Proceedings of the 1992 IEEE
International Conference on Robotics and Automation
Nice, France - May 1992

Trajectory Filtering and Prediction for Automated Tracking and Grasping of a Moving Object

Peter K. Allen, Aleksandar Timcenko, Billibon Yoshimi and Paul Michelman *
Department of Computer Science, Columbia University, NY, NY 10027

Abstract

Most robotic grasping tasks assume a stationary or fixed object. In this paper, we explore the requirements for grasping a moving object. This task requires proper coordination between at least 3 separate subsystems: real-time vision sensing, trajectory-planning/arm-control, and grasp planning. As with humans, our system first visually tracks the object's 3-D position. Because the object is in motion, this must be done in real-time to coordinate the motion of the robotic arm as it tracks the object. The vision system is used to feed an arm control algorithm that plans a trajectory. The arm control algorithm is implemented in two steps: 1) filtering and prediction, and 2) kinematic transformation computation. Once the trajectory of the object is tracked, the hand must intercept the object to actually grasp it. We present experimental results in which a moving model train is tracked, stably grasped, and picked up by the system.

1 INTRODUCTION

The focus of our work is to achieve a high level of interaction between a real-time vision system that is capable of tracking moving objects in 3-D and a robot arm that contains a dexterous hand that can be used to intercept, grasp and pick up a moving object. We are interested in exploring the interplay of hand-eye coordination for dynamic grasping tasks such as grasping of parts on a moving conveyor system, assembly of articulated parts or for grasping from a mobile robotic system. Coordination between an organism's sensing modalities and motor control system is a hallmark of intelligent behavior, and we are pursuing the goal of building an integrated sensing and actuation system that can operate in dynamic as opposed to static environments. The algorithms we have developed that relate sensing to actuation are quite general and applicable to a variety of complex robotic tasks that require visual feedback for arm and hand control.

The system we have built addresses three distinct problems in robotic hand-eye coordination for grasping moving objects: fast computation of 3-D motion parameters from vision, predictive control of a moving robotic arm to track a moving object, and grasp planning. The system is able to operate at approximately human arm movement rates, using visual feedback to track, stably grasp, and pickup a moving object.

The system consists of two fixed cameras that can image a scene containing a moving object (see Figure 1). A PUMA-560 with a parallel jaw gripper attached is used to track the object with the goal of stably grasping and picking up the object as it moves. The system operates as follows:

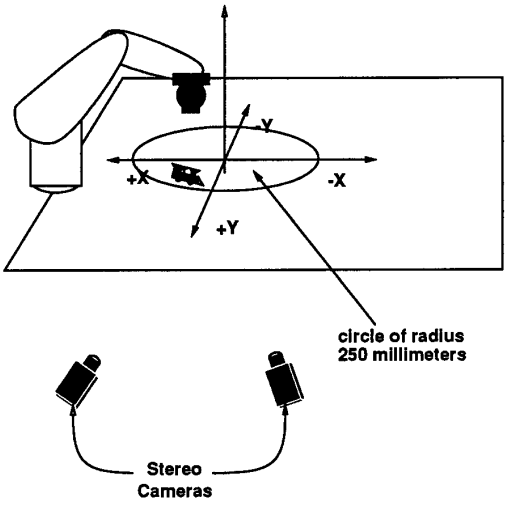

Figure 1: Tracking Grasping System

1. The imaging system performs a stereoscopic optic-flow calculation at each pixel in the image. From these optic-flow fields, a motion energy profile is obtained that forms the basis for a triangulation that can recover the 3-D position of a moving object at video rates.
2. The 3-D position of the moving object computed by step 1 is initially smoothed to remove sensor noise, and a non-linear filter is used to recover the correct trajectory parameters which can be used for forward prediction, and the updated position is sent to the trajectory-planner/arm-control system.
3. The trajectory planner updates the joint level servos of the arm via kinematic transform equations. An additional fixed gain filter is used to provide servo-level control in case of missed or delayed communication from the vision and filtering system.
4. Once tracking is stable, the system commands the arm to intercept the moving object and the hand is used to stably grasp the object and pick it up.

The following sections of the paper describe each of these subsystems in detail along with experimental results. Space does not

*This work was supported in part by DARPA contract N00039-84-C-0165, NSF grants DMC-86-05065, DCI-86-08845, CCR-86-12709, IRI-86-57151, IRI-88-1319, North American Philips Laboratories, Siemens Corporation and Rockwell Inc.

0-8186-2720-4/92 $3.00 ©1992 IEEE

allow us to reference the numerous previous efforts in tracking, control and grasp planning that have influenced our work. We refer the reader to our technical report [2] for a detailed list of references.

2 VISION SYSTEM

The vision system used in this research is described in detail in [3] and we briefly review the method here. In a visual tracking problem, motion in the imaging system has to be translated into 3-D scene motion. Our approach is to initially compute local optic-flow fields that measure image velocity at each pixel in the image. A variety of techniques for computing optic-flow fields have been used with varying results including matching based techniques [5, 7, 19] gradient based techniques [12, 16, 8] and spatio-temporal energy methods [11, 1]. Optic-flow was chosen as the primitive upon which to base the tracking algorithm since it can be extracted quickly and reliably from our images, and it quantifies actual motion in the scene which we need to detect. We are using 2 fixed cameras that are calibrated with the 3-D scene, but there is no explicit need to use registered (i.e scan-line coherence) cameras. The identical algorithm for extracting optic-flow is run on each camera's image in parallel using the PIPE parallel image processor [14]. Once the motion centroids are known for each camera, they are back-projected into the scene using the camera calibration matrices and triangulated to find the actual 3-D location of the movement. This 3-D position is computed every 1/60th second, but with a processing delay of roughly 100 msec.

3 ROBOTIC ARM CONTROL

The second part of the system is the arm control. The robotic arm has to be controlled in real-time to follow the motion of the object, using the output of the vision system. The raw vision system output is not sufficient as a control parameter since its output is both noisy as well as delayed in time. The control system needs to do the following:

- Filter out the noise with a digital filter
- Predict the position to cope with delays introduced by both vision subsystem and the digital filter
- Perform the kinematic transformations which will map the desired manipulator's tip position from a Cartesian coordinate frame into joint coordinates, and actually perform the movement

Our vision algorithm provides in each sampling instant a position in 3D space as a triplet of Cartesian coordinates (x, y, z). The task of the control algorithm is to smooth and predict ahead the trajectory, thus positioning the robot where the object is during its motion.

A well known and useful solution is the Kalman filter approach, because it successfully performs both smoothing and prediction. However, the assumption the Kalman filter makes is that the noise applied to the system is white. That fact directly depends on the parametrization of the trajectory and, unfortunately in our case, the simplest possible parametrization - Cartesian- does not support this noise model. Our previous work [3] used a variant of this approach and obtained tracking that was smooth but not accurate enough to allow actual grasping of the moving object. Our solution to this problem was to appeal

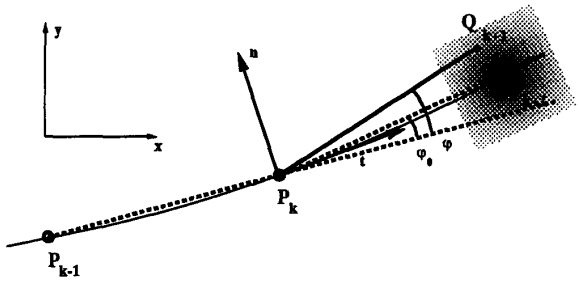

Figure 2: Trajectory: the moving object is in P_{k+1} while the vision computes Q_{k+1}

to a local coordinate system that was able to model the motion and system noise characteristics more accurately, thus producing a more accurate control algorithm.

3.1 The Model of the 3D Motion

The main idea in the trajectory parametrization used in this paper is to describe a point in a *local* coordinate frame, relative to the point from the previous sampling instant, by the triplet of coordinates (s, ϕ, z) where

- s is the length of an arc between two points
- ϕ is the "bending" of the trajectory (see figure 2)
- z is the altitude difference in two consecutive points

Due to the existence of noise, all three coordinates are random variables with certain distributions. We have made the following assumptions, as a result of both reasoning about the vision algorithm and certain necessary simplifications:

- In sampling instant k our object is in point P_k
- In the next sampling instant $k+1$ the object is in P_{k+1} and the point returned by the vision algorithm is Q_{k+1}
- Q_{k+1} is normally distributed around P_{k+1}. The noise can be expressed by its two components, tangential n_t and normal n_n
- n_t and n_n are both zero-mean, with the same dispersion and mutually not correlated. Experimentally, it has been determined that their coefficient of correlation is between 0.1 and 0.2.

Under these assumptions it can be shown [2] that the velocity v and curvature κ are:

$$v = \lim_{T \to 0} s/T \tag{1}$$

$$\kappa = \lim_{T \to 0} \tan \varphi_0 / s_0 \tag{2}$$

where $s_0 = ||P_{k+1} - P_k||$ and $\varphi_0 = \pi - \angle P_{k-1} P_k P_{k+1}$.

What are advantages of such a parametrization? The most obvious one is the simplicity of the prediction task in this framework; all we need is to multiply the velocity $v = s/T$ and the "bending" parameter ϕ by time $\tau > T$ we want to predict ahead. The next advantage is that in order to achieve an accurate prediction, we do not need a high-order model with the mostly heuristic tuning of numerous parameters. The price we have to pay is that *filtering is not straightforward.* It turns out that we cannot just apply a low-pass filter in order to recover a DC

component from s, but rather we need more elaborate approach which takes into account a probabilistic distribution of s.

While this model introduces more complexity than a standard Cartesian model, we will see below that it is more effective in allowing us to accurately predict and smooth our trajectory. The initial experiments with this model separate 3-D space into an XY plane and the Z axis, and addresses these two components of motion separately. However, the method for the XY plane can be extended to include another parameter which will create a full Frenet frame at each instant of time in the trajectory. Our initial experiments (described below) tracked a planar curve, allowing us to use this simplification. Motion in the Z direction is tracked with a Cartesian displacement as outlined in [3].

Our model assumes the following coordinate transformation that relates the moving object's coordinate frame at one instant with the next instant in time:

$$\mathsf{Rot}(z, \phi_0) \circ \mathsf{Trans}(x, s) \circ \mathsf{Trans}(z, \triangle z) \tag{3}$$

where Rot and Trans are rotation / translation around / along given axis.

3.2 Probability Distributions of s and ϕ

In this section, we will motivate the choice of model used to recover the parameter values s_0 and φ given the estimate of the arclength s. Let $s = ||Q_{k+1} - P_k||$ be the distance between the object and the next position returned by the vision algorithm. According to figure 2 we have

$$\begin{aligned} s &= \left\| \begin{bmatrix} \cos\varphi_0 & \sin\varphi_0 \\ -\sin\varphi_0 & \cos\varphi_0 \end{bmatrix} \begin{bmatrix} n_t \\ n_n \end{bmatrix} + \begin{bmatrix} s_0 \\ 0 \end{bmatrix} \right\| = \\ &= \sqrt{(n_t' + s_0)^2 + n_n'^2} \end{aligned} \tag{4}$$

where n_t' and n_n' are Gaussian with dispersion σ. According to the definition of the probability distribution, we can write the distribution $F(s)$ as

$$F(s) = \int\int_D \frac{1}{2\pi\sigma^2} \mathrm{e}^{-\frac{1}{2}\left[\left(\frac{t-s_0}{\sigma}\right)^2 + \left(\frac{n}{\sigma}\right)^2\right]} \mathrm{d}t\mathrm{d}n \tag{5}$$

where D is a disk of the radius s.

Now by introducing substitution $t = r\cos\theta$, $n = r\sin\theta$ we get

$$F(s) = \frac{1}{2\pi\sigma^2} \int_0^s r \int_0^{2\pi} \mathrm{e}^{-\frac{1}{2}\left[\left(\frac{r\cos\theta - s_0}{\sigma}\right)^2 + \left(\frac{r\sin\theta}{\sigma}\right)^2\right]} \mathrm{d}\theta\mathrm{d}r \tag{6}$$

Distribution density is given as $f(s) = \frac{\mathrm{d}F(s)}{\mathrm{d}s}$ or after differentiation

$$f(s) = \frac{s\mathrm{e}^{-\frac{s^2+s_0^2}{2}}}{2\pi\sigma^2} \int_0^{2\pi} \mathrm{e}^{-\frac{ss_0}{\sigma^2}\cos\theta} \mathrm{d}\theta \tag{7}$$

The last integral can be expressed by a modified Bessel function $I_0(z)$:

$$f(s) = \frac{s}{\sigma^2} \mathrm{e}^{-\frac{s^2+s_0^2}{2\sigma^2}} I_0\left(\frac{ss_0}{\sigma^2}\right) \tag{8}$$

A graph of $f(s)$ is given in figure 3. Here s_0 is fixed to 1 and σ varies from 0.4 to 1.0. Our job is to recover s_0 given $f(s)$.

It is apparent from the figure 3 that the peak value of $f(s)$ depends on σ, and drifts towards higher values as σ grows. The expectation for s also depends on σ. In particular, we have

$$s_1 = E(s) = \int_0^\infty s f(s) \mathrm{d}s = \sigma u\left(\frac{s_0}{\sigma}\right) \tag{9}$$

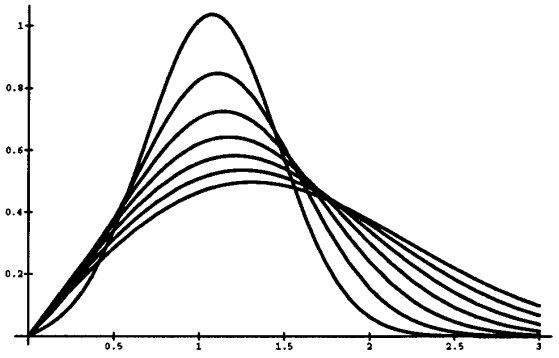

Figure 3: Distribution density $f(s), s_0 = 1, \sigma = 0.4 - 1.0, increment = 0.1$

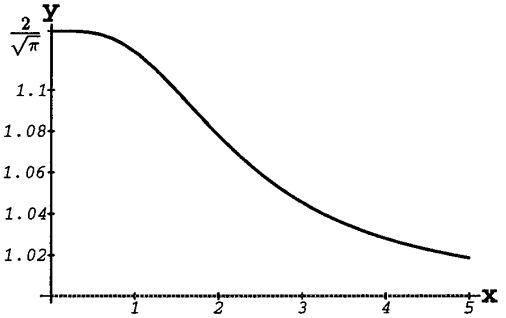

Figure 4: $y = u_1(x)$

where

$$u(x) = \sqrt{\frac{\pi}{2}} \mathrm{e}^{-x^2/4} (I_0(\frac{x^2}{4}) + \frac{x^2}{2}(I_0(\frac{x^2}{4}) + I_1(\frac{x^2}{4}))) \tag{10}$$

Here σ is the constant for the given system and it is related to s_0. In order to estimate σ we will use second-order moment:

$$s_2^2 = E(s^2) = \int_0^\infty s^2 f(s) \mathrm{d}s = s_0^2 + 2\sigma^2 \tag{11}$$

Equations 9 and 11 are derived in [2].

Now by eliminating s_0 from 9 and 11 we have

$$1 = z\, u\left(\frac{\sqrt{p^2 - 2z^2}}{z}\right) \tag{12}$$

where $p = s_2/s_1$ and $z = \sigma/s_1$. Now by setting $x = \frac{\sqrt{p^2-2z^2}}{z}$ we end up with an equation

$$u_1(x) = \frac{\sqrt{x^2+2}}{u(x)} = p \tag{13}$$

Equation 13 relates our known control inputs ($p = s_2/s_1$) to x. We can create a table of values for this function offline, and then by interpolation calculate a value of x given p.

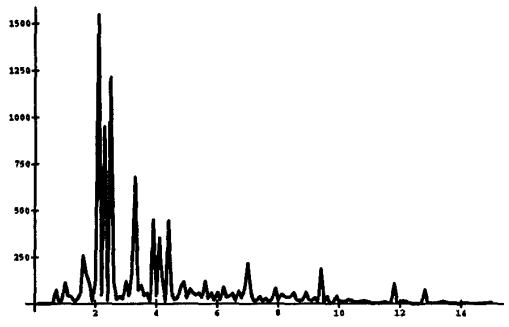

Figure 5: Density of s_1

Let $x_0(p)$ be the solution of 13. Now we can express s_0 and σ as functions of s_1 and s_2 as follows:

$$s_0 = s_2 \frac{x_0\left(\frac{s_2}{s_1}\right)}{\sqrt{2 + x_0\left(\frac{s_2}{s_1}\right)^2}} \tag{14}$$

$$\sigma = s_2 \frac{1}{\sqrt{2 + x_0\left(\frac{s_2}{s_1}\right)^2}} \tag{15}$$

This method requires little on line computation - an interpolation table of values of u_1 is all we need to recover the arclength parameter s_0. Figure 5 is the experimentally measured density of s_1 taken from the triangulated optic-flow fields. This distribution's resemblance to figure 3 (the theoretical density) is clear.

To find the bending parameter ϕ_0, we use the same technique as for the distribution of s, and we get the following formula:

$$f(\phi) = \frac{\cos(\phi - \phi_0)}{\sqrt{2\pi}k} e^{-\frac{\sin^2(\phi-\phi_0)}{2k^2}} \tag{16}$$

where $k = \sigma/s_0$ and $\phi - \phi_0 \in (-\pi/2, \pi/2)$. It is obvious that f is symmetric around ϕ_0, which also means that the expectation $E\phi = \phi_0$. Hence, we so not need to perform a non-linear filtering to recover ϕ_0.

3.3 Smoothing of the Control Inputs

In the previous section, we showed how to extract parameters s_0 and ϕ_0 from the updated positions determined from the vision system. The signals s_1, s_2^2 described in equations 9 and 11 are in fact the smoothed expected values of the control signals s, s^2 which are the arclength and the arclength squared. The smoothing filter we use to compute these signals is a moving-average (MA) filter using a Kaiser window [13]. This filter provides the largest ratio of signal energy in the main lobe and a side lobe, which usually results in a filter of lower order. The windowing function is given by

$$w_K(n) = \frac{I_0(\beta\sqrt{1-(1-2n/M)^2})}{I_0(\beta)} \tag{17}$$

where I_0 is the modified zeroth-order Bessel function, β is the shape parameter which defines the width of the main lobe and M is the order of the filter. According to [13], β and M are given by

$$M \approx \frac{A - 7.95}{14.36\Delta\omega}$$

and

$$\beta = \begin{cases} 0.1102(A - 8.7), & A \geq 50 \\ 0.5842(A-21)^{0.4} + 0.07886(A-21), & 21 < A < 50 \end{cases}$$

where A is the stopband attenuation and $\Delta\omega = (\omega_r - \omega_c)/\omega_s$, ω_r is the stopband frequency, ω_c is the passband frequency and ω_s is the sampling frequency.

We have adopted $A = 30$ and $\Delta\omega = 0.05$ which results in $M = 30$. Since the frequency of the vision algorithm is about 60 Hz, the overall length of the window is about 0.5 seconds. We also apply this MA filter to the bending parameter ϕ.

The implementation of MA filter is straightforward: once the weights are computed off-line, a window of length M of measurements is retained and each sample is multiplied by an appropriate weight in the sampling period, which requires M multiplications and $M-1$ additions. This allows reasonably wide windows (even up to several hundreds entries) to be used in computing the smoothed signal.

3.4 Prediction and Synchronization

The host computer controls the initial vision processing and subsequent computation of control parameters described above. The host computer is able to predict ahead the trajectory using the derivation of velocity and curvature in equations (1) and (2). These updated predictions are sent to the trajectory generator that is actually controlling the robot arm. The trajectory generator is a separate system that has two parallel tasks: a low-priority task which reads the serial line receiving updated control signals and high-priority task which calculates the transformation equation and moves the manipulator. Those two tasks communicate via shared memory. The job of the robot controlling program is to synchronize its two tasks (i.e. to obtain mutual exclusion in accessing shared data), to unpack input packets read from the serial line, and to update the joint servos every 30 msec.

The asynchronous nature of the communication between the host computer and the trajectory generator can result in missed or delayed communications between the two systems. Since the updating of the robotic arm parameters needs to be done at very tightly specified servo rates (30 msec), it is imperative that the trajectory generator can provide updated control parameters at these rates, regardless of whether it has received a new control input from the host. Therefore, we have implemented a fixed gain $\alpha - \beta - \gamma$ filter as part of the trajectory generator [18]. This filter provides a small amount of prediction to the trajectory parameters if the control signals from the host are delayed.

We are using RCCL [10] to control the robotic arm (a PUMA 560). RCCL (Robot Control C Library) allows the use of C programming constructs to control the robot as well as defining transformation equations (as described in [17]). The transformation equations permit dynamic updating of arm position by generating the 4 x 4 transform of the moving object's position from the vision system and sending this information to the arm control algorithm.

4 GRASPING

The remaining part of our system is the interception and grasping of the object. We have examined the human psychological literature in order to find useful paradigms for robotic visual-motor coordination strategies that include arm movement and grasping from visual inputs. In this section we briefly describe some relevant theories and their relation to our own work.

There are several theories on the organization of skilled human motor control. Richard Schmidt [9] has proposed a theory of generalized motor programs, or movement *schemas*. In this view, a skilled action is composed of an ordered set of parametrized motor control programs of short duration (less than 200 msec), each of which accomplishes one part of the task. As one program is completed, the next one is executed. Generalized motor programs accomplish several objectives: (1) they specify *which muscle* to move in a given motion; (2) the *order of contraction* of the muscles; (3) the *phasing* within the sequence, i.e., the temporal relationships among the contractions; (4) the *relative force* of each element. At the initiation of a skilled task, the parameters of the motor control program are determined by sensory input and task demands, and then the programs are executed to completion. If the wrong program is selected for some reason, the program cannot be stopped by use of sensory information. As in playing table tennis, the motion of the racket is determined before the beginning of the swing and visual input has little effect after the initiation of motion. As an example of Schmidt's theory, the skilled task of grasping a moving object could be partitioned into two motor control schemas: one to position the arm and a second one to control the grasping action.

The schema concept maps into Von Hofsten's ideas about the development of grasping skills in children [21]. He believes there are two separate sensorimotor systems responsible for reaching: one for approaching the target and one for grasping it. During early childhood, the precise timing between these two systems develops as the child learns how to catch. The reaching system develops first, before a child is capable of grasping. But even before he is capable of closing his hand at precisely the right moment, he has begun to develop the ability to move his hand toward a moving object and predict the location at which his hand will intercept the object. With growth, a child learns to control the timing between reaching and grasping, that is, to close his hand at the correct moment. Experimental evidence has shown that there is a window of approximately 14 msec during which the hand must begin closing. Unlike Schmidt, however, Von Hofsten does not consider vision and grasping to be two mutually exclusive tasks [20]. Visual tracking is used to guide the reaching arm *during* its motion, not only before motion. A coordinated motion is a combination of perceptual schemas and motor schemas [6].

Vision is used during the reaching phase of the task for what psychologists call "prospective control". Prospective control corresponds to predictive filtering, as used by control theorists. In grasping a moving object, it is necessary for the hand to move not to the current position of the object, but to plan ahead to where it will be shortly. Vision, rather than haptics, provides the basis of prospective control because touch cannot provide the anticipatory information required to predict the course of a moving object. There are two predominant theories about what visual schema is used to track a moving object and aid in predicting the intersection of the reaching hand and that object. Lee [15] proposes the use of vision to measure the expansion of the image on the retina in order to estimate the time until contact. The attraction of this theory is that humans would not need to compute the velocity and location of the moving object, but would calculate the more useful time-until-contact information. A person catching an object uses this image to compute when to begin the correct motion commands (usually at about 300 msec before the actual grasp). Von Hofsten disputes the use of retinal expansion information because it is clear that people are able to track targets in which there is no such expansion, such as objects that are circling or passing across the field of view. He suggested an alternative schema in which people calculate the distance to a moving object by using the vergence angle to the object. Vision seems to be used predominantly to track the moving object, but the catcher also tracks his hand during reaching to aid his non-visual proprioceptive senses, that is, to help judge the position of his hand in relation to the environment. Finally, vision must be used during the reaching phase to orient the hand correctly in relation to the object that is being caught.

We also note a relevant fact for human contact and grasping of objects. The central factor to the final grasp is the time of the onset of hand closure. In early childhood (up to about 5 months), closing the hand is triggered primarily by touch. Children tend to begin grasping only when they are already in contact with the object. By the time a child is 13 months old, however, the hand closes before touch, on average as early as for adults. We take the view below that our robotic system is past early childhood - we will close the hand before actual contact is made.

The initial strategy we have adopted in picking up the object is an open loop strategy, similar in spirit to the pre-programmed motor control schemas described in the psychological literature. Schmidt's schema theory holds that for tasks of short duration, perception is used to find a set of parameters to pass to a motor control program. It is not used during the execution of a task. When grasping a moving object, for example, once vision determined the trajectory of the object, the reach and grasping motor schemas take over with no interference from vision.

In our implementation of this strategy, vision is not used to continually monitor the grasping, but only to provide a final position and velocity from which the arm is directed to very quickly move to the object. This automatic movement is done by establishing coordinate frames of action for each of the components of the system and solving transformation equations.

The transformation equations permit dynamic updating of the arm position by generating the 4×4 transform of the moving object's position from the vision system and sending this information to the arm control algorithm. This positional information from the vision system is used to update the transformaton equations. The other transforms in the equation are known, and this allows the system to solve for the unknown control transform which is the transform used to update the manipulator's joints and develop a straight line path in Cartesian coordinates that will bring the hand into contact with the moving object. Because the movement of the hand requires a small amount of time during which the object may have moved, the object's trajectory is predicted ahead during the movement using the $\alpha - \beta - \gamma$ predictor. By keeping the fingers of the hand spread during this maneuver, no actual contact takes place until the gripper reaches the position of the moving object.

5 EXPERIMENTAL RESULTS

We have implemented the system described above in order to demonstrate the capability of the methods. The goal was to track a moving model train, intercept it, stably grasp it and pick it up. The train was moving in an oval trajectory; however, the system had no *a priori* knowledge of this particular trajectory. The setup of our system is presented in figure 6. The velocity of the train was $10-20cm/s$. In this section we present some results obtained by experiments. First, in figure 7 we have the actual measured arclength signal s_1 (black) and the filtered signal s_0 (gray). It is noticeable that s_0 is somewhat *below* the expected value of s_1. The nature of s_1 is quite noisy; however, the analysis described in section 4 was able to accurately extract the correct

control signal. The arm control is particularly smooth and jerk free, as well as accurate enough to intercept and grasp the object between the jaws of the gripper. Figure 8 shows the moving object's trajectory points computed by the vision algorithm (black) and the commanded control signals after filtering (gray). As can be seen, the control system is able to accomplish its task of both smoothing for noise and extracting an accurate position of the moving object.

Because we are using a parallel jaw gripper, the jaws must remain aligned with the tangent to the actual trajectory of the moving object. The system controls the gripper direction (joint 6 on the robot) to be parallel to this tangential direction, allowing grasping to occur at any point in the trajectory.

Figure 6 shows 3 frames taken from a video tape of the system intercepting, grasping and picking up the object (this video tape is part of the video proceedings of the 1992 IEEE Robotics and Automation conference). The system is quite repeatable, and is able to track other arbitrary trajectories in addition to the one shown.

6 SUMMARY

We have developed a robust system for tracking and grasping moving objects. The system relies on real-time stereo triangulation of optic-flow fields and is able to cope with the inherent noise and inaccuracy of visual sensors by applying parameterized filters that smooth and can predict ahead the moving object's position. Once this tracking is achieved, a grasping strategy is applied that performs an analog of human arm movement schemas.

Our future work is concerned with implementing other possible grasping strategies. One strategy we are currently exploring is to visually monitor the interception of the hand and object and use this visual information to update the control transform at video update rates. This approach is computationally more demanding, requiring multiple moving object tracking capability. The initial vision tracking described above is capable of single object tracking only. If we attempt to visually servo the moving robotic arm with the moving object, we have introduced multiple moving objects into the scene.

We have identified 2 possible approaches to tracking these multiple objects visually. The first is to use the PIPE's region of interest operator that can effectively "window" the visual field and compute different motion energies in each window concurrently. Each region can be assigned to a different stage of the PIPE and compute its result independently. This approach assumes that the moving objects can be segmented. This is possible since the motion of the hand in 3-D is known - we have commanded it ourselves. Therefore, since we know the camera parameters and 3-D position of the hand, it will be possible to find the relevant image-space coordinates that correspond to the 3-D position of the hand. Once these are known, we can form a window centered on this position in the PIPE, and concurrently compute motion energy of the moving object and the moving hand in each camera. Each of these motion centroids can then be triangulated to find the effective positions of both the hand and object and compute the new control transform. Both computations must, however, compete for the hardware histogramming capability needed for centroid computation, and this will effectively reduce the bandwidth of position updating by a factor of 2.

Another approach is to use a coarse-fine hierarchical control system that uses a multi-sensor approach. As we approach the object for grasping, we can shift the visual attention from

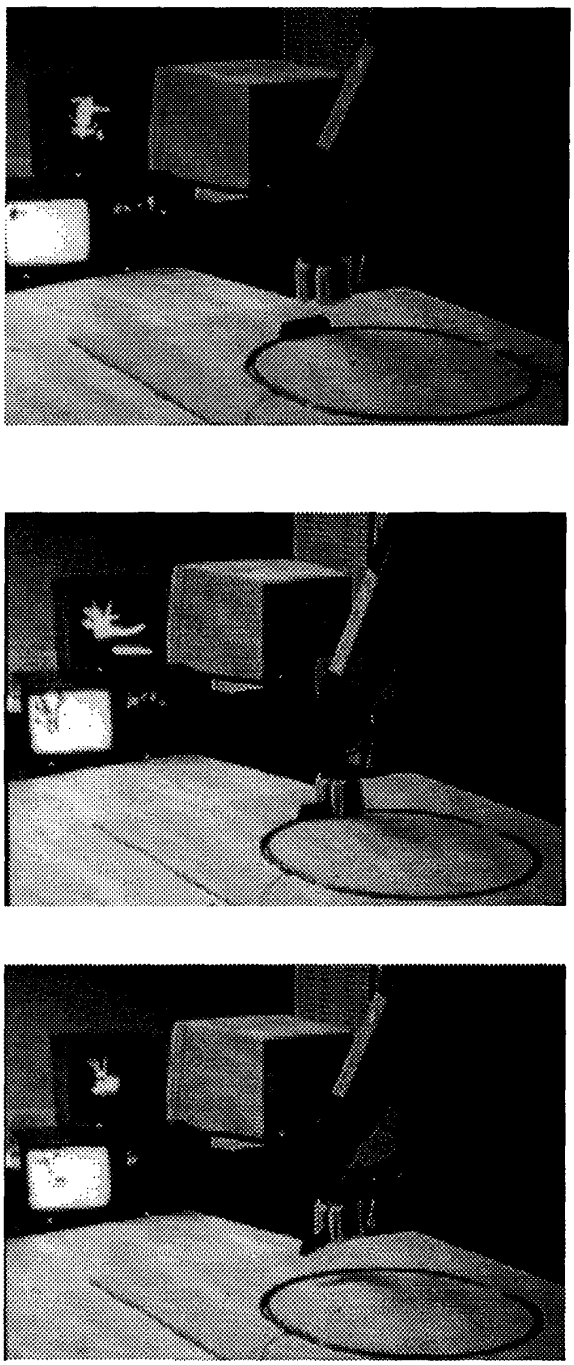

Figure 6: Intercepting, picking up and grasping the object

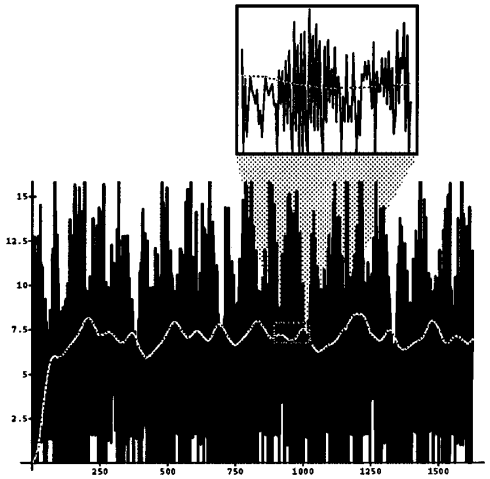

Figure 7: Input signal s_1 (black) and filtered signal s_0 (gray)

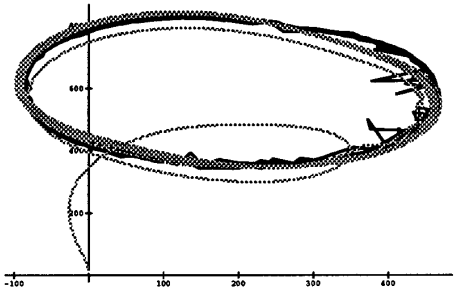

Figure 8: Input trajectory (black) and filtered trajectory (gray)

the static cameras used in 3-D triangulation to a single camera mounted on the wrist of the robotic hand. Once we have determined that the moving object is in the field of view of this camera, we can use its estimates of motion via optic-flow to keep the object to be grasped in the center of the wrist camera's field of view. This control information will be used to compute the control transform to correctly move the hand to intercept the object. We have implemented such a tracking system with a different robotic system [4] and can adapt this method to this particular task.

References

[1] E. H. Adelson and J. R. Bergen. Spatio-temporal energy models for the perception of motion. *Journal of the Optical Society of America*, 2(2):284–299, 1985.

[2] P. K. Allen, A. Timcenko, B. Yoshimi, and P. Michelman. Automated tracking and grasping of a moving object with a robotic hand-eye system. Technical report, Department of Computer Science, Columbia University, New York, NY, 1991.

[3] P. K. Allen, B. Yoshimi, and A. Timcenko. Real-time visual servoing. In *Proceedings of the IEEE Conference on Robotics and Automation*, 1991.

[4] Peter Allen. Real-time motion tracking using spatio-temporal filters. In *Proceedings of DARPA Image Understanding Workshop*, Palo Alto, May 1989.

[5] P. Anandan. Measuring visual motion from image sequences. Technical Report COINS TR-87-21, COINS Dept., University of Massachusetts-Amherst, 1987.

[6] Michael Arbib, Thea Iberall, and Damian Lyons. Coordinated control programs for movements of the hand. Technical Report COINS TR 83-25, Dept. of CS University of Massachusetts, August 1983.

[7] P. J. Burt, C. Yen, and X. Xu. Multi-resolution flow-through motion analysis. In *Proceedings of the IEEE CVPR Conference*, pages 246–252, 1983.

[8] B. F. Buxton and H. Buxton. Computation of optic flow from the motion of edge features in image sequences. *Image and Vision Computing*, 2, 1984.

[9] H. Heuer H. Cruse, J. Dean and R.A. Schmidt. Utilization of sensory information for motor control. In Herbert Heuer and Andries F. Sanders, editors, *Perspectives on Perception and Action*, pages 43–79. Lawrence Erlbaum, 1987.

[10] Vincent Hayward and Richard Paul. Robot manipulator control under UNIX. In *Proc. of the 13th ISIR*, pages 20:32–20:44, Chicago, April 17-21 1983.

[11] David Heeger. A model for extraction of image flow. In *First International Conference on Computer Vision*, London, 1987.

[12] B. K. P. Horn and B. Schunck. Determining optical flow. *Artificial Intelligence*, 17:185–203, 1983.

[13] L. B. Jackson. *Digital Filters and Signal Processing*. Kluwer Academic Publishers, 1986.

[14] E. W. Kent, M. O. Shneier, and R. Lumia. Pipe: Pipelined image processing engine. *Journal of Parallel and Distributed Computing*, (2):50–78, 1985.

[15] D.N. Lee, D.S. Young, P.E. Reddish, S. Lough, and T.M.H Clayton. Visual timing in hitting an accelerating ball. *Quarterly Journal of Experimental Psychology*, 35A:333–346, 1983.

[16] H. H. Nagel. On the estimation of dense displacement vector fields from image sequences. In *Workshop on motion: Representation and Perception*, pages 59–65, Toronto, 1983.

[17] Richard Paul. *Robot Manipulators*. MIT Press, Cambridge, MA, 1981.

[18] R. B. Safadi. An adaptive tracking algorithm for robotics and computer vision application. Master's thesis, University of Pennsylvania, 1988.

[19] G. L. Scott. Four-line method of locally estimating optic flow. *Image and Vision Computing, 5(2)*, 1986.

[20] Claes von Hofsten. Catching. In Herbert Heuer and Andries F. Sanders, editors, *Perspectives on Perception and Action*, pages 33–36. Lawrence Erlbaum Associates, 1987.

[21] Claes von Hofsten. Early development of grasping an object in space-time. In Melvyn A. Goodale, editor, *Vision and Action: The Control of Grasping*, pages 65–79. Ablex Publishing Company, 1990.