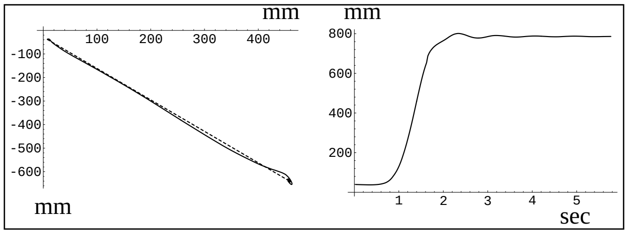

Figure 5: Linear target path and manipulator trajectory

manipulator trajectory from one of these experiments. With the pyramiding and search optimizations, the system was able to track the corner of a book which was moving along a linear path at 80 cm/sec. In contrast, the maximum speed reported by Papanikolopoulos [11] was 7 cm/sec, representing an order of magnitude increase in tracking speed. The dashed line is the target and the solid line represents the manipulator.

Once system performance met our minimal requirements, testing proceeded to a computer-generated target where the trajectory and speed could be accurately controlled. The target was displayed on a video monitor and the MVRT tracked the target. The target (a white box) traced a path on the video monitor while the MVRT tracked the target.

The experiments traced a square and demonstrated robust tracking in spite of the target exhibiting infinite accelerations (at initial start-up) and discontinuities in the velocity curves (at the corners of the square). The results in Figure 6 show the oscillatory nature of the manipulator path at the points where the velocity curve of the target is discontinuous.

6 Conclusion

This paper presents robust techniques for the operation of robotic agents in uncalibrated environments. The techniques presented provide ways of recovering unknown workspace parameters using the Controlled Active Vision framework [11]. In particular, this paper presents novel techniques for computing depth maps and for visual tracking through controlled active exploration with an eye-in-hand system.

For the computation of depth maps, we propose a scheme that is based on the automatic selection of features and the design of specific trajectories on the image plane for each individual feature. Unlike similar approaches [5][7][8][12][14], this approach helps us design trajectories that provide maximum identifiability of the depth parameter.

For the problem of visual tracking, we propose a technique based upon earlier work in visual servoing [11] that achieves superior speed and accuracy through the introduction of several performance enhancing techniques. In particular, the dynamic pyramiding technique provides a satisfactory compromise to the speed/accuracy trade-off inherent in static pyramiding techniques. The method-specific optimizations presented also enhance overall system performance without affecting worst-case execution times.

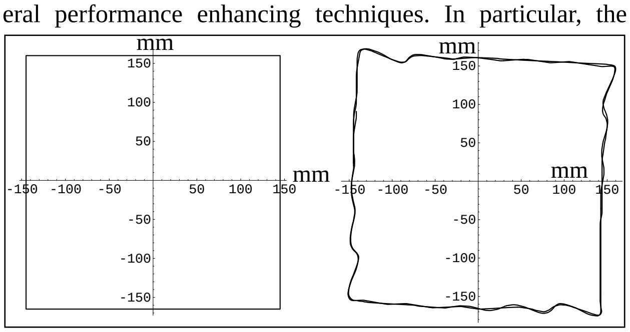

Figure 6: Target and manipulator paths

Issues for future research include the automatic selection of the feature window size (an issue discussed in [9]) in order to select a window that has some texture variations, and the implicit incorporation of the robot dynamics.

7 Acknowledgments

This work has been supported by the Department of Energy (Sandia National Laboratories) through Contract #AC-3752D, the Center for Transportation Studies through Contract #USDOT/DTRS93-G-0017-01, the 3M Corporation, the Center for Advanced Manufacturing, Design, and Control (CAMDAC — University of Minnesota), the Graduate School of the University of Minnesota, and the Department of Computer Science of the University of Minnesota.

8 References

[1] P. Anandan, “Measuring visual motion from image sequences,” Technical Report COINS-TR-87-21, COINS Department, University of Massachusetts, 1987.

[2] S. Brandt, “Enhanced robotic visual tracking under the controlled active vision framework”, Master’s Thesis, Department of Computer Science, University of Minnesota, 1993.

[3] J.T. Feddema and C.S.G. Lee, “Adaptive image feature prediction and control for visual tracking with a hand-eye coordinated camera,” *IEEE Transactions on Systems, Man, and Cybernetics*, 20(5):1172-1183, 1990.

[4] D.B. Gennery, “Tracking known three-dimensional objects,” *Proceedings of the AAAI 2nd National Conference on AI*, 13-17, 1982.

[5] K.N. Kutulakos and C.R. Dyer, “Recovering shape by purposive viewpoint adjustment,” *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 16-22, 1992.

[6] R.C. Luo, R.E. Mullen Jr., and D.E. Wessel, “An adaptive robotic tracking system using optical flow,” *Proceedings of the IEEE International Conference on Robotics and Automation*, 568-573, 1988.

[7] L. Matthies, T. Kanade, and R. Szeliski, “Kalman filter-based algorithms for estimating depth from image sequences,” *International Journal of Computer Vision*, 3(3):209-238, 1989.

[8] L. Matthies, “Passive stereo range imaging for semi-autonomous land navigation,” *Journal of Robotic Systems*, 9(6):787-816, 1992.

[9] M. Okutomi and T. Kanade, “A locally adaptive window for signal matching,” *International Journal of Computer Vision*, 7(2):143-162, 1992.

[10] N. Papanikolopoulos, P.K. Khosla, and T. Kanade, “Visual tracking of a moving target by a camera mounted on a robot: a combination of control and vision,” *IEEE Transactions on Robotics and Automation*, 9(1):14-35, 1993.

[11] N. Papanikolopoulos, “Controlled active vision,” Ph.D. Thesis, Department of Electrical and Computer Engineering, Carnegie Mellon University, 1992.

[12] C. Shekhar and R. Chellappa, “Passive ranging using a moving camera,” *Journal of Robotic Systems*, 9(6):729-752, 1992.

[13] C. Smith and N. Papanikolopoulos, “Derivation of depth maps through controlled active exploration,” To appear, *IEEE International Conference on Robotics and Automation*, 1994.

[14] C. Tomasi and T. Kanade, “Shape and motion from image streams under orthography: a factorization method,” *International Journal of Computer Vision*, 9(2):137-154, 1992.

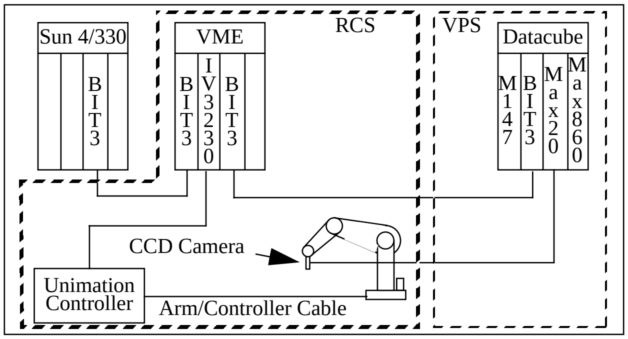

Figure 2: MRVT system architecture

effectively searches every other pixel in a 64×64 pixel patch. Consequently, the fourth and highest level searches every fourth pixel in a 128×128 pixel patch.

3 Feature Point Selection

In addition to the system latency and the effect of large displacements, an algorithm based upon the SSD technique may fail due to repeated patterns in the intensity function of the image or due to large areas of uniform intensity in the image. Both cases can provide multiple matches within a feature point's neighborhood, resulting in spurious displacement measures. In order to avoid this problem, our system automatically evaluates and selects feature points.

Feature points are selected using the SSD measure combined with an auto-correlation technique in the neighborhood Ω to produce a SSD surface [1][11]. Several possible confidence measures can be applied to the surface to measure the quality of a potential feature point. For discussion of our approach and proposed confidence measures, see [13].

4 Modeling and Controller Design

Modeling of the system in question is critical to the design of a controller to perform the task at hand. In our application areas, the modeling and controller designs are similar. In particular, Smith *et al*. [13] provide the depth recovery modeling and controller design. Papanikolopoulos *et al*. [10] provide an in-depth treatment of the modeling and controller design for robotic visual tracking applications.

5 Experimental Design and Results

5.1 The MRVT System

We have implemented both the depth recovery and the visual tracking on the Minnesota Robotic Visual Tracker (MRVT) system (see Figure 2). The MRVT is a multi-architectural system which consists of two main parts: the Robot/ Control Subsystem (RCS) and the Vision Processing Subsystem (VPS). A discussion of the MRVT appears in [2].

5.2 Derivation of Depth

The initial experimental runs were conducted using two soda cans wrapped in a checkerboard surface as targets. The leading edges of the cans were placed at 53 cm and 41 cm in depth from the nodal point of the camera (see Figure 3 (a)). The initial depth estimate was set to 25 cm in depth. The reconstructed surfaces are shown in Figure 3 (b).

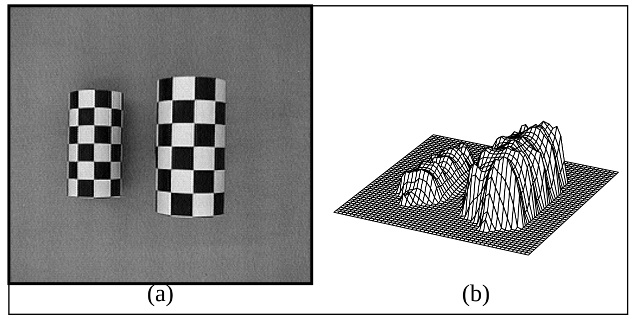

Figure 3: Single can target and surface

We duplicated the first set of experiments without the checkerboard surfaces wrapped around the cans. Instead, we used the actual surface of the soda cans under normal lighting conditions. This reduced the number of suitable feature points as well as produced a non-uniform distribution of feature points across the surfaces of the cans. Additionally, the surfaces exhibited specularities and reflections which added noise to the displacement measures. The targets and the results of the experiment are shown in Figure 4.

For all experiments, the camera scaling factors and focal length were taken directly from the documentation provided by the manufacturer and controller gains were set to match the specifications of the RCS. Neither camera calibration nor gain adjustment was performed for these experiments.

The resulting data for the feature points was not subjected to sub-pixel fitting nor multi-grid methods. The error in the recovered depth of the majority of feature points was less than the error expected for a measured displacement error of one pixel.

5.3 Visual Tracking

We conducted multiple experimental runs for the tracking of objects that exhibited unknown two-dimensional, translational motion with a coarse estimate of the depth of the objects. The targets for these runs included books, batons, and a computer-generated target displayed on a large video monitor. During the experiments, we tested the system using targets with linear and curved trajectories.

The first experiments were conducted using a book and a baton as targets in order to test the performance of the system both with and without the dynamic pyramiding. These initial experiments served to confirm the feasibility of performing real-time pyramid level switching and to collect data on the performance of the system under the pyramiding and search optimization schemes. Figure 5 shows the target path and

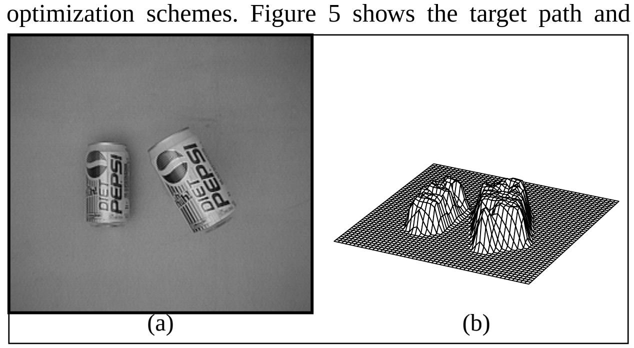

Figure 4: Single can with real texture and surface

tions both use the same basic visual measurements that are based upon a simple camera model and optical flow measurements in a sequence of images (see [11][13]). The visual measurements are combined with search-specific optimizations and a dynamic pyramiding technique in order to enhance the visual processing from frame-to-frame and to optimize the performance of the system in our selected applications.

2.1 Camera Model and Optical Flow

The derivation of the model and optical flow may be found in [13]. In this paper we only present the outcome of the derivation for the Sum-of-Squared Differences (SSD) optical flow [1]. For a point $\mathbf{p}(k-1)$, the SSD algorithm selects the displacement $\Delta\boldsymbol{x} = (u, v)^{\mathrm{T}}$ that minimizes the SSD measure

$$e(\mathbf{p}(k-1), \Delta\boldsymbol{x}) = \sum_{m, n \in N} \begin{bmatrix} I_{k-1}(x(k-1)+m), y(k-1)+n) - \\ I_k(x(k-1)+m+u, y(k-1)+n+v) \end{bmatrix}^2 \tag{2.1}$$

where $u,v \in \Omega$, N is the neighborhood of $\mathbf{p}$, m and n are indices for pixels in N, x and y are the indices of $\mathbf{p}$ in an image I, and I_{k-1} and I_k are the intensities in images $(k-1)$ and (k).

The size of the neighborhood N must be carefully selected to ensure proper system performance. Too small an N fails to capture large displacements while too large an N increases the associated computational overhead and enhances the background. In either case, an algorithm based upon the SSD technique may fail due to inaccurate displacements. An algorithm based upon the SSD technique may also fail due to too large a latency in the system or displacements resulting from motions of the object which are too large for the method to accurately capture. We introduce several techniques related to search optimizations and dynamic pyramiding to counter these concerns.

2.2 Search Optimizations

The primary source of latency in a vision system that uses the SSD measure is the time needed to compute $(u, v)^{\mathrm{T}}$ in Equation (2.1). To find the true minimum, the SSD measure must be calculated over each possible $(u, v)^{\mathrm{T}}$. The time required to produce an SSD surface and to find the minimum can be greatly reduced by employing two schemes that, when combined, decrease the search time significantly in the expected case.

The first optimization used is loop short-circuiting. During the search for the minimum on the SSD surface (the search for $(u_{\min}, v_{\min})$), the SSD measure must be calculated according to Equation (2.1). This requires nested loops for m and n. During the execution of these loops, the SSD measure is calculated as the running sum of the squared pixel value differences. If the current SSD minimum is checked against this sum as a condition on these loops, the execution of the loops can be short-circuited when the running sum exceeds the current minimum. On average, this type of short-circuit realizes a decrease in execution time by a factor of two.

The second optimization is based upon the heuristic that the best place to begin the search for the minimum is at the point where the minimum was found previously and to expand the search radially from this point. This works well when the disturbances being measured are relatively regular.

Under this heuristic, the search pattern in the (k) image is altered to begin at the point on the SSD surface where the minimum was located for the $(k-1)$ image. The search pattern then spirals out from this point, searching over the extent of u and v. This is in contrast with the typical indexed search pattern where the indices are incremented in a row-major scan fashion. Figure 1 contrasts a traditional row-major scan and the proposed spiral scan where the center position corresponds to the position where the minimum was last found. This search strategy may also be combined with a predictive controller to begin the search for the SSD minimum at the position that the predictive aspect of the controller indicates as the possible location of the minimum. In the general case, search time is approximately halved.

When combined, these two optimizations find the minimum of the SSD surface ten times faster on average than the unmodified search. Experimentally, the search times for the unmodified algorithm averaged 136 msec over 5000 frames under a variety of relative feature point motions. The combined short-circuit/spiral search algorithm produced search times which averaged 13 msec under similar tests.

2.3 Dynamic Pyramiding

Dynamic pyramiding is a heuristic technique which attempts to resolve the conflict between accurate positioning of a manipulator and high-speed tracking when the displacements of the feature points are large. Previous applications have typically depended upon one preset level of pyramiding to enhance either the manipulator's top tracking speed or its positioning accuracy with respect to the target [11].

In contrast, dynamic pyramiding uses multiple levels of pyramiding. The level of the pyramiding is selected based upon the observed displacements of the target's feature points. If the displacements are small relative to the search area, the pyramiding level is reduced; if the displacements are large compared to the search area, then the pyramiding level is increased. This results in a system that enhances the tracking speed when required, but is always biased in favor of the maximum accuracy achievable.

During the search process, the SSD measurements are centered upon particular pixels in the pyramided search area. Which pixel positions are selected ($\Delta\boldsymbol{x} = (u, v)^{\mathrm{T}}$ in Equation (2.1)) is dependent upon which of the four levels of pyramiding is currently active. The lowest level searches a square 32×32 pixel patch of the current frame. The second level

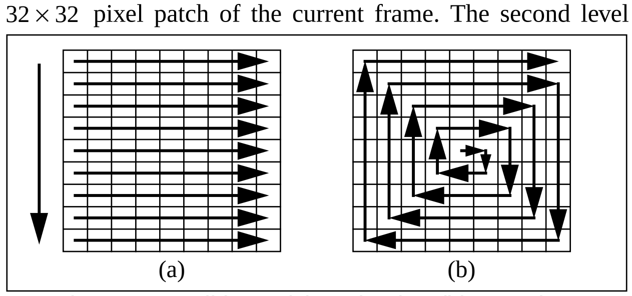

Figure 1: Traditional (a) and spiral (b) searches

Presented at the 1994 IEEE International Conference on Computer Vision and Pattern Recognition, June 20-24, 1994, Seattle, WA

Controlled Active Exploration of Uncalibrated Environments

Christopher E. Smith Scott A. Brandt Nikolaos P. Papanikolopoulos

Artificial Intelligence, Robotics, and Vision Laboratory
Department of Computer Science
University of Minnesota
4-192 EECS Building
200 Union St. SE
Minneapolis, MN 55455

Abstract

Flexible operation of a robotic agent in an uncalibrated environment requires the ability to recover unknown or partially known parameters of the workspace through sensing. Of the sensors available to a robotic agent, visual sensors provide information that is richer and more complete than other sensors. In this paper we present robust techniques for the derivation of depth from feature points on a target's surface and for the accurate and high-speed tracking of moving targets. We use these techniques in a system that operates with little or no a priori knowledge of the object- and camera-related parameters to robustly determine such object-related parameters as velocity and depth. Such determination of extrinsic environmental parameters is essential for performing higher level tasks such as inspection, exploration, tracking, grasping, and collision-free motion planning. For both applications, we use the Minnesota Robotic Visual Tracker (a single visual sensor mounted on the end-effector of a robotic manipulator combined with a real-time vision system) to automatically select feature points on surfaces, to derive an estimate of the environmental parameter in question, and to supply a control vector based upon these estimates to guide the manipulator.

1 Introduction

In order to be effective, robotic agents in uncalibrated environments must operate in a flexible and robust manner. The computation of unknown parameters (e.g., the velocity of objects and the depth of object feature points) is essential information for the accurate execution of many robotic tasks (e.g., manipulation, inspection, and exploration) in unstructured settings. The determination of such parameters has traditionally relied upon the accurate knowledge of other related environmental parameters. For instance, traditional approaches to the problem of depth recovery [3][4][7] have assumed that extremely accurate measurements of the camera parameters and the camera system geometry are provided *a priori*, making these methods useful in only a limited number of situations. Similarly, previous approaches [4][6] to visual tracking assumed known and accurate measures of camera parameters, camera positioning, manipulator positioning, target depth, target orientation, and environmental conditions.

This type of detailed information is not always available or, when it is available, not always accurate. Inaccuracies are introduced by positioning, path constraints, changes in the robotic system, and changes in the operational environment. In addition, camera calibration and determination of camera parameters can be computationally expensive and error prone. In particular, depth derivation and tracking techniques that rely upon stereo vision systems require careful geometry measurements and the solution of the correspondence problem, making the computational overhead prohibitive for real- or near-real-time systems. Furthermore, many structure-from-motion algorithms use simple accidental motion of the camera that does not guarantee the best possible identifiability of the depth parameter.

One solution to these problems can be found under the Controlled Active Vision (CAV) framework [11]. Papanikolopoulos [11] gives an overview of CAV; the techniques presented are useful under many situations, including the application areas we have selected: depth recovery and robotic visual tracking.

Instead of an accidental motion of the eye-in-hand system commonly used in depth extraction techniques [7][12][14], we propose a controlled exploratory motion that provides identifiability of the depth parameter. To reduce the influence of workspace-, camera-, and manipulator-specific inaccuracies, an adaptive controller is utilized to provide accurate and reliable information regarding the depth of an object's feature points. This information may then be used to guide operations such as tracking, inspection, and manipulation [3][11].

We also propose a visual tracking system which does not rely upon accurate measures of environmental and target parameters. An adaptive controller is used to track feature points on a target's surface in spite of the unconstrained motion of the target, possible occlusion of feature points, and changing target and environmental conditions. Relatively high-speed targets are tracked with only rough operating parameter estimates and no explicit target models. Tracking speeds are ten times faster and have similar tracking errors to those reported by Papanikolopoulos [11].

We first present an enhanced SSD surface construction strategy through search optimizations. We then briefly discuss the motivation for our feature point selection scheme. Finally, we discuss results from experiments in both the selected applications using the Minnesota Robotic Visual Tracker.

2 Visual Measurements

Our depth recovery and robotic visual tracking applica-