represented by the planes increases from left to right.

relative to the hand's center of rotation, and consequently have higher velocity. The hand is also held close to the camera, so projection tends to increase the speed of the tool tip in the image relative to background motion.

As described in detail within [11], the optical flow computation first uses block matching to estimate the most likely motion for each edge along with a 2D covariance matrix that models the matching error around this best match. Next, a global 2D affine motion model is fit to these measurements. Finally, this motion processing results in a weighted edge map, where the weight for each edge is the Mahalanobis distance between the edge's measured motion model and the global motion model.

Next, we use this motion weighted edge map to detect shapes that are moving rapidly and have an approximately convex projection onto the image. To do this we apply a multi-scale interest point operator from [12] to the motion weighted edge map. This algorithm results in a set of histograms, each of which represents the significance of image positions at a particular scale. Locations in these histograms that are locally maximal and have a strong response are likely to correspond with the position and size of the tool tip. When detecting the tip in the robot's hand, we select the position and size associated with the strongest response across all the histograms. When observing a human demonstration we select the top 10 locally maximal positions as tip candidates that are worth tracking.

The multi-scale histograms generated by the detector (Figure 5) have similarities to the output from classic image processing techniques such as the distance transform, medial axis transform, and Hough transform for circles [6]. The detector implicitly assumes that the end of an object will consist of many strongly moving edges that are approximately tangent to a circle at some scale. Consequently, during the robot's tip estimation behavior, the detector will respond strongly to parts of the object that are far from the hand's center of rotation and have approximately convex projections onto the image. We have previously demonstrated that the resulting detections correspond well with human-labeled tips.

B. 3D Estimation

After acquiring the 2D tip detections in a series of images with distinct views, we use the robot's kinematic model to combine these 2D points into a single 3D estimate of the tool tip's position in the hand's coordinate system. To do this, we use the same 3D estimation technique described in [11], which we summarize here. With respect to the hand's coordinate system, $\{H\}$, the camera moves around the hand while the hand and tool tip remain stationary. This is equivalent to a multiple view 3D estimation problem where we wish to estimate the constant 3D position of the tool tip, x_t, with respect to $\{H\}$ (For clarity we will use x_t to denote the tip position in the hand frame ${}^H x_t$). In an ideal situation, only two distinct 2D detections would be necessary to obtain the 3D estimate, as illustrated in Figure 6. However, we have several sources of error, including noise in the detection process and an imperfect kinematic model.

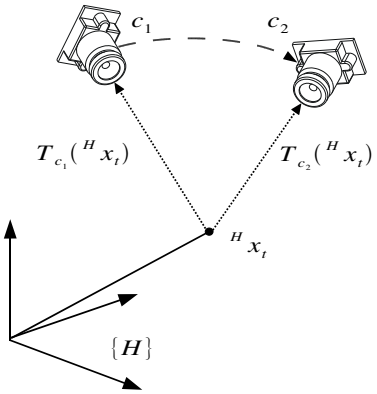

Fig. 6. The geometry of the tool tip 3D estimation problem. With respect to the hand's coordinate system, $\{H\}$, the camera moves around the hand. In an ideal situation, only two distinct 2D detections would be necessary to obtain the 3D estimate. Given two observations with kinematic configurations c_1 and c_2, the tool tip, ${}^H x_t$, appears in the image at $T_{c_1}({}^H x_t)$ and $T_{c_2}({}^H x_t)$.

We estimate x_t by performing maximum likelihood estimation with respect to a generative probabilistic model. We model the conditional probability of a 2D detection at a location d_i in the image i given the true position of the tool tip, x_t, and the robot's configuration during the detection, c_i, with the following mixture of two circular Gaussians,

$$p(d_i|x_t, c_i) = (1-m)\mathcal{N}_t(T_{c_i}(x_t), \sigma_t^2 I)(d_i) + m\mathcal{N}_f(0, \sigma_f^2 I)(d_i). \quad (1)$$

$\mathcal{N}_t$ models the detection error dependent on x_t with a 2D circular Gaussian centered on the true projected location of the tool tip in the image, $T_{c_i}(x_t)$, where T_c is the transformation that projects the position of the tool tip, x_t, onto the image plane given the configuration of the robot, c_i. T_{c_i} is defined by the robot's kinematic model and the pin hole camera model for the robot's calibrated camera. $\mathcal{N}_f$ models false detections across the image that are independent of the location of the tool tip with a 2D Gaussian centered on the image with mean 0 and a large variance σ_f. m is the mixing parameter.

Assuming that the detections over a series of images, i, are independent and identically distributed, and that the position of the tip, x_t, is independent of the series of configurations $c_1 \ldots c_n$, the following expression gives the maximum likelihood estimate for x_t,

$$\widehat{x_t} = \text{Argmax}_{x_t} \left(\log\left(p(x_t)\right) + \sum_i \log\left(p(d_i|x_t, c_i)\right) \right) \quad (2)$$

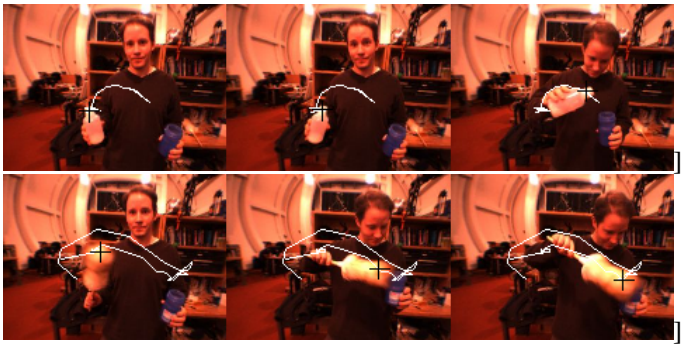

Fig. 7. The output of the tip detector and tracker for two human demonstration sequences. The white curve shows the estimated tip trajectory over the whole sequence and the black cross shows the position of the tip detection in the frame. The top sequence shows a demonstration of pouring with a bottle and the bottom sequence shows a demonstration of brushing.

We define the prior, $p(x_t)$, to be uniform everywhere except at positions inside the robot's body or farther than 1 meter from the center of the hand. We assign these unlikely positions approximately zero probability. We use the Nelder-Mead Simplex algorithm implemented in the open source SciPy scientific library to optimize this cost function [9].

IV. Human Demonstration

During human demonstration of a task the robot detects and tracks the tip of the tool. The resulting trajectory is then used by the robot to control the tool. For this work, we ignore the depth of the tools and assume that the task demonstrated to the robot can be described as a planar activity.

We use the tip detector from Section III-A to select candidate positions and sizes for the tips of the tools being used by the human demonstrator. Typically during tasks, the tip of a tool will tend to be one of the fastest moving, approximately convex shapes within an image. Consequently, the detection method will tend to select regions corresponding with the tool's tip. We can further ensure that the tip will be selected by defining a demonstration protocol that begins with the human rotating the tool so that the tip moves rapidly in the image. More broadly, the human can wave the tool in front of the robot to get the robot's attention. As illustrated in Figure 2, an alternative protocol can be used if the robot will be manipulating the same tool as the demonstrator. In this case, the robot can process the video from the observed demonstration after the robot has built a visual model of the tip. The robot can then use this visual model to detect and track the tool tip within the demonstration video.

After detecting the tool tips, the robot can track them through the video to obtain the trajectory associated with the task. A wide variety of methods from computer vision are applicable to this task. For the preliminary results shown in Figure 7, we follow a procedure similar to our work in [13]. We first collect up to 10 image patches per frame of the demonstration video. These patches correspond to the top 10 tip detections above threshold for each frame. We then use K-Means to cluster the resulting patches using a patch descriptor that primarily consists of color and texture information. For each resulting visual cluster, we find the minimum cost paths that connect tip detections that are members of the cluster, where cost is defined in terms of the velocity required to move between two patches, the visual similarity between the two patches, and the density of the sampling in time. For the preliminary results we present here, we simply select the path that takes place over the longest period of time as the trajectory that describes the task.

V. Visual Modelling and Tracking

After demonstrating the task, the human places the objects within the robot's hands. The robot then estimates the 3D position and size of the tip of each tool using its tip estimation behavior from Section V. The robot also builds a visual model for each tip. The robot uses its kinematic model to predict the position and size of the tip as the robot rotates it, which allows the robot to collect normalized patches that describe the appearance of the tip from a variety of views. (The kinematic tip prediction is not perfect, due to kinematic errors, so the tip detector is combined with the kinematic predictions to select these patches.) Each image patch is normalized by the scale of the detected tip and rotated to a canonical angle, as determined by the projected angle of the vector from the robot's hand to the tool tip. This set of normalized patches can be considered to be a training set that describes the visual appearance of the tip from various viewing angles. For example, we could take patches from the background and train a descriminant based detector. For this paper, we ignore rotations in depth and create a generative Gaussian model of the patch descriptors. This Gaussian model represents the probability of the tool tip generating an image patch with a particular descriptor.

The robot uses this visual model to detect and track the tip while performing visual servoing. When performing visual tracking for visual servoing, the tracker selects a region centered around the kinematically predicted position of the tip in the image. It then normalizes this selected sub-image in scale and rotation, so that the predicted tip size and angle are constant. This normalization lets it efficiently search for the best size and location of the tip by reducing the scales, rotations and positions over which it must search. Given this normalization, a number of options exist for detecting the tip, including convolution. For the preliminary results we present here, we select candidate patches out of this sub-window and evaluate their likelihood given the Gaussian visual descriptor model, their pixel distance to the kinematically predicted location, and their pixel distance to a position based on a linear extrapolation of the previous error vector (The error vector between the actual tip position in the image and the kinematically estimated tip position changes smoothly.).

VI. Control of the Tool in the Image

In the previous section, we presented a task-relevant representation for capturing the human demonstration of tool use. This representation provides a trajectory of the position and orientation of each tool during use, ignoring the details of

the demonstrator's kinematic configuration. We would now like to control similar tools, grasped by the robot, in terms of the position and orientation trajectories generated by the demonstration. In this section we describe a visual servoing approach which combines visual tracking of the tool tip with the kinematic prediction of its appearance. Our approach is a variant of the well studied area of resolved-rate motion control [16] and operational-space control [14].

The detection and estimation process described in Section V produces ${}^{H}x_t$, a 3D estimate of the tip's location within the hand's coordinate frame $\{H\}$. This effectively extends our kinematic model, providing many options for visually controlling the tip. The accuracy of the estimate, and consequently of a strictly feed-forward controller, is dependent on the kinematic and camera calibration. High degree-of-freedom robots such as humanoids will inevitably incur estimation errors.

We compensate for this error by adaptively re-estimating the position of the tip ${}^{H}x_t$ as ${}^{H}x_v$ based on visual feedback from the feature tracker. For a image detection of the tip by the tracker, we first find the ray r in the hand's coordinate frame which passes through the detection pixel and the camera's focal point. We then choose ${}^{H}x_v$ as the closest point on r to ${}^{H}x_t$, providing robustness to tip occlusions and tracker latency.

We can now control the predicted location of ${}^{H}x_v$ in the image. A Jacobian transpose approach allows us to minimize the error between the desired tool pose and the visually re-estimated pose, if the joint angles start close to their final state [2]. For world frame $\{W\}$, the Jacobian, ${}^{W}J^{T}$, is known from the kinematic model and relates hand forces to joint torques as $\tau = {}^{W}J^{T}\,{}^{W}f$. Instead of controlling the arm's joint torque directly, we control the joint angle, and our controller takes the form of $\Delta\theta = \sigma\,{}^{W}J^{T}\,{}^{W}f$ for controller gains σ.

We control the position and orientation of the tip through simulated forces, ${}^{W}f$, created by virtual springs in the hand's coordinate frame $\{H\}$. One virtual spring controls the position of the tip by connecting the estimated position of the tip,${}^{H}x_v$, with the target location, ${}^{H}x_d$. The other virtual spring controls the orientation of the tip by connecting the estimated position of the robot's hand, ${}^{H}x_p$, with a target location ${}^{H}x_o$. The target locations for the tip and the hand are constrained to lie at a fixed depth along the camera's optical axis. The virtual forces acting at the hand are then:

$$ {}^{H}f_t = {}^{H}J^{T}({}^{H}x_v) \begin{bmatrix} ({}^{H}x_d - {}^{H}x_v) & 0 & 0 & 0 \end{bmatrix}^{T} \quad (3) $$

$$ {}^{H}f_p = {}^{H}J^{T}({}^{H}x_p) \begin{bmatrix} ({}^{H}x_o - {}^{H}x_p) & 0 & 0 & 0 \end{bmatrix}^{T}. \quad (4) $$

where ${}^{H}J^{T}({}^{H}x)$ relates forces in $\{H\}$ to a wrench at the hand, as:

$$ {}^{H}J^{T}({}^{H}x) = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix}, P = \begin{bmatrix} 0 & -c & b \\ c & 0 & a \\ -b & a & 0 \end{bmatrix}, \quad (5) $$

for ${}^{H}x = [a, b, c]$. We can transform forces from frame $\{H\}$ to $\{W\}$ through:

$$ {}^{W}_{H}J^{T} = \begin{bmatrix} {}^{W}_{H}R & 0 \\ 0 & {}^{W}_{H}R \end{bmatrix}. \quad (6) $$

giving ${}^{W}f_t = {}^{W}_{H}J^{T}\,{}^{H}f_t$ and ${}^{W}f_p = {}^{W}_{H}J^{T}\,{}^{H}f_p$, where ${}^{W}_{H}R$ is the rotational component of ${}^{W}_{H}T$. A spherical 3 DOF wrist allows decoupling of the control problem into position control by the arm and orientation control by the wrist, giving the controllers:

$$ \Delta\theta_{wrist} = {}^{W}J^{T}\left(\sigma_{twrist}\,{}^{W}f_t + \sigma_{pwrist}\,{}^{W}f_p\right) \quad (7) $$

$$ \Delta\theta_{arm} = {}^{W}J^{T}\left(\sigma_{tarm}\,{}^{W}f_t + \sigma_{parm}\,{}^{W}f_p\right) \quad (8) $$

for controller gains σ. The wrist used in our experiments has only 2 DOF and consequently we must ignore the third joint and assume that the correct orientation is locally achievable with the restricted kinematics. These decoupled controllers bring the estimated tool pose into alignment with a desired pose if the controller is initialized at a joint pose near the final solution.

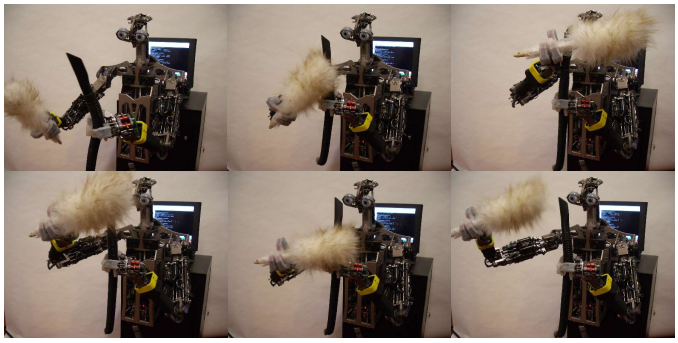

Fig. 8. Video stills of the task execution using a large brush to clean a flexible hose.

VII. Results

We conducted preliminary testing of our method on a brushing task where the robot is to brush a flexible hose held in its hand. The robot, pictured in Figure 1 performing the task, is a 29 DOF upper-torso humanoid name Domo, with 6 DOF in each arm, 4 DOF in each hand, and 9 DOF in the head. Domo is mechanically distinctive in that it incorporates compliance and force sensing throughout its body, as described in [3]. The passive compliance allows local adaptation of the manipulator, and the grasped tool, to interaction forces experienced during the brushing. We can also directly control the robot in terms of forces, allowing us to lower the stiffness of the manipulator during the task when interaction forces are present. These features allow the robot to robustly maintain contact between the brush and the object despite kinematic and perceptual uncertainly. It also suggests complementary control schemes for distinct parts of task execution. Bringing the tool tip into the general vicinity of the point of action can be performed rapidly in an open-loop fashion using the kinematic model. Once the tip is near the point of action, visual servoing can

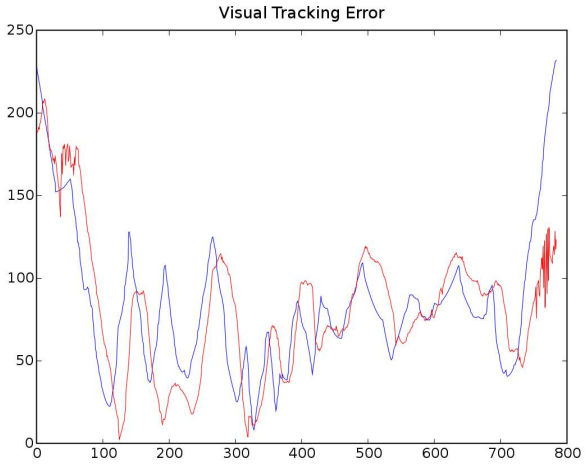

Fig. 9. The visual tracking error of the distance between the two tips during the brushing task execution. The blue indicates the desired inter-tip distance specified by the hand-annotated human demonstration trajectory. The red indicates the inter-tip distance achieved using the model based tracker for visual feedback.

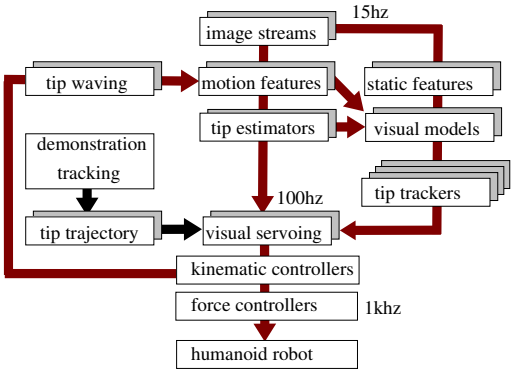

Fig. 10. The system architecture. The computation was distributed across multiple nodes of a Linux cluster. Visual features are computed for the left and right camera image streams. The motion features, generated by waving a grasped object, are combined with a kinematic model to estimate the tip location of the object in the hand. Visual models for the object tips are then computed and used to instantiate a tip tracker for each eye and each hand. The visual servo controller controls the position and orientation of each tip in the image. The trajectories of the two tips are captured during human demonstration and used by the robot to control the grasped objects over time. The real-time data paths are colored red and off-line paths black. Duplicate processes for each eye and arm are indicated with a grey box.

be used to carefully bring the tip into contact with the point of action. Finally, as the tool tip gets close to the point of action, tactile and force sensing coupled with low stiffness control can be used to maintain contact between the tip and the point of action.

The system architecture of our approach is illustrated in Figure 10. We distributed the computation across 10 Pentium based Linux nodes, using the Yarp [5] library for interprocess communication, allowing for an integrated, behavior based implementation of our approach. The detection, tracking and control of the task relevant features are achieved in real-time, while the model estimation and human demonstration tracking are computed off-line. We validated our object tip detection and estimation method for two different shaped bottles and a brush, as shown in Figure 4.

For the task, the robot waves each grasped object in front of the two cameras for about 15 seconds, and an estimate of the objects position in the hand is computed. A visual model for each object and each camera is then computed over the cached data-stream generated during the estimation process. Each visual model automatically instantiates an object tip tracker and the robot begins visual servoing of the two tips according to the task trajectory. The visual position and orientation trajectory of each tip, during demonstration, were previously computed. For our preliminary experiments, we hand annotated the demonstration trajectories.

Figure 8 depicts the robot's execution of the brushing task. The human demonstration of the task involved bringing the tip of the brush to the flexible hose and brushing it using repeated up-and-down and back-and-forth motions. Figure 9 illustrates the performance of the robot in this task, as measured by the ability to control the desired inter-tip distance during the experiment. The robot successfully repeated the brushing task on the flexible hose despite visual occlusion, a natural, cluttered setting, and the flexible movement of the hose. The compliance and force control of the manipulator allowed it to maintain contact throughout most of the brushing.

VIII. Discussion

We have provided preliminary results of our approach on a single task. An important aspect of the approach is the potential to generalize a human demonstrated task to different objects, and we plan to extend our experiments to include multiple tasks and tools. While we can visually servo the tips within the image to offset estimation errors, we have not yet accounted for errors in depth. In the future, we hope to combine the tip trackers' visual features to compute the stereo depth of each tip. We are also interested in extending our approach beyond planar tasks.

References

[1] H. Asada and Y. Asari. The direct teaching of tool manipulation skills via the impedance identification of human motions. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 1269–1274, December 1988.

[2] J. Craig. *Introduction to Robotics*. Addison Wesley, 2 edition, 1989.

[3] Aaron Edsinger-Gonzales and Jeff Weber. Domo: A Force Sensing Humanoid Robot for Manipulation Research. In *Proceedings of the 2004 IEEE International Conference on Humanoid Robots*, Santa Monica, Los Angeles, CA, USA., 2004. IEEE Press.

[4] P. Fitzpatrick, G. Metta, L. Natale, S. Rao, and G. Sandini. Learning About Objects Through Action: Initial Steps Towards Artificial Cognition. In *Proceedings of the 2003 IEEE International Conference on Robotics and Automation (ICRA)*, Taipei, Taiwan, May 2003.

[5] Paul Fitzpatrick and Giorgio Metta. *YARP: Yet Another Robot Platform*. MIT Computer Science Artificial Intelligence Laboratory, http://sourceforge.net/projects/yarp0, 2004.

[6] D. A. Forsyth and Jean Ponce. *Computer Vision: a modern approach*. Prentice Hall, 2002.

[7] E. Huber and K. Baker. Using a hybrid of silhouette and range templates for real-time pose estimation. In *Proceedings of ICRA 2004 IEEE International Conference on Robotics and Automation*, volume 2, pages 1652–1657, 2004.

[8] M. Jagersand and R. Nelson. Visual Space Task Specification, Planning and Control. In *Proceedings of the IEEE International Symposium on Computer Vision*, pages 521–526, 1995.

[9] Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001.

[10] Sing Bing Kang. *Robot Instruction by Human Demonstration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1994.

[11] Charles C. Kemp and Aaron Edsinger. Visual Tool Tip Detection and Position Estimation for Robotic Manipulation of Unknown Human Tools. Technical Report AIM-2005-037, MIT Computer Science and Artificial Intelligence Laboratory, 2005.

[12] Charles C. Kemp and Aaron Edsinger. Robot manipulation of human tools: Autonomous detection and control of task relevant features. In *In Submission to: 5th IEEE International Conference on Development and Learning (ICDL-06)*, Bloomington, Indiana, 2006.

[13] Charles C. Kemp and Aaron Edsinger. What can i control?: The development of visual categories for a robot's body and the world that it influences. In *In Submission to: 5th IEEE International Conference on Development and Learning (ICDL-06)*, Bloomington, Indiana, 2006.

[14] O. Khatib. A unified approach to motion and force control of robot manipulators: The operational space formulation. *International Journal of Robotics and Automation*, 3(1):43–53, 1987.

[15] Nathan Koenig and Maja J. Matarić. Demonstration-based behavior and task learning. In *AAAI Proceedings, AAAI Spring Symposium To Boldy Go Where No Human-Robot Team Has Gone Before*, 2006.

[16] D. Kragic and H. I. Chrisensen. Survey on visual servoing for manipulation. Technical report, Computational Vision and Active Perception Laboratory, 2002.

[17] Y. Kuniyoshi, M. Inaba, and H. Inoue. Learning by watching: Extracting reusable task knowledge from visual observation of human performance. *IEEE Transactions on Robotics and Automation*, 10(6):799–822.

[18] Michel, Gold, and Scassellati. Motion-based robotic self-recognition. In *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[19] Justus H. Piater and Roderic A. Grupen. Learning appearance features to support robotic manipulation. *Cognitive Vision Workshop*, 2002.

[20] N. Pollard and J.K. Hodgins. Generalizing Demonstrated Manipulation Tasks. In *Proceedings of the Workshop on the Algorithmic Foundations of Robotics (WAFR '02)*, December 2002.

[21] R.G. Radwin and J.T. Haney. An ergonomics guide to hand tools. Technical report, American Institutional Hygiene Association, 1996. http://ergo.engr.wisc.edu/pubs.htm.

[22] R St. Amant and A.b Wood. Tool use for autonomous agents. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*, pages 184–189, 2005.

[23] A. Ude and R. Dillmann. Trajectory reconstruction from stereo image sequences for teaching robot paths. In *Proceedings of the 24th International Symposium on Industrial Robots*, pages 407–414, November 1993.

[24] M. Williamson. *Robot Arm Control Exploiting Natural Dynamics*. PhD thesis, Massachusetts Institute of Technology, 1999.