Proceedings of the
International Conference on Intelligent Manipulation and Grasping
May 21–23, 2006, Quebec City, Quebec, Canada
Clément M. Gosselin, editor

Autonomous Detection and Control of Human Tools for Robot Manipulation

CHARLES C. KEMP
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts
cckemp@csail.mit.edu

AARON EDSINGER
Computer Science and Artificial Intelligence Laboratory
Massachusetts Institute of Technology
Cambridge, Massachusetts
edsinger@csail.mit.edu

Abstract: *Robots that can manipulate everyday objects in unstructured, human settings could more easily work with people and perform tasks that are important to people. Ideally, a robot would be able to detect and control task-relevant visual features of an object it has not previously encountered, since the set of objects found within human environments is large and diverse. For a significant set of tasks and tool-like objects, detection and control of the distal end of the object is sufficient for its use. The tips of these objects, such as the end of a screwdriver or the mouth of a bottle, have an approximately convex projection into the visual scene. In this paper we address the problem of visually detecting and controlling the tip of an unknown tool-like object that is rigidly grasped by a robot. We present a multi-scale motion-based feature detector that visually detects a tool tip and we show results in using a series of these 2D detections to produce a 3D estimate of the tool tip's position in the hand's frame. We also describe and evaluate a method for controlling the tool tip's position and orientation in the image.* [1]

1 Introduction

Robots that can manipulate everyday objects in unstructured, human settings could more easily work with people and perform tasks that are important to people. Ideally, a robot would be able to detect and control task-relevant visual features of an object it has not previously encountered, since the set of objects found within human environments is large and diverse. For a significant set of tasks and tool-like objects, detection and control of the distal end of the object is sufficient for its use. The tips of these objects, such as the end of a screwdriver or the mouth of a bottle, have an approximately convex projection into the visual scene. In this paper we address the problem of visually detecting and controlling the tip of an unknown tool-like object that is rigidly grasped by a robot.

[1] This work was sponsored by the NASA Systems Mission Directorate, Technical Development Program under contract 012461-001.

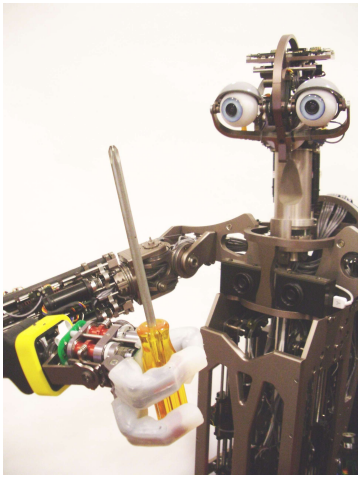

Figure 1: Domo, the robot with which we obtained our results.

We decompose this general problem into the detection of task relevant visual features, the estimation of the pose or other state variables associated with these visual features, and the control of these features. This decomposition has the advantage that it immediately focuses the robot's resources on task relevant aspects of the object, rather than attempting to reconstruct potentially irrelevant complexities in the appearance of the object.

One example of a 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 tool blade relative to a screw head but depends little on the details of the tool handle and shaft. Radwin and Haney [1996] describes 19 categories of common power and hand tools. Approximately 13 of these tool types feature a distal point on the tool which can be considered the primary interface between the tool and the world.

Figure 2: We previously demonstrated the approach on these tools (hot-glue gun, screwdriver, bottle, electrical plug, paint brush, robot finger, pen, pliers, hammer, and scissors).

We have previously presented a method that uses the maximum point of optical flow to detect the tip of an unmodeled tool and estimate its 3D position with respect to the robot's hand [Kemp and Edsinger, 2005]. In this approach, the robot rotates the tool while using optical flow to detect the most rapidly moving image points. It then finds the 3D position with respect to its hand that best explains these noisy 2D detections. The method was shown to perform well on the wide variety of tools pictured in Figure 2. However, the detector was specialized for tools with a sharp tip, limiting the type of objects that could be used.

In this paper, we extend this work in two important ways. First,we present a new multi-scale motion-based feature detector that incorporates shape information. This detector performs well on objects that do not have a sharp point, allowing us to expand our notion of the tip of an object to include such items as a bottle with a wide mouth, a cup, and a brush. The bottle and the cup are not tools in a traditional sense, yet they still have a tip or endpoint that is of primary importance during control. We show that this new feature detector significantly outperforms our previous method on these three objects. We also show that the estimated position and scale of the tip can be used to extract visual features associated with the tip in a manner similar to visual interest point operators. Second, we describe a method for control of the position and orientation of the tool in the image given an estimate of the tip location in the hand's coordinate frame. We show results from the humanoid robot (Figure 1) described in Edsinger-Gonzales and Weber [2004], using an integrated behavior system that first performs tip detection and estimation, and then uses open-loop control to servo the tool in the image to a target location and orientation.

We start by discussing related work in Section 2 and then review the tool tip detection method of Kemp and Edsinger [2005] in Section 3. Next, in Sections 4 and 5, we describe methods for tip detection and control. We conclude with experimental results and a discussion of our approach as applied to three different objects.

2 Related Work

Research involving robot tool use often assumes a prior model of the tool or constructs a model using complex perceptual processing. Industrial robot arms typically use specialized, well-modeled tools that are rigidly mounted using exchangeable end-effectors [Kurfess, 2005]. For example, [Ruf et al., 1997] has demonstrated a real-time system that can visually localize the tool end of a manipulator using a polyhedral model of the tool. A recent review of robot tool use by St. Amant and Wood [2005] fails to find significant examples of robots using human tools outside of work at NASA. NASA has explored the use of human tools by robots with the Robonaut platform [Bluethmann et al., 2004]. They used a detailed set of tool templates combined with stereo depth information to successfully guide a standard power drill to fasten a series of lugnuts [Huber and Baker, 2004]. These approaches are not likely to scale to the wide variety of human tools since they depend on detailed models.

In the work of Brooks [1999], perception is directly coupled to action in the form of modular behaviors that eschew complex intermediate representations. Our method relates to this work in three ways. First, the robot's action is used to simplify the perceptual problem. Second, the method directly detects the tip of the tool without requiring an initial segmentation of the tool or reconstruction of its shape. Third, our approach is suitable for implementation as a real-time modular behavior.

The robot hand can be considered as a specialized type of tool, and many researchers have created autonomous methods of visual hand detection through motion. Fitzpatrick and Metta [Fitzpatrick et al., 2003] used image differencing to detect ballistic motion and optic-flow to detect periodic motion of the robot hand. For the case of image differencing they also detected the tip of the hand by selecting the motion pixel closest to the top of the image. Natale [2004] applied image differencing for detection of periodic hand motion with a known frequency, while Arsenio and Fitzpatrick [2003] used the periodic motion of tracked points. Michel et. al. used image differencing to find motion that is coincident with the robot's body motion [Michel et al., 2004]. Kemp [2005] combined the motion model described in Section 3 with a wearable system to detect the hand of the wearer and learn a kinematic model. These methods localize the hand or arm, but do not select the endpoint of the manipulator in a robust way.

With respect to the computer vision literature, our tip detector is a form of spatio-temporal interest point operator that gives the position and scale that are likely to correspond with the moving tool tip [Laptev, 2005]. The multi-scale histograms generated by the detector have similarities to the output from classic image processing techniques such as the distance transform, medial axis transform, and hough transform for circles [Forsyth and Ponce, 2002], all of which can be viewed in terms of wave fronts that start at the edges and propagate away from the edges, intersecting one another at significant locations, see Figure 4.

In our work, we use our knowledge of how the robot's hand rotates while holding the tool to make 3D estimations about the

location of the tool tip. This relates to methods for 3D scanning in which objects are placed on a rotating platform in front of a single camera [Fitzgibbon et al., 1998]. These methods, however, typically rely on a well modeled background to cleanly segment the object, simple platform motion, and occlusion free views of the object. More generally, our estimation technique relates to the well-studied area of 3D estimation from multiple views [Hartley and Zisserman, 2004].

The tool tip can be viewed as an extension of the robot hand by an additional rigid link with an unknown pose. A wide variety of control techniques can then applied to the tool. We chose a variant of the operational-space control method [Khatib, 1987] because of its simplicity. A large literature exists for visually servoing a robot hand to an object [Kragic and Chrisensen, 2002] and our method is a natural extension this literature.

3 Review of Basic Tip Detection and Estimation

In this section we summarize the basic tool tip detection method, which we describe in detail within Kemp and Edsinger [2005]. Our approach consists of two components. First, a tool tip detector finds candidate 2D tool tip positions 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.

3.1 Tip Detection

We wish to detect the 2D image position 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 points 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 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.

In our initial work, the tool tip detector returned the location of the edge pixel with the most significant motion relative to a global motion model. In this paper, we use the same optical flow algorithm to compute the significance of an edge's motion, but perform multi-scale processing on a motion-weighted edge map to detect the tool tip.

As described in detail within Kemp and Edsinger [2005], 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, the significance of the motion for each edge is computed as the Mahalanobis distance between the edge's measured motion model and the global motion model. This motion measurement incorporates both the magnitude of the edge's motion and the uncertainty of the measurement.

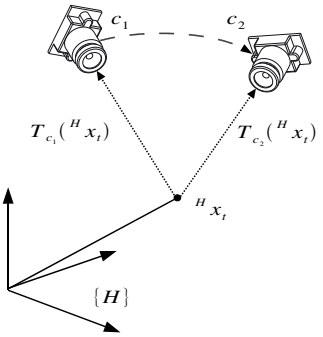

Figure 3: The geometry of the tool tip 3D estimation problem. With respect to the hand's coordinate system, $\{H\}$, the camera moves around the hand. In an ideal situation, only two distinct 2D detections would be necessary to obtain the 3D estimate. Given two observations with kinematic configurations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

3.2 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 Kemp and Edsinger [2005], 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 3. However, we have several sources of error, including noise in the detection process and an imperfect kinematic model.

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

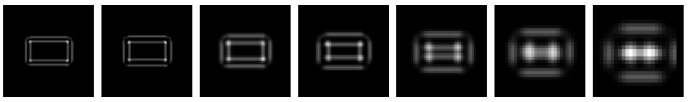

Figure 4: An example of the raw interest point detector scale-space produced from a rectangle of edges weighted equally with unit motion. Strong responses in the planes correspond with corners, parallel lines, and the ends of the rectangle.

the 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 $c_1 \dots c_n$, the following expression gives the maximum likelihood estimate for x_t,

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

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 [Jones et al., 2001] to optimize this cost function.

4 Interest Point Detection

In our original approach we modeled the tip of a tool as a single point within the image. Here we extend this approach by modeling the tip of a tool as occupying a circular area of some radius. In this section we describe this extension, which has better performance on several tools with tips that do not come to a sharp point. Since this new estimate includes the spatial extent of the tip, it also facilitates the use of visual features that describe the appearance of the tip over this spatial extent. For example, given the position and radius we can collect appropriately scaled image patches, see Figure 10.

With respect to our goal of detecting the tip of a tool, this 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, 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 into the image.

The input to the interest point detector consists of a set of weighted edges, e_i, where each edge i consists of a weight, w_i, an image location, x_i, and an angle, θ_i. For this paper, we use a Canny edge detector to produce edge locations and orientations, to which we assign weights that are equal to the estimated motion, where we use the same motion measurement as described within Kemp and Edsinger [2005]. In a manner similar to a Hough transform for circles [Forsyth and Ponce, 2002], each edge votes on locations in a scale-space that correspond with the centers of the coarse circular regions the edge borders. For each edge, we add two weighted votes to the appropriate bin locations at each integer scale s.

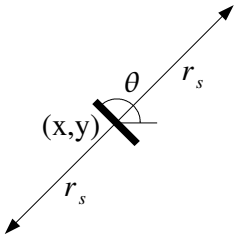

Figure 5: This figure depicts the approximate locations in the image of the two votes at scale s cast by an edge with orientation θ and position (x, y).

As depicted in Figure 5, within the original image coordinates the two votes are approximately at a distance r_s from the edge's location and are located in positions orthogonal to the edge's length. We assume that the angle θ_i denotes the direction of the edge's length and is in the range $[-\frac{\pi}{2}, \frac{\pi}{2})$, so that no distinction is made between the two sides of the edge.

For each scale s there is a 2D histogram that accumulates votes for interest points. The planar discretization of these histograms is determined by the integer bin length l_s, which is set with respect to the discretization of the scale-space over scale, $l_s = \lceil \beta(r_{s+0.5} - r_{s-0.5}) \rceil$, where β is a scalar constant that is typically close to 1.

We define r_s such that r_{s+1} is a constant multiple of r_s, where s ranges from 1 to c inclusive. We also define r_s to be between r_{max} and r_{min} inclusive, so that

$$r_s = \exp\left(\frac{\log(r_{max}) - \log(r_{min})}{c-1}(s-1) + \log(r_{min})\right) \quad (3)$$

and

$$\frac{r_{s+1}}{r_s} = \exp\left(\frac{\log(r_{max}) - \log(r_{min})}{c-1}\right). \quad (4)$$

Setting r_{min} and r_{max} determines the volume of the scale-space that will be analyzed, while c determines the resolution at which the scale-space will be sampled. Higher values of c result in the scale-space being sampled at higher resolution in both scale and location, since c determines the number of planes and l_s depends on the spacing between the planes. Alternatively, we can specify a desired resolution, $\frac{r_{s+1}}{r_s}$, and find a value for c that closely approximates this resolution with

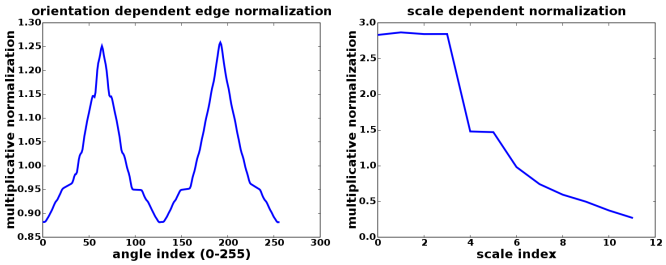

Figure 6: Examples of the two calibrated normalization functions. The left function weights edge points by their angle and the right function weights each plane of the scale space.

$$c = \text{round}(\log_{\frac{r_{s+1}}{r_s}}(\frac{r_{max}}{r_{min}}) + 1). \quad (5)$$

We compute the bin indices, (b_x, b_y), for the 2D histogram at scale s with

$$b_s(x, \theta) = \text{round}(\frac{1}{l_s}(x + r_s \begin{bmatrix}\cos(\theta + \frac{\pi}{2}) \\ \sin(\theta + \frac{\pi}{2})\end{bmatrix})), \quad (6)$$

which adds a vector of length r_s to the edge position x and then scales and quantizes the result to find the appropriate bin in the histogram.

Algorithmically, we now iterate through the edges, adding their weighted contributions to the appropriate bins. We can write the equation for the resulting interest point detection maps, m_s, using delta functions, δ, so that

$$m_s(u) = \sum_i w_i(\delta(u - b_s(x_i, \theta_i)) + \delta(u - b_s(x_i, \theta_i + \pi))), \quad (7)$$

where $\delta(x) = \begin{cases} 1 \text{ if } (x_x = 0) \wedge (x_y = 0) \\ 0 \text{ otherwise} \end{cases}$.

In order to soften the effects of our block discretization, we low-pass filter each 2D histogram, m_s, with a separable, truncated, FIR Gaussian, which is approximately equal to giving each edge a Gaussian vote distribution, since

$$G \star m_s = \sum_i w_i(G(u - b_s(x_i, \theta_i)) + G(u - b_s(x_i, \theta_i + \pi))), \quad (8)$$

where G is an ideal Gaussian. This is also approximately equal to blurring the weighted edge map by scale varying Gaussians, or blurring the scale-space volume across scale.

4.1 Calibration

Ideally, the values of corresponding interest points resulting from a shape would be invariant to translation, scaling, and rotation of the shape. We introduce two scalar functions n_s and n_θ to reduce scale dependent variations and angle dependent variations respectively. These functions are incorporated as follows:

$$m_s(u) = n_s \sum_i n_{\theta_i} w_i(G(u - b_s(x_i, \theta_i)) + G(u - b_s(x_i, \theta_i + \pi))). \quad (9)$$

The values for these two functions are determined empirically using a rotating half-plane as a calibration pattern. The half-plane results in the scale-invariant shape of a straight edge, which allows us to simultaneously find normalization values for all scales without explicitly scaling the input shape. We first find a function for n_θ and then find a function for n_s that together make the average values of the interest points equivalent over rotations of this calibration pattern. A more natural and varied set of calibration images might result in better estimates for these functions, but this method is simple and effective.

4.2 Filtering the Interest Points

Once we have these various interest point maps, we would like to filter out points that are uninteresting and select points that are likely to correspond with salient regions. In general, we do this by selecting points that are local maxima in the scale space, thresholding the points by their value, and looking at curvature and shape information to filter out interest points that result from a single strong edge. For this paper, however, we only require the best point in the scale space, so we simply select the point with the strongest response as the most likely position and scale of the moving tool tip.

5 Control of the Tool in the Image

In this section we describe a method for controlling the tool's position and orientation in the image. The approach is a variant of the well studied area of resolved-rate motion control [Kragic and Chrisensen, 2002] and operational-space control [Khatib, 1987]. The robot used in this paper, seen in Figure 1, has 4 DOF in the arm and 2 DOF in the wrist.

The robot uses Series Elastic Actuators [Pratt and Williamson, 1995] at each joint. These actuators allow safe, passively compliant force control of the joint through a simple PID control loop around the deflection of a spring. A secondary PID loop around the joint angle commands a desired torque to the force controller.

A kinematic model of the 7 DOF in the robot's head and 6 DOF in the robot's arm is known. We assume that the camera's intrinsic parameters are known and that the radial distortion in the image has been removed. The transform between world coordinates and image coordinates, ${}^W_I T$, is known. We also assume that the head remains fixed and therefore ${}^W_I T$ is constant.

A Jacobian transpose approach will allow us to minimize the error between the desired tool pose and the estimated pose if the joint angles start close to their final state [Craig, 1989]. The Ja-

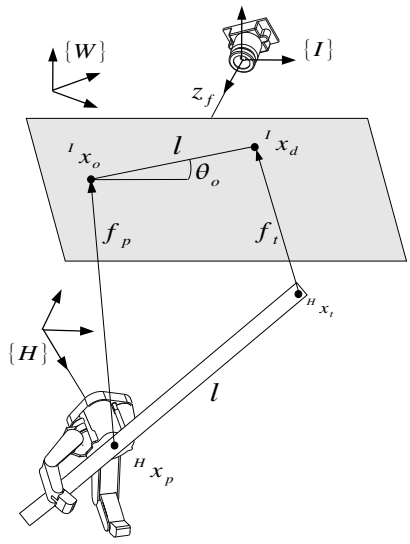

Figure 7: The geometry of the tool tip control problem. The tip estimation process computes the most likely 3D tool tip position, ${}^H x_t$, in the hand's coordinate frame $\{H\}$. The desired tip location is ${}^I x_d$ in the image frame $\{I\}$ with fixed depth z_f. The desired tool orientation, θ_o, corresponds to location ${}^I x_o$ in the image given tool length l. A virtual spring between the tip ${}^H x_t$ and ${}^H x_d$ generates force ${}^H f_t = {}^H x_d - {}^H x_t$. A virtual spring between the palm ${}^H x_p$ and ${}^H x_o$ generates force ${}^H f_p = {}^H x_o - {}^H x_p$. A Jacobian transpose method is used move the tool to the target position and orientation with a control law of the form $\Delta\Theta = \sigma\ {}^W J^T \left({}^W f_t + {}^W f_p \right)$ for controller gains σ .

cobian, ${}^W J^T$, 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 in order to limit latency dependent instability, and our controller takes the form of $\Delta\theta = \sigma\ {}^W J^T\ {}^W$ for controller gains σ.

The geometry of the control problem is illustrated in Figure 7. The target pose for the tool is constrained to be at a fixed depth z_f along the optical axis. The tip estimation process computes the most likely 3D tool tip position, ${}^H x_t$, in the hand's coordinate frame $\{H\}$. The desired tip location ${}^I x_d$ in the image frame $\{I\}$ has hand coordinates ${}^H x_d = {}^H_I T\ {}^I x_d$. In order to control the tool's orientation, we assume that the tool is grasped such that it passes through the palm at a fixed location ${}^H x_p$. The desired tool orientation, θ_o, corresponds to a location ${}^I x_o$ in the image given tool length l.

The Jacobian transpose approach can be thought of as applying virtual springs between the tool and its desired pose in the image plane. We can transform virtual forces on the tool into the hand's coordinate frame. For a point in the hand's frame, ${}^H x = [a, b, c]$, the Jacobian relating force ${}^H f$ at ${}^H x$ to forces at the hand frame $\{H\}$ is

$$ {}^H J^T({}^H x) = \begin{bmatrix} I & 0 \\ P & I \end{bmatrix}, P = \begin{bmatrix} 0 & -c & b \\ c & 0 & a \\ -b & a & 0 \end{bmatrix}. \quad (10) $$

The virtual forces acting at the hand are then:

$$ {}^H f_t = {}^H J^T({}^H x_t) \begin{bmatrix} \left({}^H x_d - {}^H x_t\right) & 0 & 0 & 0 \end{bmatrix}^T \quad (11) $$

$$ {}^H f_p = {}^H J^T({}^H x_p) \begin{bmatrix} \left({}^H x_o - {}^H x_p\right) & 0 & 0 & 0 \end{bmatrix}^T \quad (12) $$

. We can transform forces from frame $\{H\}$ to $\{W\}$ by:

$$ {}^W_H J^T = \begin{bmatrix} {}^W_H T & 0 \\ 0 & {}^W_H T \end{bmatrix}. \quad (13) $$

giving us ${}^W f_t = {}^W_H J^T\ {}^H f_t$ and ${}^W f_p = {}^W_H J^T\ {}^H f_p$. 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:

$$ \Delta\theta_{wrist} = {}^W J^T \left(\sigma_{twrist}\ {}^W f_t + \sigma_{pwrist}\ {}^W f_p \right) \quad (14) $$

$$ \Delta\theta_{arm} = {}^W J^T \left(\sigma_{tarm}\ {}^W f_t + \sigma_{parm}\ {}^W f_p \right) \quad (15) $$

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

6 Results

We validated the method on a bottle, a cup, and a brush, as pictured in Figure 8. The items were chosen for their varying tip size and length. The feature detector, estimator, and controller were integrated into a real-time behavior module for the robot. The detection algorithm runs at $15Hz$ on a $3GHz$ Pentium computer without optimization. When the tool is placed in the robot's hand, it automatically generates a short sequence of tool motion of about 200 samples over 15 seconds. Each detection and kinematic configuration is logged and then batch processed by the estimator. The estimated tip location, ${}^H x_t$, is passed to the tool pose controller and it servos the tool to a potentially time-varying location and orientation in the image.

For each tool we compare the multi-scale detector of this paper to the edge-motion detector. Figure 8 shows the mean prediction error, as measured by the tool tip projection into the image, for the two detectors. The multi-scale detector significantly improves the predicted location for these three objects that have

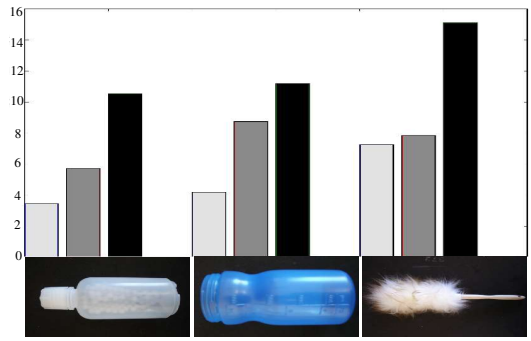

Figure 8: The mean prediction error, in pixels, for each tool. The 3D tool pose in the hand is estimated in three ways, using: the hand labelled tool tips [left bar], feature-based interest points [middle bar], and the edge pixel with the maximum motion [right bar]. The error is computed by projecting the predicted tool tip onto the image plane for each sample in the test set and comparing the projection to the hand labelled tip. The left bar is an indication of baseline errors in the kinematic and camera calibrations.

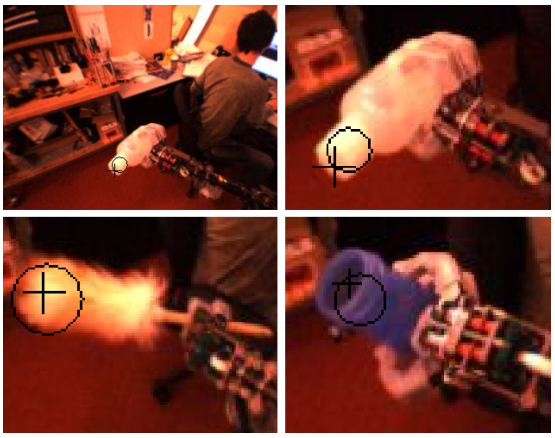

Figure 9: The upper left image gives an example of the images used during estimation. The movement of the person in the background serves as a source of noise. In the other three images the black cross shows the hand annotated location and has a size equivalent to the mean pixel error for prediction over the test set. The black circle is at the tip prediction with a size equal to the average feature scale. These circles have a radius of 8.96, 9.38, and 14.96 pixels [clockwise] respectively.

large, broad tips. Consequently, it extends the notion of a tool tip beyond sharply pointed objects.

The multi-scale detector enables online modeling of the tip. Figure 9 show the average estimated tip scale for each tool, which demonstrates the ability of the detector to appropriately extract the size of the tool tip. Figure 10 illustrates the ability to construct a pose and scale normalized visual model of the tip.

For each tool, the tip position was estimated and the controller was commanded to servo the tip to the center of the image with a horizontal orientation. Figure 11 shows the typical errors for the controller relative to the projection of the estimated tip location into the image, and relative to the actual hand labelled tip location. The controller errors are low with respect to the predicted tip location but are larger with respect to the hand-labelled location due to the reliance of the method on precise kinematic calibration.

Our work affords many avenues for further exploration. The reliable prediction of the tool tip in the visual scene allows us to model the tool's visual features. A model could be used to visually track the tip and could allow the robot to actively test and observe the endpoint during interactions with the world. It could also be used to more precisely control the tool by visual servoing.

We have described a general method for visual manipulation of human tools rigidly held by a robot. Our method extends the notion of a tool to include objects with broad tips and is robust to tools of unknown size and shape. It is a step towards robots that autonomously learn to manipulate novel, unmodeled objects in human-centric environments.

References

A. Arsenio and P. Fitzpatrick. Exploiting cross-modal rhythm for robot perception of objects. In *Proceedings of the Second International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, December 2003.

W. Bluethmann, R. Ambrose, A. Fagg, M. Rosenstein, R. Platt, R. Grupen, C. Brezeal, A. Brooks, A. Lockerd, R. Peters, O. Jenkins, M. Mataric, and M. Bugajska. Building an Autonomous Humanoid Tool User. In *Proceedings of the 2004 IEEE International Conference on Humanoid Robots*, Santa Monica, Los Angeles, CA, USA., 2004. IEEE Press.

Rodney A. Brooks. *Cambrian Intelligence*. MIT Press, Cambridge, MA, 1999.

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

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.

Andrew W. Fitzgibbon, Geoff Cross, and Andrew Zisserman. Automatic 3d model construction for turn-table sequences. In R. Koch and L. VanGool, editors, *Proceedings of SMILE Workshop on Structure from Multiple Images in Large Scale*

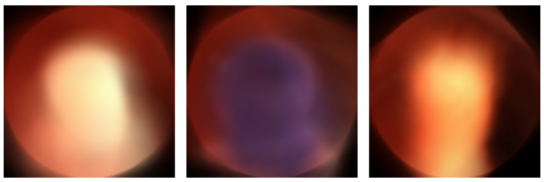

Figure 10: These results from averaging clipped and normalized image patches of each of the three tips illustrate the potential for visually modeling the tips using the tip predictions. Each square image patch was centered on the best tip detection relative to the tip prediction. The patch was then scaled and rotated to a canonical pose using the detected scale and visual orientation of the tip. Prior to collecting these patches, the detected positions, scales, and angles were smoothed over time to improve the estimates at each frame. Distinct appearances due to rotations in depth were averaged together for this illustration.

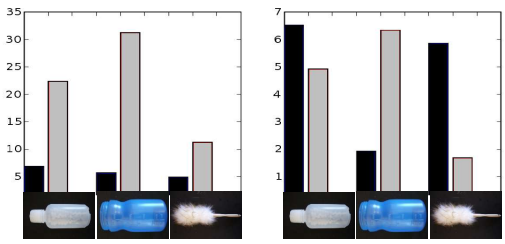

Figure 11: The controller error for the tip position [left,pixels] and orientation [right,degrees] using the multi-scale detector. The black bar indicates the error as measured by the projection of the predicted tip position and orientation into the image. The grey bar indicates the hand measured position and orientation error. For each tool, the tip position was estimated and the controller was commanded to servo the tip to the center of the image with a horizontal orientation.

Environments, volume 1506 of *Lecture Notes in Computer Science*, pages 154–170. Springer Verlag, June 1998.

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.

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

R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

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.

Eric Jones, Travis Oliphant, Pearu Peterson, et al. SciPy: Open source scientific tools for Python, 2001. URL `http://www.scipy.org/`.

Charles C. Kemp. *A Wearable System that Learns a Kinematic Model and Finds Structure in Everyday Manipulation by using Absolute Orientation Sensors and a Camera*. PhD thesis, Massachusetts Institute of Technology, May 2005.

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.

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.

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

Thomas Kurfess, editor. *Robotics and Automation Handbook*. CRC Press, Boca Raton, Florida, 2005.

I. Laptev. On space-time interest points. *Int. J. Computer Vision*, 64(2):107–123, 2005.

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

Lorenzo Natale. *Linking Action to Perception in a Humanoid Robot: A Developmental Approach to Grasping*. PhD thesis, LIRA-Lab, DIST, University of Genoa, 2004.

G. Pratt and M. Williamson. Series Elastic Actuators. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS-95)*, volume 1, pages 399–406, Pittsburg, PA, July 1995.

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.

A. Ruf, M. Tonko, R. Horaud, and H. Nagel. Visual tracking of an end-effector by adaptive kinematic prediction. In *Intelligent Robots and Systems IROS'97*, 1997.

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.