ACQUISITION OF MOVING OBJECTS AND HAND-EYE COORDINATION

R. T. Chien and V. C. Jones
Coordinated Science Laboratory
University of Illinois
Urbana, Illinois 61801
U.S.A.

tract

Many applications for robotic systems require the ability to handle moving objects. A method for providing a robot with real-time visual processing of moving objects is presented. By appropriately isolating the tracking task. It is possible to reduce the tracking problem to two dimensional pattern recognition. Numerous objects (potentially moving randomly) can be tracked while identifying one object at a time. The tracking ability can also be used to provide a robot with visual feedback (in real-time) for hand-eye coordination.

1. Introduction

Many potential applications for robotic systems, ranging from surveillance to semiconductor fabrication to extraterrestrial exploration, require ability to work with moving objects. Many industrial processes which should be automated are not because parts come down the assembly line with random orientation, indeterminate velocity, or otherwise unpredictable location. While current industrial robots cannot cope with this worst of randomness, a robot with dynamic vision could; even if only used to restore orientation for further processing by conventional automation techniques.

Detailed traffic monitoring [12] is another of those tedious tasks begging to be automated. The work required to analyze several hours worth of aerial photographs taken at the rate of one (or more) per second is staggering.

The difficulty is the "one-shot" approach to vision. Most effort in computer vision has been expended on analyzing a single picture to obtain all desired data. This approach is adequate only as long as the world is static enough for the results obtained to be valid by the time they are available. Clearly, what is needed is some way to dynamically update the world model, possibly even as it is still being built.

This is a presentation of an approach to avoid the "one-shot" problem by tracking significant (or potentially significant) features until their place in the world model can be determined. In many applications, it is possible to use standard (slow) recognition techniques even though the world is not static. Visual feedback for hand-eye coordination can also be attained using this technique.

This work supported by the Joint Services Electronics Program (U.S. Army, U.S. Navy and U.S. Air Force) under contract number DAAB-07-72-C-0259.

2. Three World Models of Dynamic Vision

The problems encountered in using robot vision in a dynamic environment are widely varied. However, robots must be able to handle motion in the field of view in order to handle virtually any dynamic problems.

Consider a stationary robot looking at a moving object. The object's location is continuously changing, so that by the time the object is identified, the robot may no longer know where it is. In addition, the object may be changing orientation or appearance. Only an extremely limited class of objects looks identical whether viewed from the front, side or back (e.g. spheres).

If the robot is moving and the rest of the world is not, the problem is very different. Here it is undesirable for the robot to require lengthy pauses to determine its new location and the location of surrounding objects each time it moves. This problem of a moving visual scene is not restricted to mobile robots. Any vision system which is not totally fixed presents this problem. Current systems [11] must use sophisticated software and highly calibrated hardware just to pan the camera over a scene. Hence, the usual solution [4][8][14] has been to simply make sure that every possible point of interest is within the camera's view, and only look at portions of the picture. While this method insures easy calibration, it leads to tremendous waste in picture resolution.

Finally, there is the problem of the moving robot in a dynamic environment. Here, not only is the robot's field of view constantly changing, but features may move within the field of view. If the robot cannot keep track of significant features in real time, the situation is virtually hopeless.

Vision Hardware

One of the fundamental problems of computer vision is getting a visual image into the computer f3]. There are two basic mechanisms currently in use: image dissectors and vidicons (TV cameras).

The image dissector, given its extremely high resolution combined with random access of any point in its field of view, at first glance appears ideal for digital computer work. Unfortunately, it is this random access ability that creates problems. If enough time is allowed to average the input at each input location (for a reasonable signal to noise ratio), there is no way to Input a picture to a computer in a reasonable amount of time [6], Hence, one cannot take advantage of the image dissector's abilities for real time vision.

The vidicon does not have the problem of input noise inherent in the image dissector. The input from a point in the field of view is effectively averaged over the entire frame period instead of only during the actual time the point is being interrogated. However, vidicons have problems of their own. Not the least is the available resolution. Standard vidicons are only capable of about 500 by 500 lines of information. While this at first seems quite high, if the camera is required to cover the entire field of view, by the time the field is partitioned to look at an object one twentieth the width of the entire field, resolution is decreased to 25 by 25 lines. Equally bad, brightly lit images in vidicons tend to spread (or bloom) over adjacent areas. This apparent enlargement of bright objects can be disastrous to many recognition schemes, and may even obscure nearby objects so much they cannot be seen at all. It is possible, through the use of a special "anti-comet-tail" gun [15] to greatly reduce blooming. However, this is only at the cost of eliminating the linear response characteristic of the vidicon, a step which introduces problems of its own.

In real time vision systems, it seems necessary to use vidicons for image input despite their limitations. Consequently, pan-tilt heads and zoom lenses are also desirable, unless the robot's visual abilities are to be restricted to objects which fill a relatively large portion of the total field of view.

3. The Basic System

Division of labor is used to obtain real time visual processing in a dynamic environment. If arbitrary features in a scene can be tracked in real time, most dynamic vision processes can be handled adequately by what would normally be considered off-line (not real time) methods.

A simple example is recognition and acquisition of a moving object by a robot. A typical approach would be to take a picture of the workspace and apply high powered heuristics to locate and identify the object. After deciding the desired object had been located, the robot would reach out and grab it, while the vision system continued taking and analyzing pictures to maintain knowledge of the current location of the object. Even for a simple domain of objects, the recognition algorithm would be complex. Not only would the size and orientation of the object be subject to variations, but the appearance of the hand in the same frame as the object further complicates matters [5]• Considering that most recognition programs require minutes of computer time [1119] \12]114], several orders of magnitude of speedup are required to make even this trivial example realistic.

By using a high speed feature tracker with no recognition ability, the same task can be accomplished quite simply. As soon as an object is detected in the field of view, two processes are initiated. The tracker is pointed at some feature of the object and instructed to keep track of it. Simultaneously, a picture of the object is given to a recognition program to identify. The two processes then run in parallel, the tracker taking pictures and updating the object's position, while the recognition program continues to grind away at the original stored picture. When the recognition program finally determines what it is looking at, the tracker uses this knowledge to aid its tracking, while the robot uses the continuously updated position information provided by the tracker to guide its pickup of the object. Since the tracker Is continuously following the object, confusion with the hand (as long as it did not obscure the object) is not a problem.

The crucial factor has been the development of an adequate tracker. The tracker is caught in a conflict between the requirement for real time feature location and identification (typically 50 milliseconds) and the necessity of accurately determining the latest position of numerous features while having only minimal knowledge of their characteristics. However, the application of a simple constraint allows the resolution of this conflict, and in fact requires a rapid frame rate to accurately identify features.

The simple example just discussed actually requires three discrete modules. Aside from the tracker and recognizer working in parallel, there is also a supervisor (or monitor) to coordinate everything and provide contact with the outside world (i.e. accept commands, take pictures and control the robot). The supervisor would also handle such tasks as locating new objects appearing in the field of view, determining the task to be accomplished, converting tracker coordinates to Robot (real space) coordinates, etc. The tracker does not burden itself with knowledge of the real world, instead it works with two dimensional features (blobs, patterns (2], lines [7][10]) in picture coordinates, a task possible in real time. Meanwhile, the recognizer only needs to work on one object at a time, and knows that when it finally succeeds, the supervisor will still know where the object is, since the tracker has been keeping track of it,

4. The Tracker

The design of a digital computer program to scan 60,000 picture points, find, and accurately identify a feature in that picture in real time would seem an impossible task considering a typical computer would require sixty milliseconds to look just once at every point in the picture. Brute force techniques are obviously not suitable. Since full pictures cannot be handled, all number crunching is done on windows extracted f rom fu11 pictures. Ail the software has been standardized to 36 by 36 windows. Windows are obtained by specifying their location and size (i.e. number of points in the full picture corresponding to a single point in the window). Since windows contain only 1298 points, complex operations can be applied to entire windows at reasonable cost in computer time.

Windows alone do not solve the problem. Assuming a window has been taken and a feature found, how can that feature be positively identified? There is no guarantee the feature being tracked is in that window. The identification of features problem has been resolved by a relatively simple constraint. The assumption is made that features are far enough apart and (or) predictable

enough in location to be distinguisable solely on the basts of position. While this may initially appear a severe constraint on the applicability of the system, it normally is not. Real objects have physical limits on velocity and acceleration, even when moving randomly.

Basically this means objects cannot collide, or in three dimensions, obscure one another. In the event of a collision, one of the features being tracked will be either lost or confused with the other (obscuring) feature. This situation cannot be handled by the Tracker, but the Tracker can warn the Supervisor that something is amiss and higher level help is required.

5. Demonstration Systems

To demonstrate the abilities of systems based on the method developed, three sample systems are presented. All of these have been implemented on the combined computer facilities of the Advanced Automation Lab in Coordinated Science Laboratory.

Traffic Monitoring

The program TRAFFIC, as its name implies, is a traffic montior. The camera is aimed out of the window (about 15 meters above ground level) at the roads outside the building. Figure 1 is a typical view, taken by the camera and reconstructed by the CSL graphics display unit. In addition to the crossroads, there are sidewalks and pedestrians, as well as numerous parked cars.

Moving objects are detected by the supervisor on the PDP-10. While the Tracker on the PDP-10 follows them, the Supervisor extracts a window containing just one object and transmits it to the PDP-11. There it is given to the Recognizer to determine its identity (i.e. car, truck, bicycle, etc.)„

Concurrently, a schematic representation of the intersection showing thelocation and status of vehicles being tracked maintained on the graphics display (Figure 2). Note that one car will be lost when it disappears behind the truck. Vehicles which have not yet been identified are displayed as "?-?", speed is indicated by the amount of exhaust.

Stacking

The applicability to hand-eye coordination Is illustrated by the program STACK. Several cubes are scattered about the Robot's worktable. These are located visually and stacked one at a time on the base cube (Figure 3). Visual feedback (using front and side views via the mirror) is used to guide the hand for precise alignment of the cubes.

The system functions quite simply. Ignoring the mirror, the approximate locations and orientations of the base and stacking cubes are determined visually. The robot then orients the wrist and grabs the cube to be stacked. As It moves the cube to over the base cube, it rotates the wrist to the estimated orientation of the base block, using care that the fingers do not obscure any of the vertical edges used for alignment.

The Supervisor then locates the front and right edges of the cubes and the mirror images of the front edges. The mirror provides the equivalent of a second camera for complete three dimensional determination of the scene. The correspondence between direct and mirror views need only be set up once, it is then maintained by the Tracker.

Once the Tracker settles on the edges, the Supervisor uses the pan-tilt head to center the stack and zooms in for a closer view (Figure A). The hand then moves parallel to the table top and the wrist rotates until the corresponding edges of both cubes are exactly aligned. The hand is then lowered while keeping the edges exactly aligned until contact is made. Note that once the edges of the cube are being tracked, there is no requirement for the base cube or vision system (mirror) to remain stationary, let alone for even mediocre calibration of the robot arm. Stacking accuracy is limited by the picture sampling width (252 points per line), in this case about 0.10 inch error with 4 inch cubes.

Insertion

INSERT has the capability of inserting the correct peg in an arbitrary hole where the hole is in a fixture held by an uncooperative human being (Figure 5). While the robot is moving to insert the peg, the human being is trying to prevent it (keeping the fixture flat on the table and not moving faster than the robot can).

The system is actually divided into three steps: detection, approach, and insertion. The detection phase Is straightforward. As soon as a hand appears In the field of view, the Tracker is set to follow it. Simultaneously, the Recognizer is invoked to determine the contents of the hand. The hand is tracked until the Recognizer has finished. Once recognized, the Tracker Is reset to utilize this knowledge and track the fixture itself. If the robot has a suitable peg for insertion, the approach phase is entered.

The approach phase uses the known and visual size of the fixture to estimate its position in three dimensional space. After the hand has retrieved the correct peg, it is moved to a position six inches above and behind the fixture. Care must be exercised by the robot during this time to prevent obscuring the fixture, and losing it.

Once in position, the hand mark is tracked, and the arm moved to always keep it in the same visual position relative to the fixture. The pan-tilt head is used to center the fixture and the Supervisor zooms In for a closer look. When the image becomes sufficiently large, the Tracker is shifted from the hand mark to the actual pin tip. When the camera is fully zoomed in, control Is passed to the insertion phase.

The insertion phase (Figure 6) uses the known vertical distance from pin tip to hole to determine the relative error in position. The robot moves to keep the pin tip over the center of the hole and correctly rotated (if necessary) for insertion. Simultaneously, of course, the

pan-tilt head is continuously adjusted to keep the hole centered. When the pin tip has been lowered to ten millimeters above the fixture, it Is inserted.

References

[1] Ambler, Barrow, Brown, Burstall, & Popplestone, "A Versatile Computer Controlled Assembly System," Advance Papers of 3IJCA.I, Stanford University, August 1973.

[2] Burr, D., - Personal communication.

1 3] Chien, R. T. & Snyder, W. E., "Hardware for Visual Image Processing," IEEE Transactions on Circuits and Systems, March 1975.

I 4] Ejiri, et al, "An Intelligent Robot," Proceedings of the 2nd Annual Joint Conference on Artificial Intelligence.

I 5] Gill, Aharon, "Visual Feedback and Related Problems in Computer Controlled Hand-Eye Coordination," Stanford AI Project Memo AIM-178, Stanford University, October 1972.

[6] Horn, B.K.P., "The Image Dissector 'eyes'," MIT AI Memo No. _178, August 1969.

[7] Hueckel, M., "An Operator Which Locates Edges in Dlgitalized Pictures," JACM, October, 1973.

[8] Minaky, M. & Papert, S., "Proposal to ARPA for Research on Artificial Intelligence at MIT," MIT AI Memo 245, October 1971.

(9] Munson, John H., "Robot Planning, Execution, and Monitoring in an Uncertain Environment," SRI AI Tech Note 59, May 1971.

[10] Smith, M. W. & Davis, W. A., "A New Algorithm for Edge Detection," 2nd international Joint Conference on Pattern Recognition, Copenhagen, Denmark, August 1974.

I l l] Sobel, Irwin, "On Calibrating Computer Controlled Cameras for Perceiving 3-D Scenes," Advance Papers of^ 3IJCAI, Stanford University, August 1973.

[12] Wolferts, K., "Special Problems in Inter-active Image-Processing for Traffic Analysis," 2nd International Joint Conference on Pattern Recognition, Copenhagen, Denmark, August 1974.

[13] Yaklmovsky, Y. & Feldman, J., "A Semantics-Based Decision Theory Region Analyser," Advance Papers of 31JCAI, Stanford University, August 1973.

[14] "The XQ 1080-series of One-inch Diameter Plumbicon Tubes," Product Information No. 38, N. V. Philips, Eindhoven, the Netherlands, 15 May 1971.

Figure 1

Figure 2

Figure 3

Figure 4

Figure 5

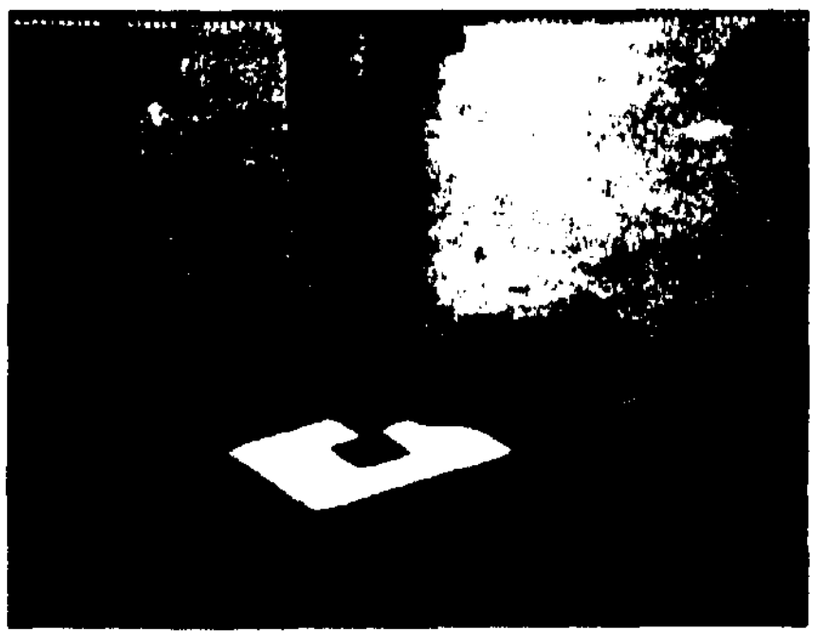

Figure 6