MASSACHUSETTS INSTITUTE OF TECHNOLOGY

ARTIFICIAL INTELLIGENCE LABORATORY

Working paper 106 July 1975

Visual Tracking of Real World Objects

Glen Speckert

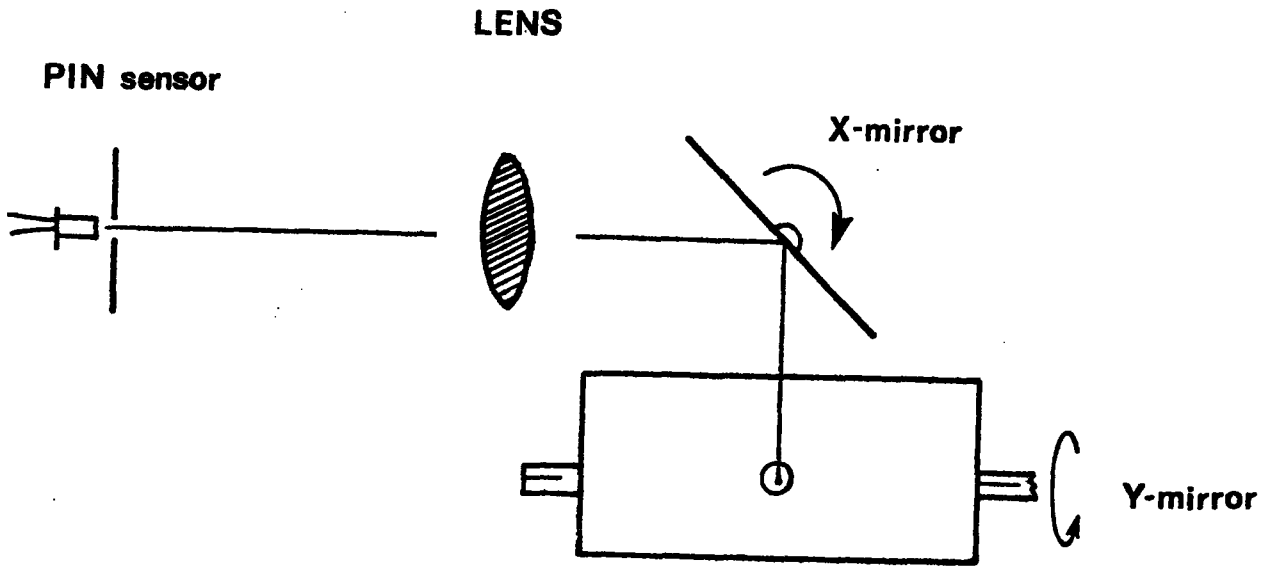
This paper describes the progress made towards tracking an object visually using a PIN diode attached to a dual mirror deflection system which enables the PIN diode to "optically point" to any position in two-space. A helium neon laser equipted with a similar mirror deflection system was used to point at the object being tracked. Actual objects tracked include a hand, a bouncing ping pong ball, and a white center on black target attached to a moving metronome.

This report describes research done at the Artificial Intelligence Laboratory of the Massachusetts Institute of Technology. Support for the laboratory's artificial intelligence research is provided in part by the Advanced Research Projects Agency of the Department of Defense under Office of Naval Research contract N00014-75C-0643

1. Hardware Characteristics and the Problem of Focusing

The hardware used in tracking should be fully understood before proceeding to the tracking algorithms. The hardware consists of a PIN diode, a camera lens system, and a mirror deflection system rigidly mounted in a frame. The camera lens system consists of a manually adjustable iris with f/stop range of f/3.2 to f/22, and a manually adjustable focus. The diode points directly (through the lens system) at a 1 inch by 1 inch mirror, whose axis of rotation is perpendicular to the axis of the diode's lens and parallel to the floor. Movement of this mirror (by currents from digital-to-analog converters under program control) enables the diode to "see" objects along a line below and in front of it. Similarly, if your head was stationary, and you had one mirror, you could see (looking in the mirror) your eye, chin, body, feet, leg, and the ground in front of you as you rotated the mirror. The "straight down" view (mirror at 45 degrees to the horizon) is chosen as the center of this range. Now assume that you have a second mirror directly below the first whose axis is along the line of sight visible from the first mirror. This mirror is physically 1 inch by 2 inch. By moving this mirror, the PIN diode can see a vertical line from each point on the second mirror that can be viewed from the first mirror. Hence the PIN has a full field of view limited only by the size of the mirrors, and the maximum deflection angle

of the mirrors. For the PIN diode this amounts to a 30 degree cone of vision. "Seeing" for the PIN diode means returning a value (through analog-to-digital converters) porportional to the intensity of light that has fallen on it in the last millisecond. See diagram.



The currents to drive the mirrors comes from a pair of scanner controllers which receive their input voltages from digital-to-analog converters whose digital input is supplied by the program. There also exists a second set of drivers attached to a smaller mirror deflection system which is rigidly attached to a helium neon laser. By placing the

laser physically close to the PIN diode, the laser can be made to point to approximately the same point as the PIN diode is looking at, so that the user can visually monitor what the program is actually watching. Unfortunately, the laser and PIN diode can only be kept "in synchronization" over a limited region due to the non-linear differences in the two mirror coordinate systems, and due to their physical separation. Also, the field of view of the laser is only about 2/3 that of the PIN diode due to the smallness of the laser mirrors. The laser mirrors are about 1 cm by 1 cm, hence are quicker (less inertia) but have a smaller cone of view, or range. The laser can also be set up to project on the wall the position that the program is viewing.

The major advantage of the PIN diode system is its ability to access randomly any position in approximately the same time. The mirrors move from one position to another in about 3.3 msec with no overshoot. Most of the movement occurs between .5 and 2.5 msec after the mirrors are "told where to go." The mirrors can also be pulsed without waiting for them to settle, or even waiting for them to reach their final position. Their position can be read back through appropriate analog to digital channels in order to know where they actually are. This method gives up precise control of position in exchange for speed. The mirrors on the laser react twice as fast, but this doesn't affect tracking speed or other characteristics of tracking, since the laser is not used for

illumination of the target, rather merely as a visual display of what the program is viewing. In fact, when tracking at low illumination levels (single 100 watt bulb), the intensity of the laser beam occasionally being in the point view of the PIN diode causes some points to be illuminated by both the laser and the other light source, while others are illuminated only by the outside light source, thus causing the dynamic thresholds to lose. With a brighter light, the intensity of the laser beam becomes insignificant, and this problem disappears. To actually use the laser as an illumination source is certainly possible, but only if it is rigidly attached to the PIN diode, and most probably only if it uses the same set of mirrors. Also, one must be wary of the fluorescent room lights that flicker at 60 cycles, if they contribute substantially to the illuminance of the target. For a better understanding of the hardware, see Working Paper 98 by Berthold K. P. Horn (June 1975).

The next problem is focusing. Time between points is not a factor. Intensities taken from adjacent points (with ample time between points) while scanning across a sharp white-black transition should, when plotted, be a step function. The number of points between the high and low plateau determines how well the system is in focus. Thus to focus the system, I have a routine which finds a black-white transition and steps across it, typing the intensities to the terminal so that the user may adjust the focus if it is not in focus. From experience, there will be at

least three points between the "high plateau" (white area) and the "low plateau" (the black area) even at best focus, possibly due to the resolution of the system. If there are more than this, the diode is not in focus, and the user can adjust accordingly. A possible better focusing system would be to mount the diode (or diode array) in the back of a single reflex camera through which the user can see the actual image which will fall on the diode(s), and can focus accordingly. Marc Raibert is currently experimenting with a 32 x 32 diode array, and is attempting to mount it in the back of a camera for ease in focusing. It would be even better if this camera mirror was half-silvered and remained down (unlike a real single reflex camera mirror which changes position to allow the light to hit the film). In this case, the user could look through this half silvered mirror and watch what the program is viewing. A computer adjustable focus would be a big feature, as the target currently can become out of focus by moving towards or away from the tracking device. This defocusing does not cause the tracker any difficulty, but high position precision is impossible without a sharply focused view. Understanding the hardware, one can proceed to the tracking algorithms.

2. GEM, The Trailing Edge Follower

In an effort to develop a quick tracker, the target I chose to have most promise was a white circle (or other shaped blob) against a black background. The entire background need not be black, just an area which completely encircles the white center of the target. Another target which holds promise is a circle which is divided into black and white alternating pie shaped wedges (similar to a surveyor mark). See the final section of this paper for information relating to the surveyor mark target tracker. Maximum allowable tracking speed seems to be a good measure of how good a tracker is, since all methods can track (and never lose the target) at *some* characteristic speed. Also speed can be traded off for other desirable characteristics, such as smallness of target, precision in position, or intensity of light needed. Thus a tracker can be fairly well characterized by how fast it can track.

The scanning pattern consists of just looking at the intensities of four points arranged in a diamond (hence the name GEM), and deciding what to do based on when, how often, and which ones are white or black.

To acquire the target initially, there are several options. The program could scan the entire field of view, starting from the center, say, and thus would latch onto the first target (first white point in this

case) that it found. This is hardly what is wanted. Or the user could be required to supply the co-ordinates of some point on the target (any point will do --see below). However, this may be impractical to even the experienced user, due to changes in the calibration of the drivers due to long movements or other users, or due to movement of the PIN diode itself from session to session. Thus some sort of limited search based around an input position seems to be the answer. I use a rectangular grid search covering an area of about three inches square. Having the laser (with its own mirror deflection system) located physically close to the PIN diode, and calibrated to point to approximately the same spot that the PIN diode is "looking at" is a tremendous boost in helping the user locate the target he wants. The user gives the program a position, watches to see what the program is looking at (see where the laser scans), and if no target is found, the user can easily refine his guess, knowing both numerically and physically where the program searched previously. One procedure for initial acquisition of the target is to give the program a set of co-ordinates, with the light source off, watch to see if the laser search touches the target, and if so, turn the light on and give the same co-ordinates. This initial search has no time constraints on it, so the program can verify (by viewing several points) that what it is viewing is a possible target, and not just a noisy point. To verify, GEM looks at four points near the possible target, three of which must be white for the

verify to succeed. GEM then finds the radius of the target (see below) and returns it to the user as a further check on correctness of the target (the user should have an idea how large the target he wants to track is). If the laser is properly positioned and adjusted, the user can look and see exactly what the program is viewing. This method is for picking up stationary targets, since it takes the time to verify its targets rather carefully.

The brute force method of tracking for this target consists of surveying each point, and if the top point goes black, drop the scanned diamond down some amount (a parameter of the program --currently tied to the radius of the target. About 1/4 to 1/2 of the radius of the target works well). Similarly, if the left point goes black, move the pattern right, etc. This is essentially a series of replacing the scan back on target whenever the target moves. At this point, GEM uses no high level knowledge about the target. However, "knowing" that the target may move to areas of greater or lesser intensities implies the need for a dynamic threshold rather than a static one. Thus I established 1/3 the maximum intensity encountered in the last 20 scans (80 points) as the threshold. Thus even if lighting changes (assuming it doesn't change drastically in 1/10 of a second), GEM has no problems.

I began at this point to track a white spot about one inch in diameter on the tip of a metronome against a black background. I chose

a metronome because it gave a measurable velocity that I could compare my progress against. The brute force tracker tracked at 72 beats per minute, or *Larghetto*, where each beat the target travels about 15 cm and changes direction.

Note that although this is a trailing edge follower (i.e. the GEM scanner isn't in the center of the target), the center and can accurately be located and the radius can be found once the target is stopped, or moving slowly compared to the time needed to find the center. This is done by taking a point known to be on the target, scanning left until darkness appears, then scanning right until you find darkness, bisect that line, similarly finding the top and bottom of a line passing through the bisected point, and bisecting that line to yield the center. Half the length of that line is then the radius. Graphically:

1) Given a point inside circle
2) Scan left to A
3) Scan right to B
4) Bisect AB to get C
5) Scan down from C to get D
6) Scan up to get E
7) Bisect DE to get the center
8) Return radius and center

This algorithm works on ellipses whose major axis is at any angle if it is iterated 3 or 4 times (The target appears as an ellipse if it is tilted). Thus the GEM tracker can follow the edge with some imprecision in position until the target stops, then accurately find the

center. This algorithm also determines the radius of the target, thus should be used upon initial acquisition of the target, since the parameters of how far the four points are separated and how much to shift the scan pattern each time a point goes black should be related directly to the radius of the target.

To improve the GEM tracker, I incorporated higher level knowledge, specifically X and Y velocity and X and Y pseudo-acceleration vectors. When I get a "hit" (a point goes black) on a "bumper" (one of the four scanned points), say the bottom bumper, I add an impulse vector (whose size is a parameter) to the Y pseudo-acceleration vector. Then I add the acceleration to the velocity. Thus a second hit on the bottom bumper changes the velocity more than the first one (if the top bumper is not hit in between), and the third changes the velocity even more. This isn't quite what is wanted, since hits widely separated in time should affect the velocity about the same, but repeated hits *should* change the velocity more with each succeeding hit. Thus I put a linear decay on the acceleration vectors, which is strong enough to decay out minor accelerations widely separated in time, yet which allows rapid changes in the velocity if the hits are consecutive. If there is a hit in the direction opposite to the direction of the acceleration vectors, the acceleration vectors are cleared before the impulse is added, i.e. the acceleration can instantaneously change signs. In addition to changing the velocity, I also use a brute

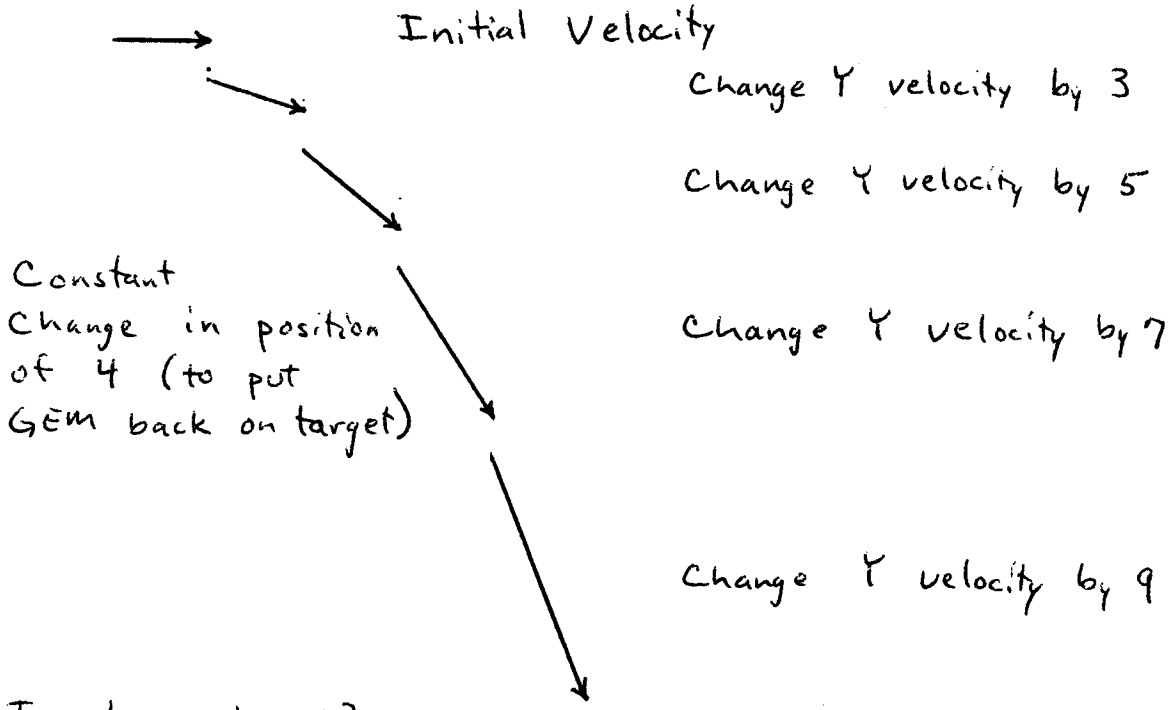
force type shift in position to put the scan back on the target. The size of the acceleration impulse and the amount of decay are fairly important parameters, hence I did a search for the best value of these parameters. The parameters I use are impulse size of target radius/8 + 1, and decay of target radius/16 + 1.

The position of the diamond scanning pattern is, of course, continuously updated via the velocity vectors. Thus the scan pattern experiences a "discontinuous" change in position, velocity, and acceleration each time it gets a hit, and when no hit occurs, the position undergoes a "continuous" change and the acceleration decays "continuously" (velocity remaining constant). A plot of position with velocity vectors shown would look something like figure 2.1.

Notice that the amount that the velocity vector changes is a function of the history. Notice, too, the constant discontinuous change in position required to bring the tracker back on target. Using this method, the metronome can be tracked at 160 beats/minute, or *Allegro*. Note that introducing higher level knowledge more than doubled the tracking speed. A ping pong ball is also a good target, so I tried to follow one, and was able to follow it until it hit the table, where I overshot and lost it.

The knowledge that the tracked object is likely to stop suddenly, or even reverse its direction can be used to prevent this overshoot problem. The tipoff to GEM is when only one of its four bumpers

Changing Velocity under Successive Hits from top bumper



Impulse size = 3
Decay size = 1

Acceleration = amount to change velocity in the event of a hit

= Acceleration last − decay + Impulse (if hit)

TIME	ACCELERATION	VELOCITY	POSITION
0	0	0	0
1	3	3	7
2	5	8	19
3	7	15	38
4	9	24	66

Figure 2.1

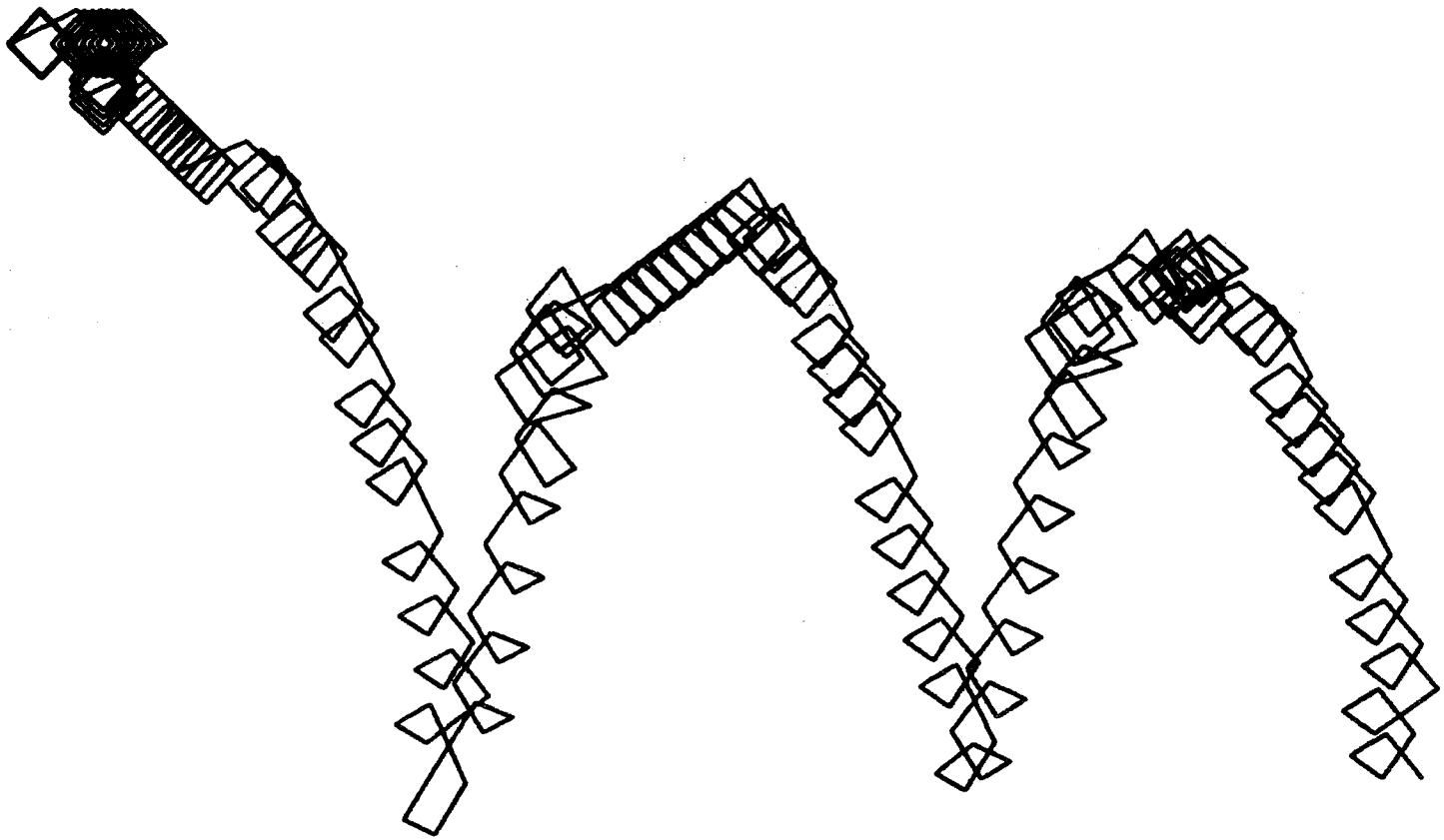
is white (a "hard hit") Only one white point implies either the target is moving much faster (but in the same direction) than anticipated, or it has possibly stopped or even reversed directions. Looking at which point it is that is white and knowing the previous velocity allows one to identify one of these cases.

If the target appears to speed up suddenly, the acceleration vector is modified twice, and hence the velocity gets a strong increase. Also the scan pattern is moved double the normal amount to bring it on target (instantaneous "catching up").

In the other case where the target appears to have stopped, again the scan pattern is moved twice the normal amount to bring it back on target, and the velocity vector gets stored away, then cleared. Thus the target is assumed (momentarily) to have stopped. An immediate second "hard hit" (caused by the target actually having reversed direction, and not stopped) causes the saved vector to be negated and re-instated, hence, if the target stops in 1/100 of a second, GEM won't overshoot, and if it bounces in 2/100 of a second, GEM can follow its bounce "without ever suspecting" that it would bounce, knowing only that the target is capable of bouncing. With this added, GEM can track the metronome at 210 beats/minute, or just past *Presto*, or as fast as the metronome can move. It is reasonably difficult to follow the target with your own eye at this pace (especially if you did not know that it could only go back and forth).

In order to still be able to use the metronome, I traded off speed for a smaller target, the new one being the size of a dime (the old one was the size of a ping pong ball). Using this technique the smaller target tracked at 160 beats/minute (*Allegro*). To my dismay, an actual ping pong ball falling reverses direction in less than 1/50 of a second upon hitting the table, and thus GEM was not quite able to follow it bounce. Had it meerly stopped, and not bounced, GEM would not have overshot it, as it was able to stop in time, but not able to reverse. However, this feature did allow GEM to follow most hand movements and quick stops.

In order to follow a bouncing ping pong ball, I had to incorporate even more higher level knowledge, specifically GEM had to have a model of what the ping pong ball was expected to do when it hit the table, and knowledge of where the table was located. When the initial target searcher finds the target, it asks the user if the target is resting on the table. If so, GEM remembers that Y value and will never attempt to track below it. If the target is not on the table, the user must supply a value for the table (the user can supply negative infinity if he doesn't want a table). Once GEM knows where the table is, it "knows" not to search below the table, so upon tracking the target to the table, it reverses the Y velocity and reflectes the Y position. Using this, GEM was able to follow the metronome with the small target at a speed of 184 beats/minute, or about four times as fast as without any high level knowledge, by



Locus of points viewed while tracking a bouncing ping pong ball.

defining the "table" to be the lowest point in the arc of the metronome. This knowledge also enabled GEM to follow a ping pong ball bouncing with no trouble. See figure 2.2 for a plot of positions viewed while tracking a bouncing ping pong ball. Bounds for the right and left walls can be added easily, but seem not to prove anything. I next proceeded to form a strategy for action to be taken upon losing the target.

3. Spider Web, Flypaper, and the FBI

The next step in GEM's development was to have a strategy for recovering from a lost target. There seem to be at least three approaches which are interrelated, but which have differing advantages and disadvantages. The first method I call the FBI approach.

Assume that a criminal is being followed, but eludes his followers. A ring can be set up around the point where he was last seen, and when he comes out, you nab him. Similarly, one can scan a circle about the last known position, then wait for any scanned point to change its intensity *relative to what it was before*. If it changes from a low value to a high one, the target is just coming through the circle. If the intensity changes from a high to a low value, the target just slipped past, so "chase" it for a second then "nab it" (i.e. the initial scan which is

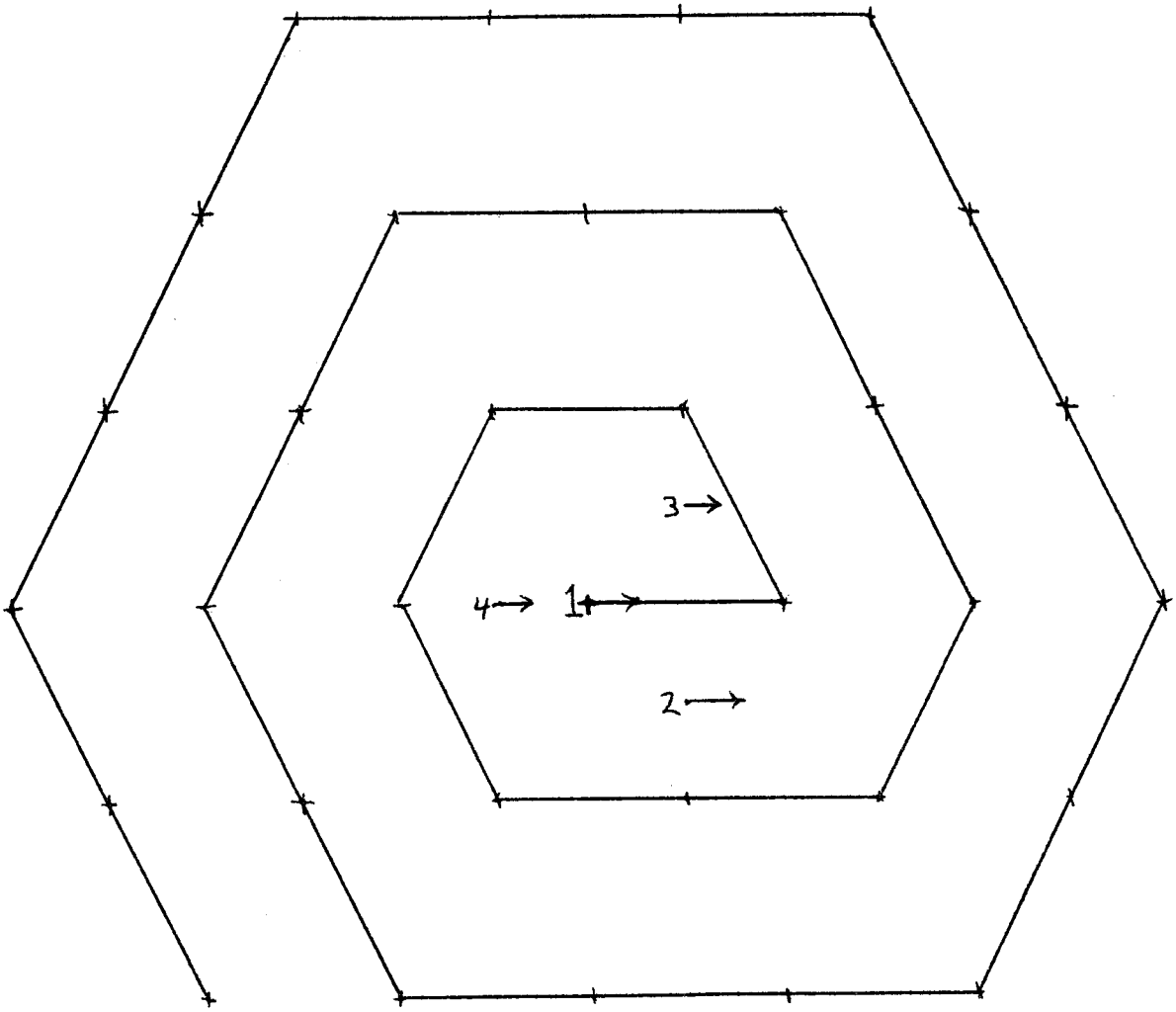
compared against actually picked up the target). This method has the real advantage that it *makes no assumptions about the background intensities*, only that it is stationary, since it compares intensities against previous values. It is quite possible, though, that the speed required to make the scan is too slow to be sure of catching the target (time of one scan of entire circle must be less than time required for target to move its own diameter), or that the time required to "set up the roadblocks" (time to obtain initial set of intensities against which to compare) may be too great. If the search is spiraled inwards, intensities can't be compared against a previous value, and a black background must them be assumed. This inward spiraling search is similar to an inverse Spider Web search (see below).

The second approach is the "flypaper" approach. Since you are trying to catch the target "on the fly", the problem is similar to trying to catch a fly. The flypaper method is simply the FBI method, except that only a part of the circle is scanned, and hence can be scanned more often. Flypaper is by definition scanned fast enough so that moving targets can not penetrate it undetected. Where to put the flypaper is dependent on the last known acceleration and velocity as well as position of the target. Two or more pieces of flypaper can be used, if time permits. This has the property of being more dense than the FBI method, but obviously, the fly can avoid the flypaper.

The third method is the Spider Web approach. This consists of an increasing spiral search, starting at the point where the target was last seen. This spiral is fairly sparce, because it must have the property that it gets a great distance away from the lost point as quickly as possible. GEM uses a hexagonal web with one target radius between points. If the first spiral fails to locate the target, a second one is started, offset by a distance of one target radius, followed by a third and fourth. These four scans form the basic web, and together cover a rectangular grid with points one target radius apart. See figure 3.1. GEM actually uses two webs (eight total different spirals) shifted by half the target radius to give a grid with spacing of one-half the target radius. In practice, when GEM detects a "white" point, before assuming that it has found the target, it does a quick verification by sampling two more points, one of which must be white or the initial hit is assumed to be noise and the spiral search continued (Note that this verification is quicker and less sure than that used to find a stationary target). The advantages are speed and distance from lost point covered, but the disadvantage is the sparcity of the web.

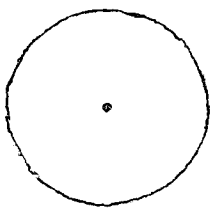
GEM uses the web, and will keep the web "up", i.e. will search around the last known position for about 10 seconds before giving up hope, hoping that the target will return. Also, GEM reflects the web about the table, if necessary, thus giving double strength near the table.

Web Search Used by GEM



Single Strand of web (.07 sec)

Four strands make a complete web



Target Size

Using the Web, GEM can track the small target on the metronome at full speed (210 beats/minute). From an actual tracking session of about five minutes of tracking the metronome, GEM encountered 107 "hard hits"(only one white point), had to make 13 Web searches, from which 12 times it recovered on the first spiral search, and one time it recovered in the first Web. It never had to go to into the "wait mode" of holding for long periods of time. The single spiral takes .07 sec, and a complete web takes .28 sec, thus GEM usually recovered in less than .07 seconds.

Demonstrations of the system lead to some interesting items. Cutting off the light with your hand while the target (say on the tip of a metronome) is moving causes GEM to go into a Web search, in which it usually finds the target immediately. Turning off the light enables the user to watch the laser form the Web and hold it. When the light is turned back on, the metronome is picked up as it sweeps through the Web. Also, a ping pong ball can be dropped through a web in wait mode, and is picked up before it hits the ground, then is followed as it bounces. The Web can be used as a wide initial search, if there is only one target in the Web, as it will latch on to the first target it finds. Note that this is not as good an initial search as the close knit rectangular search that is currently used for initial acquisition of the target, because it does not verify the targets it finds as well (verification takes time). When

picking up the ping pong ball, if dark gloves are not used, GEM sometimes tracks your hand, which is easily detectable by the laser spot on you hand, or possibly crawling up your arm. It will stay there through moderate hand jerking, and all normal hand waving or other normal speed hand movements. A piece of black paper can be used to "push" the laser beam down your arm, along your finger, and onto the target, exactly as if the laser beam were an insect or other small being.

Thus the GEM tracker has high promise as a fast, simple tracker, and can follow a ping pong game (Three trackers could watch a game, one watching the ball, and one watching each paddle in order to know when to expect the ball to change directions). The GEM tracker is quite insensitive to tilt of the marker, and I defocused over the entire range while it was tracking the moving metronome, and it did not lose the target. A set of 4 different sized aperitures allow the user to trade off resolution and light intensity required (smaller aperature means more resolution, but needs more light for the same target). Thus it appears quite reliable under many and varied adverse conditions. The tracking speed is fast enough to follow all normal hand movements, and even moderately quick hand jerks. It can follow a bouncing ping pong ball, a rapidly moving metronome, and other moving targets. Thus the GEM scan method appears to be a good tracking algorithm.

4. The Surveyor Mark Tracker

The GEM tracker is essentially a modification of a scanning pattern consisting of two crossed lines tracking a target which is a circle. Note that with two full lines instead of just four points, the center of the target can be accurately found each scan. Similarly, if the target has these crossed lines on it (as does a surveyor mark) and the scanning pattern is a circle, the center can be found each complete scan. This leads to the other basic type of target, the surveyor mark.

The general idea is to scan a circle which contains the center of the target. The intensities read back should then form a square wave (see figure 4.1). By finding the transition points, say A, B, C, and D, one can then locate the center fairly accurately. Since this requires having at least 40 intensity values to play with, the obvious method of waiting 3.3 msec between each point becomes vastly too slow. However, by waiting only a fraction of a millisecond between sending the mirrors new co-ordinates, it becomes possible to read the position and intensity without waiting for the mirrors to get to their final position, i.e., the mirrors lag behind the sent position by some amount, but since both *position and intensity* are read back, this lag does not impair the tracking ability, as the intensity is the actual intensity at the point read back.

6. Future Directions

I believe that I have shown that either type of target can successfully be tracked, with the GEM target being simpler and quicker. It also adapts itself more easily to a multiple diode tracking device.

A better hardware device would be a single reflex camera (so that the user can focus and possibly monitor what is being viewed if a half silvered mirror were used) with a computer controllable focus, to allow dynamic focusing. Multiple diodes would be arranged in the back of the camera in a pattern which is more dense in the center, for example, two concentric diamonds, or if the surveyor marker is to be tracked, concentric rings. With an array similar to the one Marc Raibert is developing, a full cross could be scanned instead of just four points, thereby having the advantage of being a center target tracker, instead of a trailing edge follower. In any case, multiple diodes would certainly speed up tracking, and the extra time can be used to check the focus, monitor more than one target, or look ahead of the target to see if another object is about to hit it (i.e., a ping pong paddle about to hit the ball). More than one target can be differentiated by differing number of sections for the surveyor marker, or different size or different color targets for the GEM

tracker. Also differing number of rings could be put around differing targets for the GEM tracker.

Two trackers can be co-ordinated to track an object in three dimensions, and eventually tracking devices can be as reliable as the human eye, or even more reliable.

--Glen Speckert