IEEE/RSJ International Workshop on Intelligent
Robots and Systems IROS '91, Nov. 3-5, 1991,
Osaka, Japan. IEEE Cat. No. 91TH0375-6

A Model for Observing a Moving Agent

Tarek M. Sobh and Ruzena Bajcsy

GRASP Laboratory
Department of Computer and Information Science
School of Engineering and Applied Science
University of Pennsylvania
Philadelphia, PA 19104, U.S.A.

Abstract

We address the problem of observing a moving agent. In particular, we propose a system for observing a manipulation process, where a robot hand manipulates an object. A discrete event dynamic system (DEDS) frame work is developed for the hand/object interaction over time and a stabilizing observer is constructed. Low-level modules are developed for recognizing the "events" that causes state transitions within the dynamic manipulation system. The work examines closely the possibilities for errors, mistakes and uncertainties in the manipulation system, observer construction process and event identification mechanisms. The system utilizes different tracking techniques in order to observe and recognize the task in an *active*, *adaptive* and *goal-directed* manner.

1 Introduction

The problem of observing a moving agent was addressed in the literature extensively. It was discussed in the work addressing tracking of targets and, determination of the optic flow [2,7,10,17], recovering 3-D parameters of different kinds of surfaces [6,12,15], and also in the context of other problems [1,3,8,9]. However, the need to *recognize*, *understand* and *report* on different visual steps within a dynamic task was not sufficiently addressed. In particular, there is a need for high-level symbolic interpretations of the actions of an agent that attaches meaning to the 3-D world events, as opposed to simple recovery of 3-D parameters and the consequent tracking movements to compensate their variation over time.

In this work we establish a framework for the general problem of observation, recognition and understanding of dynamic visual systems, which may be applied to different kinds of visual tasks. We concentrate on the problem of observing a manipulation process in order to illustrate the ideas and motive behind our framework. We use a discrete event dynamic system as a high-level structuring technique to model the visual manipulation system. Our formulation uses the knowledge about the system and the different actions in order to solve the observer problem in an efficient, stable and practical way. The model incorporates different hand/object relationships and the possible errors in the manipulation actions. It also uses different tracking mechanisms so that the observer can keep track of the workspace of the manipulating robot. A frame work is developed for the hand/object interaction over time and a stabilizing observer is constructed. Low-level modules are developed for recognizing the "events" that causes state transitions within the dynamic manipulation system. The process uses a coarse quantization of the manipulation actions in order to attain an active, adaptive and goal-directed sensing mechanism.

The work examines closely the possibilities for errors, mistakes and uncertainties in the visual manipulation system, observer construction process and event identification mechanisms, leading to a DEDS formulation with uncertainties, in which state transitions and event identification is asserted according to a computed set of 3-D uncertainty models.

We describe the automaton model of a discrete event dynamic system in the next section and then proceed to formulate our framework for the manipulation process and the observer construction. Then we develop efficient low-level event-identification mechanisms for determining different manipulation movements in the system and for moving the observer. Next, the uncertainty levels are described in details. Some results from testing the system is enclosed and future extensions to the system are discussed.

2 Discrete Event Dynamic Systems

Discrete event dynamic systems (DEDS) are dynamic systems (typically asynchronous) in which state transitions are triggered by the occurrence of discrete events in the system. DEDS are usually modeled by finite state automata with partially observable events together with a mechanism for enabling and disabling a subset of state transitions [11,13,14]. We propose that this model is a suitable framework for many vision and robotics tasks, in particular, we use the model as a high-level structuring technique for our system to observe a robot hand manipulating an object. We can represent a DEDS by the quadruple :

$$G = (X, \Sigma, U, \Gamma) \tag{1}$$

where X is the finite set of states, Σ is the finite set of possible events, U is the set of admissible control inputs consisting of a specified collection of subsets of Σ, corresponding to the choices of sets of controllable events that can be enabled and $\Gamma \subseteq \Sigma$ is the set of observable events.

We can visualize the concept of DEDS by an example as in Figure 1, the graphical representation is quite similar to a classical finite automaton. Here, circles denote states, and events are represented by arcs. The first symbol in each arc label denotes the event, while the symbol following "/" denotes the correspond-

ing output (if the event is observable). Finally, we mark the controllable events by ":u".

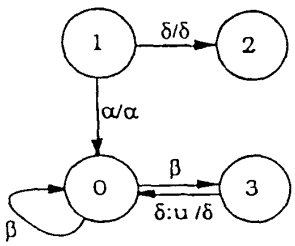

Figure 1 : A Simple DEDS Example

Thus, in this example, $X = \{0,1,2,3\}$, $\Sigma = \{\alpha,\beta,\delta\}$, $\Gamma = \{\alpha,\delta\}$, and δ is controllable at state 3 but not at state 1.

Stability can be defined with respect to the *states* of a DEDS automaton. Assuming that we have identified the set of "good" states, E, that we would like our DEDS to "stay within" or do not stay outside for an infinite time, then stabilizability can be formally defined as follows :

Given a live system A and some $E \subset X$, $x \in X$ is stabilizable with respect to E (or E-stabilizable) if there exists a state feedback K such that x is alive and E-stable in A_K. A set of states, Q, is a stabilizable set if there exists a feedback law $K(s)$ (a control pattern) so that every $x \in Q$ is alive and stable in A_K, and A is a stabilizable system if X is a stabilizable set.

A DEDS is termed *observable* if we can use the observation sequence to determine the current state exactly at intermittent points in time separated by a *bounded* number of events. More formally, taking any sufficiently long string, s, that can be generated from any initial state x. For any observable system, we can then find a prefix p of s such that p takes x to a *unique* state y and the length of the remaining suffix is bounded by some integer n_o. Also, for any other string t, from some initial state x', such that t has the same output string as p, we require that t takes x' to the same, unique state y.

The basic idea behind strong output stabilizability is that we will know that the system is in state E iff the observer state is a subset of E. The compensator should then *force* the observer to a state corresponding to a subset of E at intervals of at most a finite integer i observable transitions. If Z is the set of states of the observer, then :

A is strongly output E-stabilizable if there exists a state feedback K for the observer O such that O_K is stable with respect to $E_O = \{ \hat{x} \in Z \mid \hat{x} \subset E \}$.

3 Modeling and Observer Construction

Manipulation actions can be modeled efficiently within a discrete event dynamic system framework.We use the DEDS model as a high level structuring technique to preserve and make use of the information we know about the way in which each manipulation task should be performed.

3.1 Building the Model

We present a simple model for a grasping task. The model is that of a gripper approaching an object and grasping it. As shown in Figure 2, the model represents a view of the hand at state 1, with no object in sight, at state 2, the object starts to appear, at state 3, the object is in the claws of the gripper and at state 4, the claws of the gripper close on the object. Different orientations for the approaching hand are allowable and observable. State changes occur only when the object appear in sight or when the hand encloses it. It should be noted that these states can be considered as the set of "good" states E, since these states are the expected different visual configurations of a hand and object within a grasping task. States 5 and 6 represent instability in the system as they describe the situation where the hand is not centered with respect to the camera imaging plane. The events are defined as motion vectors or motion vector probability distributions, as will be described later, that causes state transitions and as the appearance of the object into the viewed scene. The controllable events are denoted by ": t'.

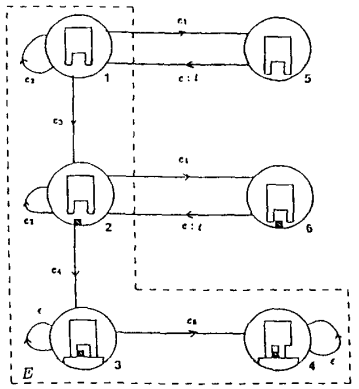

Figure 2 : A Model for a Grasping Task

3.2 Developing the Observer

In order to know the current state of the manipulation process we need to observe the sequence of events occurring in the system and make decisions regarding the state of the automaton, state ambiguities are allowed to occur, however, they are required to be resolvable after a *bounded* interval of events. The goal will be to make the system a strongly output stabilizable one and/or construct an observer to satisfy specific task-oriented visual requirements. As an example, for the model of the grasping task, an observer can be formed for the system as shown in Figure 3. It can be easily seen that the system can be made stable with respect to the set E_O.

3.3 Identifying Motion Events

We use the image motion to estimate the hand movement. This task can be accomplished by either feature tracking or by computing the full optic flow. The image flow detection technique we use is based on the sum-of-squared-differences optic flow. The sensor acquisition procedure (grabbing images) and uncertainty

in image processing mechanisms for determining features are factors that should be taken into consideration when we compute the uncertainty in the optic flow.

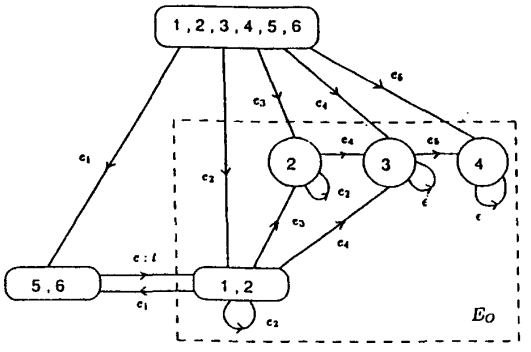

Figure 3 : Observer for the Grasping System

One can model an arbitrary 3-D motion in terms of stationary-scene/moving-viewer as shown in Figure 4. The optical flow at the image plane can be related to the 3-D world as indicated by the following pair of equations for each point (x, y) in the image plane [12] :

$$v_x = \left\{ x\frac{V_Z}{Z} - \frac{V_X}{Z} \right\} + \left[xy\Omega_X - \left(1 + x^2\right)\Omega_Y + y\Omega_Z \right] \quad (2)$$

$$v_y = \left\{ y\frac{V_Z}{Z} - \frac{V_Y}{Z} \right\} + \left[\left(1 + y^2\right)\Omega_X - xy\Omega_Y - x\Omega_Z \right] \quad (3)$$

where v_x and v_y are the image velocity at image location (x, y), (V_X, V_Y, V_Z) and $(\Omega_X, \Omega_Y, \Omega_Z)$ are the translational and rotational velocity vectors of the observer, and Z is the unknown distance from the camera to the object. In this system of equations, the only knowns are the 2-D vectors v_x and v_y, if we use the formulation with uncertainty then basically the 2-D vectors are random variables with a known probability distribution. A number of techniques can be used to linearize the system of equations and to solve for the motion and structure parameters as random variables [4,5,15].

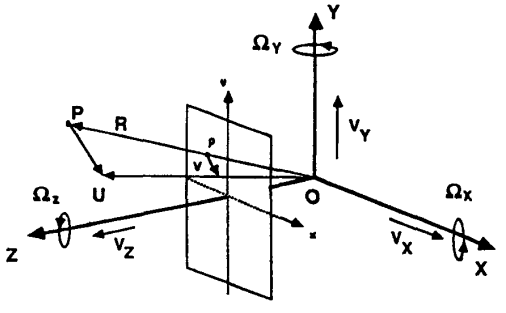

Figure 4 : 3-D Formulation for Stationary-Scene/Moving Viewer

4 Modeling and Recovering 3-D Uncertainties

The uncertainty in the recovered image flow values results from sensor uncertainties and noise and from the image processing techniques used to extract and track features. We use a static camera calibration technique to model the uncertainty in 3-D to 2-D feature locations. The strategy used to find the 2-D uncertainty in the features 2-D representation is to utilize the recovered camera parameters and the 3-D world coordinates (x_w, y_w, z_w) of a known set of points and compute the corresponding pixel coordinates, for points distributed throughout the image plane a number of times, find the actual feature pixel coordinates and construct 2-D histograms for the displacements from the recovered coordinates for the experiments performed. The number of the experiments giving a certain displacement error would be the z axis of this histogram, while the x and y axis are the displacement error. The three dimensional histogram functions are then normalized such that the volume under the histogram is equal to 1 unit volume and the resulting normalized function is used as the distribution of pixel displacement error.

The spatial uncertainty in the image processing technique can be modeled by using synthesized images and corrupting them, then applying the feature extraction mechanism to both images and computing the resulting spatial histogram for the error in finding features. The probability density function for the error in finding the flow vectors can thus be computed as a spatial convolution of the sensor and strategy uncertainties. We then eliminate the unrealistic motion estimates by using the physical (geometric and mechanical) limitations of the manipulating hand. Assuming that feature points lie on a planar surface on the hand, then we can develop bounds on the coefficients of the motion equations, which are second degree functions in x and y in three dimensions, $v_x = f_1(x, y)$ and $v_y = f_2(x, y)$.
Figure 5 indicates the maximal v_x that can ever be registered on the CCD array of the camera, the x and y are in millimeters and the $x - y$ plane represents the CCD image plane, the depth Z is the maximal v_x in millimeters on the CCD array that can ever be registered. As an example, we write the equation governing the maximum v_x value in the first quadrant of the $x - y$ plane (x^+, y^+).

$$v_{x_{max}} = \left(-\frac{fV_{X_s}}{Z_{o_s}} - f\Omega_{Y_s} \right) + \left(\frac{V_{Z_l}}{Z_{o_s}} + \frac{max(p_l V_{X_l}, p_s V_{X_s})}{Z_{o_s}} \right) x +$$
$$\left(\frac{max(q_l V_{X_l}, q_s V_{X_s})}{Z_{o_s}} + \Omega_{Z_l} \right) y + \left(\frac{\Omega_{X_l}}{f} - \frac{min(q_l V_{Z_s}, q_s V_{Z_l})}{f Z_{o_s}} \right) xy$$
$$- \left(\frac{min(p_l V_{Z_s}, p_s V_{Z_l})}{f Z_{o_s}} + \frac{\Omega_{Y_s}}{f} \right) x^2 \quad (4)$$

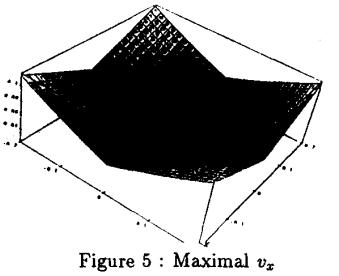

Figure 5 : Maximal v_x

where the subscripts s and l denote lower and upper limits, respectively. The above envelopes are then used to reject unrealistic 2-D velocity estimates at different pixel coordinates in the image. The 2-D uncertainties are then used to recover the 3-D uncertainties in the motion and structure parameters. The system is linearized by either dividing the parameter space into three subspaces for the translational, rotational and structure parameters and solving iteratively or using other linearization techniques and/or assumptions to solve a linear system of random variables [4,5,6,15,16,18]. As an example, the recovered 3-D translational velocity cumulative density function in the Z direction for an actual world motion, $V_X = 0\ cm$, $V_Y = 0\ cm$ and $V_Z = 13\ cm$, is shown in Figure 6.

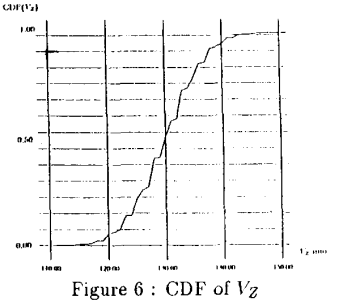

Figure 6 : CDF of V_Z

5 Conclusions

State transitions are asserted within the DEDS observer model according to the probability value of the occurrence of an event. Events are thus defined as ranges for the different parameters. The problem then reduces to computing the corresponding areas under the refined distribution curves. An obvious way of using those probability values is to establish some threshold values and assert transitions according to those thresholds. It might be the case that none of the obtained probability values exceeds the set threshold value and/or all values are very low. In that case, there is a good chance that we are at either the wrong automata state. The remedy to such problems can be implemented through time proximity, that is, wait for a while (which is to be preset) till a strong probability value is registered and/or *backtrack* in the automaton model for the observer till a high enough probability value is asserted, a fail state is reached or the initial ambiguity is asserted. The backtracking strategy can be implemented using a stack-like structure associated with each state that has already been traversed, which includes a sorted list of the computed event probabilities and a father-state variable.

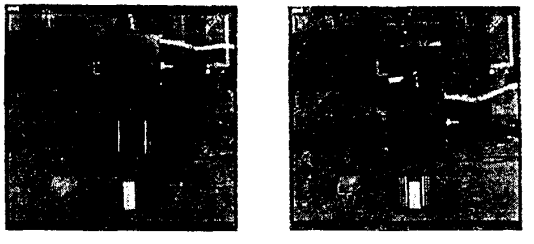

Figure 7 : A Grasping Task

Experiments were performed to observe the robot hand. The low level visual feature acquisition is performed on the Datacube MaxVideo pipelined video processor at frame rate. The observer and manipluating robots are both PUMA 560's and the Lord experimental gripper is used as the manipulating hand. A grasping task using the Lord gripper, as seen by the observer, is shown in Figure 7. Thus, we have proposed a new approach to solving the problem of observing a moving agent. Our approach uses the formulation of discrete event dynamic systems as a high-level model for the framework of evolution of the visual relationship over time. The proposed formulation can be extended to accommodate for more manipulation processes. Increasing the number of states and expanding the events set would allow for a variety of manipulating actions.

References

[1] J. Aloimonos and A. Bandyopadhyay, "Active Vision". In *Proceedings of the 1st International Conference on Computer Vision*, 1987.

[2] P. Anandan, "A Unified Perspective on Computational Techniques for the Measurement of Visual Motion". In *Proceedings of the 1st International Conference on Computer Vision*, 1987.

[3] R. Bajcsy, "Active Perception", *Proceedings of the IEEE*, Vol. 76, No. 8, August 1988.

[4] R. Bajcsy and T. M. Sobh, *A Framework for Observing a Manipulation Process.* Technical Report MS-CIS-90-34 and GRASP Lab. TR 216, University of Pennsylvania, June 1990.

[5] R. Bajcsy and T. M. Sobh, *Observing a Moving Agent.* Technical Report MS-CIS-91-01 and GRASP Lab. TR 247, Computer Science Dept., School of Engineering and Applied Science, University of Pennsylvania, January 1991.

[6] J. L. Barron, A. D. Jepson and J. K. Tsotsos, "The Feasibility of Motion and Structure from Noisy Time-Varying Image Velocity Information", *International Journal of Computer Vision*, December 1990.

[7] P. J. Burt, et al., "Object Tracking with a Moving Camera", *IEEE Workshop on Visual Motion*, March 1989.

[8] F. Chaumette and P. Rives, "Vision-Based-Control for Robotic Tasks", In *Proceedings of the IEEE International Workshop on Intelligent Motion Control*, Vol. 2, pp. 395-400, August 1990.

[9] J. Hervé, P. Cucka and R. Sharma, "Qualitative Visual Control of a Robot Manipulator". In *Proceedings of the DARPA Image Understanding Workshop*, September 1990.

[10] B. K. P. Horn and B. G. Schunck, "Determining Optical Flow", *Artificial Intelligence*, vol. 17, 1981, pp. 185-203.

[11] Y. Li and W. M. Wonham, "Controllability and Observability in the State-Feedback Control of Discrete-Event Systems", *Proc. 27th Conf. on Decision and Control*, 1988.

[12] H. C. Longuet-Higgins and K. Prazdny, *The interpretation of a moving Retinal Image*, Proc. Royal Society of London B, 208, 385-397.

[13] C. M. Özveren, *Analysis and Control of Discrete Event Dynamic Systems : A State Space Approach*, Ph.D. Thesis, Massachusetts Institute of Technology, August 1989.

[14] P. J. Ramadge and W. M. Wonham, "Modular Feedback Logic for Discrete Event Systems", *SIAM Journal of Control and Optimization*, September 1987.

[15] T. M. Sobh and K. Wohn, "Recovery of 3-D Motion and Structure by Temporal Fusion". In *Proceedings of the* 2^{nd} *SPIE Conference on Sensor Fusion*, November 1989.

[16] M. Subbarao and A. M. Waxman, *On The Uniqueness of Image Flow Solutions for Planar Surfaces in Motion*, CAR-TR-113, Center for Automation Research, University of Maryland, April 1985.

[17] S. Ullman, "Analysis of Visual Motion by Biological and Computer Systems", *IEEE Computer*, August 1981.

[18] S. Ullman, *Maximizing Rigidity: The incremental recovery of 3-D structure from rigid and rubbery motion*, AI Memo 721, MIT AI lab. 1983.