To Appear: Int'l Conf. on Advanced Robotics,
Tokyo, Japan, November 1-2, 1993

A Direct Approach to Vision Guided Manipulation

Marcos Salganicoff

GRASP Laboratory
Department of Computer and
Information Science
University of Pennsylvania
Philadelphia, PA, USA
sal@grip.cis.upenn.edu

Giorgio Metta Andrea Oddera
Giulio Sandini

Laboratory for Integrated Advanced
Robotics (LIRA - Lab)
Department of Communication,
Computer and Systems Science
University of Genoa
Via Opera Pia 11A - I16145
Genoa, Italy
giulio@vision.dist.unige.it

Abstract—**This paper describes a method for robotic manipulation that uses direct image-space calculation of optical flow information for continuous real-time control of manipulative actions. State variables derived from optical flow measurements are described. The resulting approach is advantageous since it robustifies the system to changes in optical parameters and also simplifies the implementation needed to succeed in the task execution. Two reference tasks and their corresponding experiments are described: the insertion of a pen into a "cap" (the capping experiment) and the rotational point-contact pushing of an object of unknown shape, mass and friction to a specified goal point in the image-space.**

I. INTRODUCTION

The visual system of an agent, either natural or artificial, has to cope with motion in at least two ways: it should be able to detect, measure and interpret the motion of external objects, and it must be able to use dynamic visual information to control, plan and coordinate its own motion.

The emphasis of this paper is in the use of vision for the continuous control of manipulative actions with the aim of understanding and implementing purposive and qualitative control mechanisms based on optical flow.

The relevance of this *continuous* use of visual information is evident in at least three important situations:

- In *learning motor actions*. In this case vision provides the only independent way of measuring motor performance and consequently of tuning motor programs, as has been demonstrated convincingly in numerous works by Held [6]. It also seems obvious that skill learning such as is necessary in the use of everyday tools such as forks, pens, computer keyboards or cars would benefit from its use.
- During the execution of *exploratory actions*. In this case vision is used to monitor the execution of motor actions in order to detect unexpected events (such as collisions) or to perform accurately.
- During *interaction with unconstrained (e.g. moving) objects*, or whenever the accuracy of proprioceptive information is not sufficient to carry out a specific task. Examples of this kind can be found in most manipulative tasks requiring dual-arm manipulation or fine, dexterous manipulation.

This research was supported by ESPRIT Projects FIRST and SECOND, the Special Projects on Robotics of the Italian National Council of Research, an NSF Postdoctoral Associateship for MS (CDA-9211136), and by NSF/ESPRIT IRI-9303980

The qualitative approach is motivated by the intention to control the robot arm without relying on shape measure, depth estimate, and calibration of the camera coordinate system with respect to the arm. This approach, which has been studied in the past with reference to vision, touch and force sensing [2, 1, 5, 14], has been tested in two different experimental situations: the insertion of a pen into an independently moving cap, and the point-contact pushing of an object of unknown shape, mass distribution and friction towards a goal point in the workspace. By point-contact pushing, we mean the pusher remains within the friction cone of the contact (i.e. only a rotational degree of freedom exists at the pusher/object contact point; this is enforced by notching the object at the contact point in the experiments).

The specific goal of this paper is to try to identify and extract the simplest (in term of computational requirements) visual cues allowing the system to accomplish the task.

II. THE CAPPING EXPERIMENT

In the capping task selected, the goal of the hand-eye system is to control the motion of the arm so that the tip of the pen correctly docks with the cap. If this action is performed open-loop, then the generation of a precise trajectory is required prior to the initiation of motion. This limits its applicability to constrained situations (e.g. no external disturbances) and requires a relatively high accuracy in the estimation of joint angles and the robot's kinematic parameters. On the contrary, if the action is performed closed-loop using vision as a feedback signal, it is necessary to constantly monitor the trajectory of the moving hand (and the tip of the pen) with respect to the cap. Which are the *visual measures* used to control the action? A possible solution is to measure the position of the tip of the pen with respect to the cap in a 4D space (x, y, z, t). The alternative hypothesis proposed in this paper is the use of a direct solution based on the measure of optical flow fields and disparity without the need of explicit 4D measures.

Considering the capping action, the goal of the visually-guided controller is to keep the tip of the pen on an ideal linear trajectory connecting, at each instant of time, the cap with the tip. The projection of this 3D trajectory on

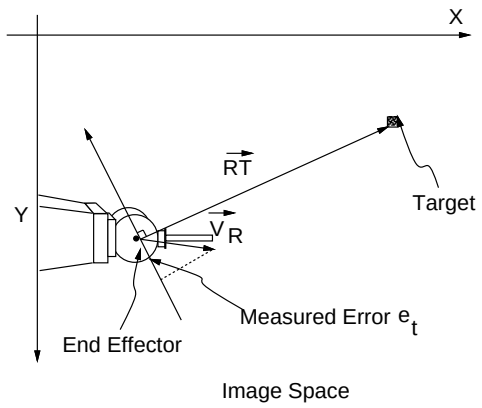


Fig. 1: The coordinate frames and measures for the pen capping experiment.

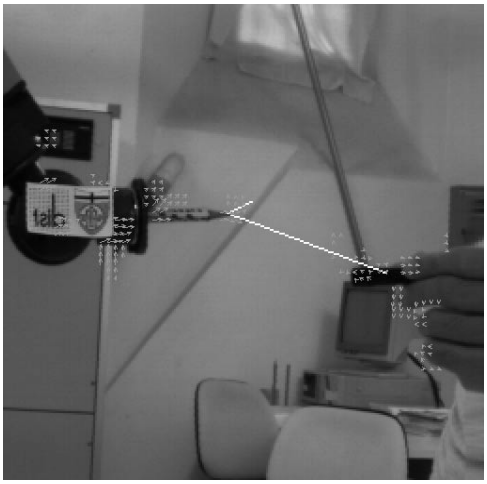


Fig. 2: Optical flow computed on one of the images of the sequence

the image plane represents the 2D trajectory that must be followed by the *image* of the moving end-effector in order to dock with the cap (see Figures 1 and 2). This is the first constraint: keep the image of the end-effector moving along this ideal 2D trajectory. Reasoning in terms of optical flow vectors, this constraint can be achieved by minimizing the component of the flow field perpendicular to this ideal trajectory.

Of course, depth must be also considered. However no explicit depth measure is necessary if the control action is performed such as to maintain the tip of the pen and the cap on the same "disparity plane" [2]

In summary, the paradigm we are proposing is based on the minimization of the following two measures:

- Component of optical flow perpendicular to the 2D trajectory connecting the docking objects.

[2]in case the fixation point is on the cap, the end-point of this ideal 3D trajectory is on the zero-disparity plane and the control action is to minimize the absolute value of disparity or, in other words, to drive the moving hand toward (or along) the zero-disparity plane. For a reasonably small camera baseline, this zero disparity surface (the horopter), while not perfectly planar, is relatively flat

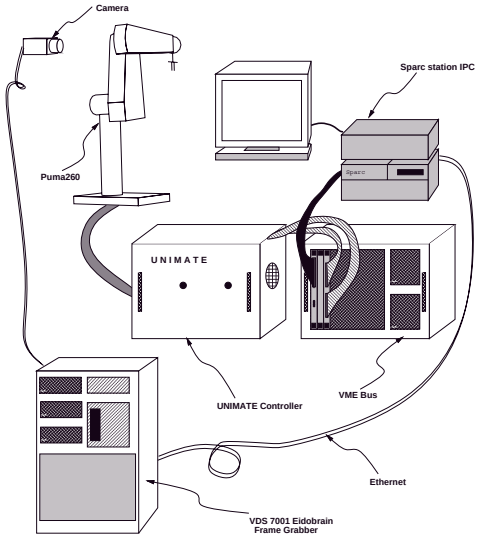


Fig. 3: The experimental system consists of a PUMA260/Controller, SparcstationRunning RCCL, and a VDS EidoBrain Image Acquisition/Processing Computer.

- Disparity.

A. Assumptions

Due to limited computational power it has not been possible to perform both the optical flow computation and the disparity measures in real-time. Therefore, the constraint imposed in the experiment reported is that the robot holding the pen and the cap lie on the same disparity plane and the controller is only required to continuously control the forward and the up/down motion of the tip of the pen.

A second constraint has been used in order to locate the position of the cap on the image by using a threshold on the gray levels and by restricting the position of the cap to the rightmost part of the image.

It is worth noting, however, that the segmentation of the moving arm, the location of the tip of the pen and the control action are based only on optical flow measures thus allowing the action to be performed on arbitrary static backgrounds. The position of the cap may vary during the action in order to test the continuous nature of the control action.

B. Experimental Set-up

The experimental setup consisted of a manipulatory and a perceptual component (see Figure 3). The manipulatory component consists of a Unimation PUMA260 Robot, Unimation Controller, a SparcStation IPC running RCCL [8], Sbus/VME Mapper, and software which allows for high-speed communication between the Sparc IPC and the Unimation Controller. The perceptual component consists of a VDS EidoBrain 7001 Image Acquisition and Processing system and CCD Camera. Communication is accomplished using TCP/IP sockets, which are adequate for the .65 second update intervals. A manipulation server process exists on the Sparc which servos the most recent rate commands from the VDS at 28 msec intervals and takes care of communication protocols.

C. *Visual processing*

The optical flow is computed on each image with a resolution of 44×40 pixels, which represents a fixed region of interest in a 64×64 subsampling from a 256×256 image. The flow took approximately 650ms to compute. From the flow result the following were extracted:

- Segmentation of the moving hand from the static background by thresholding based on velocity magnitude (note that the background cannot be separated on the basis of static measures).
- Computation of the position of the tip of the pen (the above threshold flow vector location closest to the cap).
- Segmentation of the cap on the basis of gray level (the darkest region in the rightmost part of the image) and the computation of the center of mass of the segmented region which is defined as the goal point.
- Computation of an error measure of the average components of the 2D velocity field perpendicular to the direction connecting the tip of the pen with the cap (e_t).

With the above measures, a proportional/derivative control law can be realized, the error measure is:

$$e_t = \frac{|\vec{V_R} \times \vec{RT}|}{|\vec{V_R}||\vec{RT}|}\text{sgn}(\vec{V_R} \times \vec{RT}) \quad (1)$$

Here $\vec{V_R}$ is the average of all the flow vectors associated with the end-effector and $\vec{RT}$ is the vector connecting the end-effector and the target (see Figure 1). The function sgn returns -1 if its argument is negative and 1 otherwise and the control law is a proportional/derivative:

$$v_y = k_p e + k_d(e_t - e_{t-1}). \quad (2)$$

with v_x held constant.

Here, the velocities, v_x and v_y, are expressed in the robot coordinate system.

D. *Pen Capping Results*

The results of the experiment are shown in Figures 4, 5. In this particular experiment the capping action was performed in about 22 sec. (or 35 frames, 650ms/frame). The origin of the cartesian coordinates is positioned on the cap and the x axis is directed toward the tip of the pen at the beginning of the capping action. It is evident that the y component of the end-effector and target eventually converge and towards the end of the trajectory the end-effector is actually moving along the desired linear trajectory. The x component is monotonically decreasing with an approximately constant velocity (see Figure 5). The y component of visual velocity of the end-effector initially oscillates around the value of 0, and eventually converges, indicating that the servo is attempting to null the perpendicular flow components as desired.

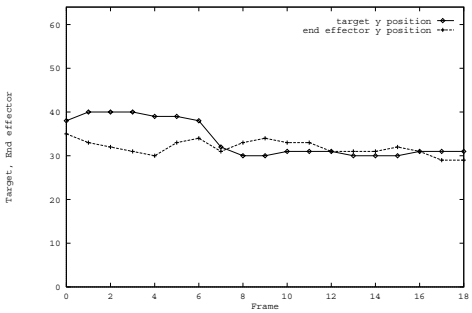


Fig. 4: The Y position of the tip of the pen during the action. Units are in terms of a 64×64 subsampling of the full resolution image

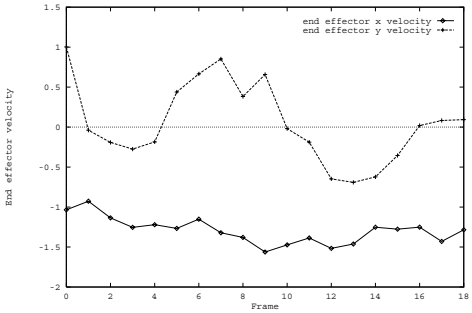


Fig. 5: V_x and V_y components of the tip of the pen during the action

III. THE PUSHING EXPERIMENT

In many situations it is desirable to move an object from one location to another, but the object may be too large to be lifted by a single agent. Two possible solutions exist, either many agents may cooperate in lifting and moving the object [3], or it may be possible for a single agent to push the object instead of lifting it. We explore the pushing case where the contact between robot and the object is single point (see Figure 6) and the pusher remains within the friction cone of the contact (i.e. only a rotational degree of freedom exists at the pusher/object contact point; this is enforced by notching the object at the contact point in the experiments).

Pushing and steering of an object to desired position in the workspace when there is only a point contact between the pusher and the object is a difficult visuomotor control problem since the relationship between the pusher and the object is unstable. Because of this, the object tends to rotate past the pusher if no corrective actions are taken. At the same time, a desired pushing direction must be achieved in order to arrive at the desired point in the robot workspace.

An additional complication is that the object motion resulting from pushing actions is a function of the frictional distribution of the object [11] on its surface of support and the mass distribution of the object, which are difficult to measure using only passive visual perception. These quantities can, in general, only be measured with active perceptual procedures [4, 9]. However, even if these

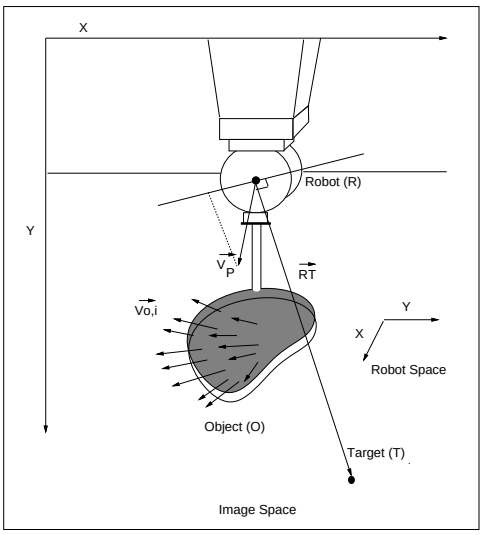


Fig. 6: The pushing task. The pusher and object are connected with a rotation-only point contact, so that the object can rotate relative to the pusher (slip between the pusher and object is prevented by notching the object at the contact point.) The objective is to move the object to the desired point in the image space.

quantities can be estimated, friction is difficult to model analytically because of its non-linear behavior.

Rather than estimate all of the above parameters and utilize an analytic model of friction we develop a simple and direct solution by measuring the effects of pushing actions using optical flow measures and servoing actions to achieve their desired image-space values.

The pushing and sliding manipulation problem has been studied extensively by Mason [10] from an analytical viewpoint, as well as from a learning perspective [11]. Lynch [9] has explored using visual measurements of object reaction to pushing actions in order to explicitly estimate the center of friction of the object. Zrimec [15] implemented a system which generated qualitative models of the effects of pushing actions through experience which were then used for planning.

A. *Steering by Controlled Instability*

Since pushing is an intrinsically unstable process with a point contact, one immediate objective might be to null the rotation of the object relative to the pusher. However, if the only objective is to zero the object's rotation relative to the pusher, then control of steering is impossible, since when this condition is achieved, no directional correction is possible and pusher trajectory is fixed. When pushing an object we desire to null it rotation *only* when the pusher is aligned with the idealized trajectory. When the current pusher trajectory is misaligned, the objective should be the controlled rotation of the object relative to the pusher in order to bring the pusher trajectory in line with the ideal trajectory. This follows because if the pusher is controlling the object at a fixed rotation rate, then the pusher direction must be changing in synchrony with the sequence of new object orientations as the object rotates. This gradual change of pusher direction aligns it with the ideal pushing trajectory $\vec{RT}$. This rotation is a controlled instability, since object rotation is a manifestation of the instability of the task.

In the image space, let $\vec{RT}$ be the vector between the current center of mass of the locations of flow vectors associated with the robot end-effector and the desired target location in the image space, and $\vec{V_P}$ be the average of all vectors associated with the pusher (see Figure 6). The direction and magnitude of desired rotation ω_d of the object is a function of the angle $\theta_{V_P,RT}$ between the pushing direction and the ideal pushing trajectory $\vec{RT}$:

$$\omega_d = -k_s \sin(\theta_{V_P,RT}) \quad (3)$$

$$= -k_s \frac{\mid \vec{V_P} \times \vec{RT} \mid}{\mid \vec{V_P} \mid\mid \vec{RT} \mid} \mathrm{sgn}(\vec{V_P} \times \vec{RT}) \quad (4)$$

Since it is difficult to control high rotation rates in practice, the ω is bounded by putting it through a saturation function:

$$\omega_d^{'} = \begin{cases} \omega_{max} & \text{if } \omega_d > \omega_{max} \\ \omega_{min} & \text{if } \omega_d < \omega_{min} \\ \omega_d & \text{otherwise} \end{cases} \quad (5)$$

This desired rotational rate $\omega_d^{'}$ then provides a reference rotation rate which must be servoed by a second proportional/ derivative control loop which is written

$$v_{y,t+1} = k_p e_t + k_d (e_t - e_{t-1}) \quad (6)$$

where

$$e_t = \hat{\omega_t} - \omega_{d,t}^{'} \quad (7)$$

Here v_y refers to the commanded y velocity of the end-effector in the robot frame, v_x is kept constant, and $\hat{\omega_t}$ is the estimate of the rotational velocity of the object, relative to the pusher, as defined below.

B. *Qualitative Image Space Measures of Rotation*

A useful and relatively reliable measure of the proportion of flow due to rotation is to compute the normalized perpendicular component of flow relative to the current pushing direction. Consider an object moving in the frame of reference of the pusher, and rotating about the contact point between it and the pusher. The velocity of any point p on the object, described by $\vec{p}$ in the reference frame of the pushing point, is the vector addition of a tangential component due purely to rotation, $\vec{V_r} = \omega \times \vec{p}$, and $\vec{V_P}$ which is the translational velocity of the pushing point. If we normalize all flow vectors measured on the object and the velocity of the pusher and compute the average, this provides an indication of the magnitude and direction of the object's rotation relative to the pushing direction and magnitude. Thus, the perpendicular velocity measure is computed by the following formula:

$$\hat{\omega} = \frac{1}{N_O} \sum_{i}^{N_O} \frac{\mid \vec{V_{O,i}} \times \vec{V_P} \mid}{\mid \vec{V_{O,i}} \mid\mid \vec{V_P} \mid} \mathrm{sgn}(\vec{V_{O,i}} \times \vec{V_P}) \quad (8)$$

where $V_{O,i}$ is one of the N_O flow vectors associated with the pushed object by the segmentation process.

C. Assumptions

As stated previously, flow provides a direct method of assessing the stability of the current pusher/object configuration. For the purposes of the analysis we make the assumption that $d >> f$ where d is the distance of object to focal point and f is the focal length, and we assume a narrow field of view, so we can model the imaging process as a scaling and orthogonal projection. This is necessary so that pure translational motion has only a small amount of perspective induced divergence that might fool the rotational measure.

D. Segmentation of Pusher and Object

The figure ground segmentation was accomplished by computing the optical flow [7] with recursive temporal filtering over the incoming image sequence and thresholding the flow vectors based on magnitude. Locations with above threshold optical flow are labelled as foreground and others as background.

Once the foreground has been labelled, the segmentation between the pusher and object must be performed. During a brief calibration motion the manipulator is swept through its pushing workspace with no object present, holding the y-component of the end-effector fixed while the x-axis position is moved in the positive direction. Simultaneously, the end-effector position is tracked in the image space. The robot x-axis positions and their associated image y-axis values are stored and simple linear fit is done to calculate the relation between the two. Later, during the execution of the pushing task, the manipulator position is used to compute the vertical position of the end effector in the image using the linear fit parameters. All flow vectors below the horizontal at this vertical position are associated with the object and vectors above it with the robot. Assuming the object can be held $\pm\frac{\pi}{2}$ of the image y-axis (approximately in front of the pusher) this provides an extremely reliable and simple segmentation method.

E. Pushing Results

Some representative sequences of the systems performance are shown in Figures 7 and 8. Because of the large delay (800ms) in computing the 64×64 flow vectors, the actual trajectory tended to oscillate about the desired trajectory. This was due to the fact the controller gains had to be made large (see Figure 9) in order for the pusher to induce large enough motions on the object relative to the low resolution at which the image was sampled (this also explains why the large features were marked on the object as seen in the image sequences). Faster processing with more rapid hardware will allow higher resolution flow computations with lower latency. Nevertheless, the control was adequate to reach the goal points. When errors occurred, it was generally due to the fact that the object could not turn far enough before the small pushing workspace of the PUMA260 robot was exceeded.

IV. DISCUSSION

It is interesting to note that the pen-capping experiment was quite insensitive to changes in camera parameters. In particular, it was possible to change the focal length (using camera zoom) and to rotate the camera a large amount along its optical axis *between* and *during* task trials with only minor effects on the performance of the task. Changing the camera parameters in this case is essentially equivalent to changing the gains of equation 2.

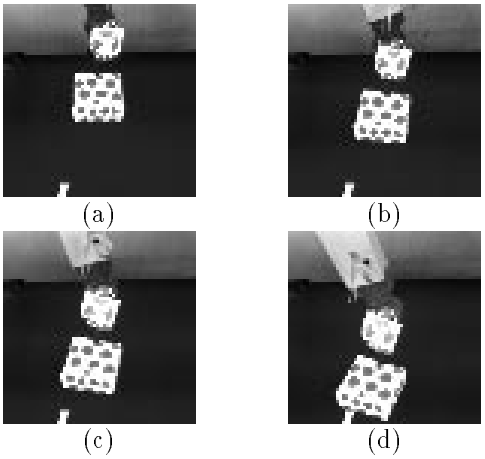


Fig. 7: An image sequence for pushing to a point in the image space to the left. The white target is found by grey-level thresholding.

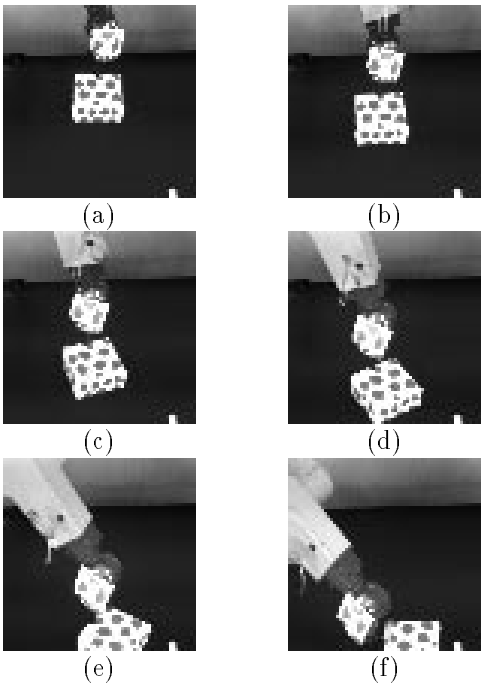
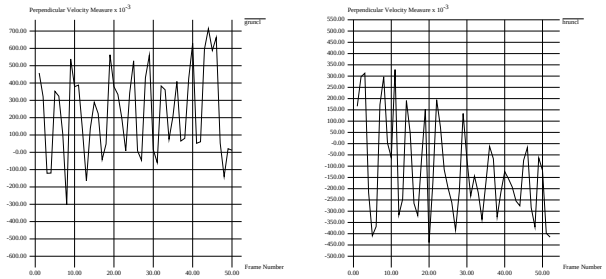


Fig. 8: An image sequence for pushing to a point in the image space to the right.



(a) (b)

Fig. 9: The value of the control variables during the pushing task corresponding to the two image sequences depicted. The commanded rotational velocity is positive in (a) while the actual rotational estimate oscillates significantly, but is almost always positive. Similary, in (b) the commanded rotational is negative and rotational velocity estimate is predominately negative.

However, within limits, as long as the sign of the gains are correct (preventing positive feedback) the servo still achieves the task. This property is quite advantageous, since the practical consequence is that a tedious camera calibration phase is unnecessary.

The utilization of optical flow simplifies the arm/background segmentation problem significantly, assuming a static background. Unlike other approaches for manipulator control in the image space [12] which require identifying and tracking markers on arm joints such as LEDS, or grey level thresholding which is quite sensitive to ambient illumination, flow measures are much more flexible since they do not require explicit tracking.

The next step in the capping experiment would be to incorporate disparity servoing (see section II) to match the disparity of the target. Since the disparity plane is, by definition, orthogonal to the x, y plane of the image, the two tasks can be easily decomposed into independent servo processes, assuming that the velocity in depth is limited so as to not induce a large divergent or convergent optical flow component on the end effector.

The segmentation between pusher and object is currently done in an *ad-hoc* fashion. Other more general approaches for this problems should be developed. In particular, knowledge of the pusher speed and direction might be utilized as constraints for separating flow vectors arising from the object from those of the pusher.

In pushing, more direct rotational measures could be tried in the task. Initially, in this work, the object's center of rotation and rotational velocity was computed using a least-squared technique [5]. However, this measure proved unreliable due to a number of factors. In particular the separation of translational and rotational flow components is a particularly difficult problem, especially when the translational component dominates. Since a relevant state-variable in the task is the rotation of the object relative to the pusher, when translation components tend to dominate, this biases the rotational estimates.

The pushing approach can benefit from a learning component, since in the current implementation the controller gains are fixed, but the dynamics of the pushing task vary as a function of the mass distribution and frictional properties of the object being pushed. It is beneficial to rapidly adapt to changes in parameters as different objects are pushed. An example of the use of a memory-based algorithm to rapidly learn a forward model of the effects of pushing actions can be found in a companion paper [13].

V. CONCLUSION

We have demonstrated the utility of optical flow as a direct and reliable qualitative measure for control of manipulator actions in real-time for capping (insertion) and pushing tasks that have many important and immediate applications. A major benefit of the approach is that it is possible to achieve good performance without extensive camera calibration or excessive control of the environment such as illumination, flat backgrounds or tracking fixtures on the manipulators. In pushing tasks, the approach does not require precise knowledge about the shape, mass and frictional properties of the object being pushed.

REFERENCES

[1] P. Allen. *Object Recognition Using Vision and Touch.* Ph.D. Dissertation, University of Pennsylvania, 1985.

[2] R. Bajcsy and C. Tsikos. Perception via manipulation. *Proc. of the Int. Symp. & Exposition on Robots*, 237–244, 1988. November 6-10.

[3] Ruzena Bajcsy, Richard Paul, Xiaoping Yun, and Vijay Kumar. A multiagent system for intelligent material handling. In *Proceedings of the International Conference on Advanced Robotics*, pages 18–23, Pisa, Italy, 1991.

[4] M. Campos and R. K. Bajcsy. *A robotic haptic system architecture.* Technical Report MS-CIS-90-51, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1990.

[5] F. Gandolfo, G. Sandini, and M. Tistarelli. Towards vision guided manipulation. In *Fifth International Conference on Advanced Robotics*, pages 661–667, 1991.

[6] R. Held and J. Bauer. Visually guided reaching in infant monkeys after restricted rearing. *Science*, 155:718–720, 1970.

[7] B.P. Horn and B.G. Schunk. Determining optical flow. *Artificial Intelligence*, 17, 1981.

[8] John Lloyd. *Implementation of a Robot Control Development Environment.* Master's thesis, McGill University, Montreal, Canada, 1986.

[9] K. Lynch. Estimating the friction parameters of pushed objects. In *Proc. of the 1993 IEEE/RSJ Int'l Conference on Intelligent Robots and Systems*, pages 186–193, 1993.

[10] M. T. Mason. Mechanics and planning of manipulator pushing operations. *International Journal of Robotics Research*, 5(3):53–71, 1986.

[11] M. T. Mason, A. D. Christiansen, and T. M. Mitchell. Experiments in robot learning. In *Proceedings of the Sixth International Workshop on Machine Learning*, pages 141–145, Morgan-Kauffman, 1989.

[12] B. Mel. *Connectionist robot motion planning: A neurally inspired approach to visually guided reaching.* Academic Press, San Diego, CA, 1991.

[13] M. Salganicoff, G. Metta, A. Oddera, and G. Sandini. A vision-based learning method for pushing manipulation. In *AAAI Fall Symposium Series: Machine Learning in Vision: What Why and How?*, Raleigh, N.C., 1993. to appear.

[14] C. J. Tsikos and R. K. Bajcsy. *Redundant Multi-Modal Integration of Machine Vision and Programmable Mechanical Manipulation for Scene Segmentation.* Technical Report MS-CIS-88-41, University of Pennsylvania, Dept. of Computer and Information Science, Philadelphia, Pa., 1988.

[15] T. Zrimec. *Towards Autonomous Learning of a Behavior by a Robot.* PhD thesis, Dept. of Electrical Engineering and Computer Science, University of Ljubljana, Ljubljana, Slovenia, 1990.