Exploratory Motion Generation for Monocular Vision-Based Target Localization[1]

Eric W. Frew, Stephen Rock
Aerospace Robotics Lab
Stanford University
Stanford, CA 94035
650-723-3608
ewf@sun-valley.stanford.edu
rock@sun-valley.stanford.edu

Abstract— This paper presents a trajectory generation algorithm for the exploratory motion of a monocular vision-based target localization system. The planning algorithm uses the Fisher Information Matrix (FIM) to quantify the information presented to the localization subsystem. Maximizing functions of the FIM minimize the best achievable error bounds on the target estimate. Current approaches solve this optimization process by assuming the object location is known. This assumption is paradoxical as the goal of the overall system is to estimate the location of the unknown object. Therefore, an iterative approach has been developed. Furthermore, current approaches maximize the information content for a given path length or time. The new algorithm presented here minimizes the path length necessary to achieve a specified level of information while incorporating other motion constraints such as range limits, viewing angles, and safety margins based on the predicted estimate covariance.

TABLE OF CONTENTS

1. INTRODUCTION

Human-guided robotic construction and assembly tasks require highly capable autonomous field vehicles. One of the fundamental capabilities of these robots is the direct sensing of unknown, unmodelled objects in the environment. These objects must be detected, identified, tracked, mapped, and modelled so that the robot can avoid, manipulate, or reason about them.

One solution to the problem of object localization, determining the global position of an object, can be achieved with a single camera by fusing image track data from a single feature with camera motion measurements. Such a system takes advantage of sensors already expected to be on a human-guided robot (motion sensors that enable navigation and cameras that provide situational awareness) and therefore requires little additional payload. Furthermore, a single camera solution can be used to add redundancy and fault tolerance to current systems that use multiple cameras, combinations of cameras and scanning lasers, or stored environment models [1,2,3].

The monocular vision based localization estimation problem has been well studied, especially in the context of passive sonar [8,9,10]. The key features of this problem are that the object location is unobservable at any given instant in time and that camera motion under some constraints is needed to obtain a solution. For example, the estimation is unsolvable for the case of a stationary camera or one heading directly toward the object. Moreover, the performance of the estimator is a strong function of the camera trajectory. Therefore, the camera motion can be used as input into an optimization process that seeks to maximize the capabilities of the estimation subsystem.

This paper will present a trajectory generation algorithm that attempts to optimize the information content provided to the estimation filter. The problem will be formulated in a way that is suited to the needs of autonomous vehicles working with real world objects. Experimental results demonstrating the solution on a micro-rover testbed will also be given.

Related Work

There are several related areas of work that attempt to solve the target localization problem using various sensor combinations with different constraints than the technique presented here. These areas include Simultaneous Localization and Mapping (SLAM), Structure from Motion

[1] 0-7803-7231-X/01/$10.00/

(SfM), and Bearings Only Tracking (BOT). In each case motion is required in order for the algorithms to perform. However, the main issues vary significantly among them.

The goal of Bearings Only Simultaneous Localization and Mapping (BO-SLAM) is to enable autonomous navigation and mapping in an unknown environment using bearing measurements to fixed landmarks, odometry, and inertial sensors. These methods are a subset of the more general SLAM field which also uses range sensors to increase the available target information [3,4]. The main distinction of the SLAM algorithms is that they do not require knowledge of the camera motion. Instead, this information is part of the estimation state. However, in order to calculate the camera motion and target location simultaneously, multiple targets must always be tracked. Furthermore, in order for the map to stay consistent over time, any new landmark must be incorporated into the estimation process causing the size of the filter to grow rapidly. Current work in this field focuses on the data management needed to allow many landmarks to be mapped [5]. The solutions assume appropriate motion and distribution of landmarks. Any attempts at optimality are made by selecting advantageous landmarks to track.

Structure from Motion (SfM) techniques use batch and recursive algorithms to calculate camera motion and object shape (relative position between camera and each feature) from a sequence of images [6,7]. Unlike the SLAM techniques which use the odometry and/or range measurements to ground the solution in a global frame, the SfM solution is only good to an unknown scale and transformation. Furthermore, a necessary condition for SfM solutions is a large number of object features that can be tracked across the image sequence. In contrast, if the camera motion is known, a single feature is sufficient to yield a position estimate.

Features tracked using computer vision are transformed, projected bearing measurements. The problem of Bearings Only Tracking (BOT) has been studied extensively in the context of passive sonar and radar applications [8,9,10] and these techniques are being brought into the computer vision arena [11]. The nonlinear estimation presented in this paper is based on BOT algorithms. The key benefits from the known camera motion are that the estimation state size is reduced, a single feature or landmark can be mapped, and each new landmark may be estimated by a new filter causing the total estimation to grow linearly as opposed to exponentially. The BOT literature contains several different formulations of the basic localization problem and considers stationary as well as moving targets. [8-14]

Additionally, the trajectory generation problem for monocular vision localization is shared by BOT and the techniques presented in this paper are derived from their concepts [12,13,14]. There is a sizable body of work investigating this problem for various camera and target motion models. In every case the location of the target object is assumed known and the optimal path is derived for a given length of time. In contrast, this paper will address the paradox of the unknown target location and will investigate a different formulation of the path planning problem.

The trajectory generation problem as stated above is a sub-problem of the broader field of optimal input or optimal experiment design. The literature of this field recognizes the paradox of the dependence of the optimal solution on the actual parameter value to be estimated [15, 16, 20]. Solution techniques that address this issue involve static, batch sequential, or fully sequentially adaptive experiments that are locally optimal (relative to the unknown parameter value) [20]. The approach presented in this paper closely resembles the batch sequential method.

The related target localization methods presented above assume either appropriate camera motion or distribution of landmarks or features. For the case of BOT where the path is considered, the solution still makes several assumptions that are not applicable. Most notable is the assumption that the target object location is known. This paper presents a solution to the trajectory generation that eliminates this assumption and includes other realistic constraints.

2. Monocular Vision Target Localization

The global location of a target object can be determined from a track of the object in the camera image plane and the known motion of that camera. In the context of this paper, an object is represented by a single feature in the image. This situation can occur when objects are far away and only represent a point on the image plane or when they are very close and only a single feature can be

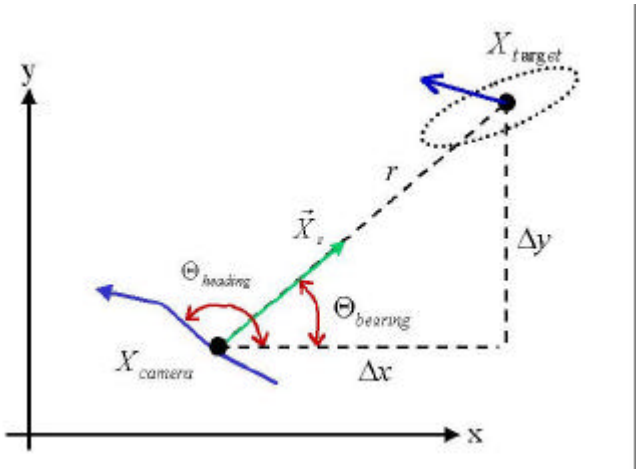

Figure 1. Two-dimensional monocular vision based localization problem geometry

distinguished.

Although monocular vision-based target localization applies to a full 3-D situation, we will restrict the discussion to the 2-D planar case in order to simplify matters. Figure 1 shows the problem geometry.

There are many different ways to formulate the estimation problem. We use the simple Cartesian coordinate system and only consider stationary objects, leading to a state vector

$$X = X_{target} = \begin{bmatrix} x_{target} \\ y_{target} \end{bmatrix} \quad (1)$$

The state equation is then

$$X[k+1] = \Phi X[k] + \Gamma w \quad (2)$$

with

$$\Phi = I \quad \Gamma = \vec{1}$$

and w is zero-mean gaussian white process noise with covariance matrix $R_{ww} = E(ww^t) = s_{ww}$

The measurement equation is

$$Z = h(X[k]) = f * \left(x_s / y_s \right) + v \quad (3)$$

$$X_s = \begin{bmatrix} x_s \\ y_s \end{bmatrix} = T_{w2s} * (X - X_{camera})$$

$$T_{w2s} = \begin{bmatrix} \cos(\Theta_{heading}) & -\sin(\Theta_{heading}) \\ \sin(\Theta_{heading}) & \cos(\Theta_{heading}) \end{bmatrix}$$

where v is zero-mean gaussian white noise with covariance matrix $R_v = E(vv^t) = s_v$.

The localization solution will utilize an Extended Kalman Filter (EKF) because it is capable of running in real time and provides a measure of certainty of the target position accuracy. Because the EKF is a linearization of the nonlinear problem, the covariance matrix supplied by the filter is an approximation of the statistics of the estimator error. However, it still yields useful information about the accuracy of the solution.

The EKF linearizes the measurement equation about the current state estimate. The resulting measurement matrix for this problem is

$$H(\hat{X}[k]) = dh/dX \Big|_{X=\hat{X}} = \left[1/y_s \quad -x_s/y_s^2 \right] T_{w2s} . \quad (4)$$

Given the state equation (2) and the linearized measurement matrix (4), the target localization algorithm uses the standard Kalman filter equations [18]. These equations will not be presented here.

It is important to note several facts about the filter above. The linearized measurement matrix H is a function of the camera motion X_{camera} and the target location X_{target}.

This first dependence shows where the necessary camera motion enters the estimation problem. Furthermore, this dependence may be exploited in order to improve the performance of the estimation.

The second dependence, on the location of the target, will introduce a paradox into the motion planning. As will be discussed in the next section, the location of the target must be known in order to judge whether a maneuver will be observable. However the whole purpose of the maneuver is to enable estimation of that target position. This problem must be addressed by any trajectory generation algorithm.

3. TRAJECTORY GENERATION

Because the monocular vision based localization is nonlinear and depends on the motion of the camera, it is possible to maximize the information provided to the estimation filter by selecting the best trajectory. This section will describe the necessary steps for the optimization process. First, an optimization cost will be defined. Second, the free variables and the camera motion model are given. Next, several important constraints are described, followed by the specific optimization method. Finally, an iterative step is presented to address the uncertainty of the target location.

Optimization Cost Function

The Fisher Information Matrix (FIM) is a representation of the sensitivity of a system to certain parameters within that system. For a nonrandom parameter, x_0, and an unbiased estimator of that parameter, $\hat{x}(\Theta)$, where Θ represents the measurement set, the Cramer-Rao Lower Bound (CRLB) relates the estimation error covariance matrix to the FIM such that

$$P_x = E\left[(\hat{x}(\Theta) - x_0)(\hat{x}(\Theta) - x_0)^t \right] \geq J_f^{-1} \quad (5)$$

where J_f is the FIM and is given by

$$J_f = -E\left[\frac{\partial^x}{\partial x^2} \log p_{\Theta|x}(\Theta \mid x) \right] = \sum_{k=1}^{N} \Phi^t H^t R^{-1} H \Phi \quad (6)$$

and $p_{\Theta|x}$ is the conditional probability density function.

The quadratic form of the error covariance matrix P_x defines an ellipsoid describing the distribution of the error. The one-sigma area of this ellipse is expressed as

$$A_{1s} = p\sqrt{\det P_x} \tag{7}$$

Relating this area to the FIM

$$A_{1s} = p\sqrt{\det P_x} \geq \frac{p}{\sqrt{\det J_f}} \tag{8}$$

In certain circumstances, with moderate sized measurement errors and appropriate paths, the maximum likelihood estimator of the target state is Gaussian, unbiased, and approaches the CRLB [11]. Other estimators and other circumstances will not necessarily approach the bound. However, seeking to optimize based on the FIM tells us the best possible path and the estimator must be designed to extract a solution from that information. Thus we can separate the path generation from the estimator design and still present an optimal plan. Maximizing the determinant of the FIM will minimize the best possible area of uncertainty about the position estimate.

Previous work has focused on the problem of minimizing the determinant of the FIM for a given time or path length. In many cases an autonomous vehicle needs a certain level of accuracy, which it knows beforehand, in order to interact with an object. Also, field vehicles tend to be limited by fuel or available time allowed to achieve a task. Therefore we would like to invert the problem and optimize the time it takes to reach a given uncertainty bound. Again the FIM will serve as an approximation to the actual bound achieved by the specific estimator.

For this problem the FIM after the Nth maneuver is written

$$J_f[N] = FIM_0 + \begin{bmatrix} \sum_{k=1}^{N} \frac{\Delta y_k^2}{s_k^2 r_k^4} & -\sum_{k=1}^{N} \frac{\Delta x_k \Delta y_k}{s_k^2 r_k^4} \\ -\sum_{k=1}^{N} \frac{\Delta x_k \Delta y_k}{s_k^2 r_k^4} & \sum_{k=1}^{N} \frac{\Delta x_k^2}{s_k^2 r_k^4} \end{bmatrix} \tag{9}$$

Referring back to Figure 1,

$$\Delta x = x_{target} - x_{camera}$$
$$\Delta y = y_{target} - y_{camera}$$
$$r^2 = \Delta x^2 + \Delta y^2$$

Camera Motion Model

In order to define a search space for the trajectory generation, a camera motion model is necessary. For this paper, we assume a finite maneuver, fixed velocity model. Furthermore, we assume a fixed time interval T_s between each maneuver. The free variable then becomes the vehicle heading $\Theta_{heading}$ and the motion equation becomes

$$\begin{aligned} x_{vehicle}[k+1] &= x_{vehicle}[k] + VT_s \cos(\Theta_{heading}) \\ y_{vehicle}[k+1] &= y_{vehicle}[k] + VT_s \sin(\Theta_{heading}) \end{aligned}. \tag{10}$$

The parameters to be specified are the time interval T_s, the vehicle velocity V, and the total number of maneuvers N. There may also be a rotation between the vehicle and camera frame. For our system these frames are identical.

Constraints

The specifics of the localization problem add several additional constraints. For many relevant systems, such as a micro air vehicle (MAV) [19], the camera will be fixed in the vehicle frame and the vehicle translation and orientation will be dependent. In such a case, the camera cannot always be pointed at the target object. The path planner must allow for a maneuver that does not immediately improve the cost, but leads to more information later in time.

The error covariance matrix output by the EKF provides the opportunity for the motion plan to account for the uncertainty in the estimate and include a safety margin around the target location. Unfortunately the error covariance will not be known beforehand as it is a function of the specific estimator used. However, the inverse of the FIM relates to the covariance via the Cramer-Rao Lower Bound. Therefore an acceptable safety margin will be the ellipsoid around the estimate defined by the inverse of the FIM. The constraint can then be written as follows where the parameter n defines the number of standard deviations the ellipse represents:

$$\Delta X[k] * J_f[k] * \Delta X[k]^t > n \tag{11}$$

$$\Delta X[k] = \hat{X}_{target}[k] - X_{camera}[k]$$

A final set of constraints can be included to define camera limitations or further safety constraints on the vicinity of the target location. For the experimental set-up described below the following constraint equation describes the maximum range of the vision sensor as well as the closest approach the robot is allowed to make,

$$R_{\min} \leq \left| \hat{X}_{target}[k] - X_{camera}[k] \right| \leq R_{\max}. \tag{12}$$

Trajectory Generation Algorithm

Previous work has been done on the problem of optimizing the determinant of the FIM given a specified path length. Several different nonlinear methods have been used. Unfortunately these methods do not easily apply to our inverted formulation of the problem.

The vehicle heading for each maneuver is still a free variable, however, because the number of maneuvers is our cost function, the optimization state vector does not have constant length throughout the optimization process. Furthermore, the FIM and the error ellipsoid it defines now serve as constraints and must be determined at every position, as a function of the path taken previous to that position.

Instead of using a gradient based optimization method, we use exhaustive search. Our goal is a specified value for the determinant of the FIM. The search space is defined by discretizing the possible heading values that can be chosen at each node. By using a Breadth First Search, we are assured that the shortest possible path length will correspond to the first path that reaches the specified value. The maximum search depth can be specified to insure that the algorithm ends if a solution is not found in a suitable time.

The possible heading values must be discretized in order to define the problem search space. The size of the search space will increase with order $O(d^r)$ where d is the maximum search depth and r is the number of possible heading values at each node. The maximum depth is mainly a function of the problem geometry and the time step between each maneuver. The number of possible heading values describes the resolution of the search and is a given parameter.

If the size of the search space, and thus algorithm speed, were not an issue, the resolution would be set as fine as possible. However, the current algorithm running on a PC with eight maneuver solutions takes several minutes to complete. This is unacceptable for real-time applications. In response, a pyramid algorithm is used. The heading angle is coarsely discretized, on the order of 10-15 degrees. A solution is found. The allowable heading space is then rediscretized around the previous solution. The number of times this iteration is performed determines the final resolution and must be specified.

It is important to note that the search-based algorithm described here can perform the more common problem of minimizing the determinant of the FIM given a fixed path length. This solution amounts to performing an exhaustive search to a given depth and then finding the lowest value of the FIM at that level. Because information can only be added and not removed, this problem is monotonic and therefore guarantees that the minimum solution is at the maximum depth.

The pyramid search algorithm will be suboptimal compared to a solution using a single highly discretized heading space. Simulations have shown that for the case of a single known stationary target object and no constraints, the solution from the pyramid search algorithm is identical to solutions in the literature using gradient based optimization techniques to minimize the determinant for a given path

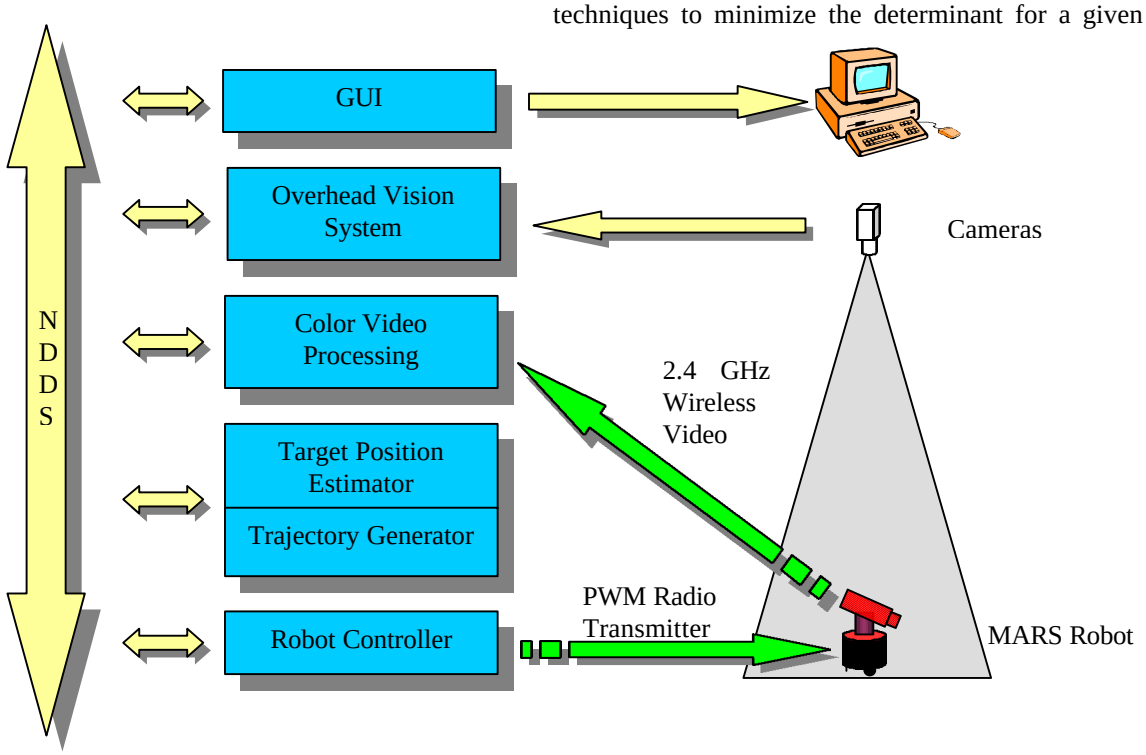

Figure 2. Experimental system diagram

length.

Unknown Target Location

The analysis above ignores an important detail about the Fisher Information Matrix for this problem. The point of the trajectory generation is to maximize the information available to an estimator in order to determine the target location. However, the above algorithm, through Equation 9, is a function of that very variable. Rather than knowing the location of the object, we have an estimate of its position and an error covariance of that estimate.

The actual algorithm used in practice uses the current target estimate in place of the actual value in Equation 9 and uses the inverse of the error covariance as the initial FIM value. Once a path has been generated, the vehicle takes off along that path. Knowing that the target estimate will change, the difference between the current estimate and the estimate used to generate the initial plan is monitored. When that value exceeds a predefined threshold, the path is replanned. This iterative procedure is continued until the specified error bound size is reached.

The iterative method described above is a variation of a batch sequential nonlinear experiment design [20]. It is different from typical sequential experiments in that the batch sizes (number of measurements between replans) are not fixed. Instead, they are functions of the behavior of the estimator. Thus the initial conditions of the estimator directly influence the behavior of the trajectory generation. The method presented here is advantageous over the batch and fully sequential designs because it only replans when the estimate has changed significantly. When the initial estimate is close to the actual value, the initial trajectory is close to the optimal path, and the system will not waste time replanning.

4. EXPERIMENTAL SYSTEM

The algorithms described above have been demonstrated experimentally on the Micro Autonomous RoverS (MARS) platform of the Aerospace Robotics Lab (ARL) [17]. A brief description of the MARS robots is given below. In addition, the remaining components needed to perform the target localization and trajectory generation are described. These include the robot controller, overhead vision positioning sensor, color vision processor, target location estimator and trajectory generator, and graphical user interface. Each program is run on a different computer and is connected by an Network Data Delivery System (NDDS) virtual data bus based on the publish-subscribe network delivery protocol. The system is shown in Figure 2.

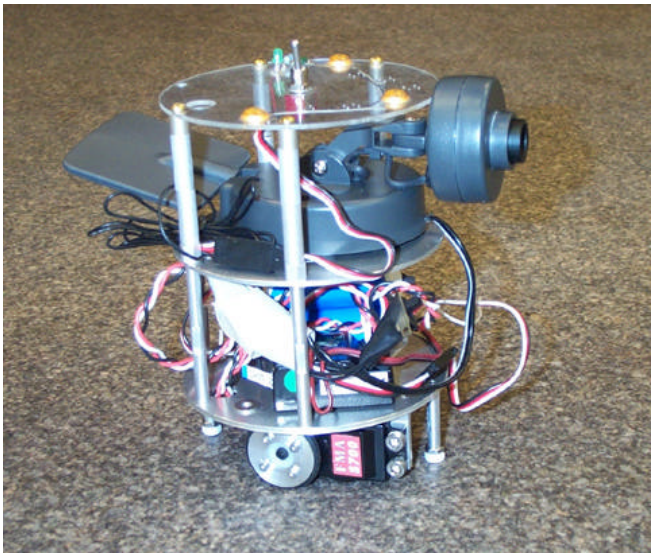

Figure 3. Modified MARS Rover

Micro Autonomous Rover

The vehicle used for the experiments presented in this paper is a modified version of the Micro Autonomous Rover (MARS) developed to study multiple robot planning problems (see Figure 3). The robot is a small, 6 inch diameter, wheeled robot that is controlled through two radio-controlled servomotors. All motion sensing and control is carried out off-board the vehicle. An overhead vision sensor provides the position of the rover on a 2 x 3 meter table. The controller sits on a second computer and sends actuator commands to the robot via a PWM radio signal.

The robot location and orientation is sensed by the overhead vision system which tracks a unique pattern of three infra-red LEDs that sit atop the rover. Three black and white cameras with IR filters span the workspace of the robot. Both position and velocity information is available for the robot and for the target object.

A small wireless camera is mounted on the top of the MARS rover at an angle looking downward towards the ground. The unit is commercially available and is integrated with a 2.4 GHz wireless video transmitter.

Color Video Processing

Orange colored cones with LEDs mounted on top are used as target objects to facilitate the target tracking aspect of this problem. This research focuses on planning appropriate paths in order to achieve localization and will leave the details of target tracking to the computer vision specialists. However, in order to realistically incorporate the error sources and constraints imposed by an actual target tracking subsystem, the simple vision system was used.

Table 1. Trajectory Generation Parameters - multiple values indicate the parameter was changed for different runs

Maximum Number of Maneuvers	8
Number of iterations	6
Number of angles	5
Speed (meters/sec)	0.075
Sigma (radians)	0.0087
Time Between Maneuvers (seconds)	2.0
Minimum Range (meters)	0.03
Maximum Range (meters)	6.0
Camera Offset (radians)	0.0
Camera Half Angle (radians)	0.35
Replan Tolerance (meters)	0.20, ∞
Desired Area (meters squared)	0.0, 0.01

A dedicated PC running the Redhat operating system uses a Matrox Meteor 1 board to digitize the color video images at 30 Hz. Objects are defined by an elliptic region in the RGB color space. Tracking may be initialized either by a human operator clicking on a colored object or by the system automatically tracking all colored sections of the image. Camera calibration was performed in order to determine the intrinsic and extrinsic parameters of the setup.

Target Location Estimator

The localization estimator and motion planner both reside on the same computer. They are run as separate programs so that they could be put on different machines if necessary.

The localization estimator runs at 30 Hz on a real-time operating system. Inputs to the program are the position and orientation information of the robot and the calibrated target track in the image plane. The estimator assumes there is a single stationary object and that all incoming vision measurements belong to that object.

The estimator is initialized by using the first 5 target measurements obtained and performing a simple batch triangulation. The initial target location is set to the result of this triangulation and the initial covariance is defined as a circle with radius equal to the distance from the vehicle after the last measurement to the initial target estimate. This covariance is conservative because it does not use the tracking noise model to flatten the initial covariance. However, the EKF quickly collapses the covariance to the appropriate shape.

Trajectory Generator

The trajectory generator receives the target location from the estimator and the robot data from the overhead vision system. Again, it assumes a single, stationary target. The current version of the trajectory generation software takes several minutes to compute a path with 8 or more maneuvers. While the computation occurs, the robot stops its motion and waits. The trajectory generator uploads the new path to the robot controller and signals it to continue moving.

The Experiment

The results presented in the next section were generated from test runs of the system. The vehicle and target started approximately 2 meters apart. The target was brought into the field of view of the robot's cameras by a human operator. Likewise, the vision tracking was initialized from the graphical user interface. Finally, the operator commanded the robot to begin its motion by moving towards the object. This motion is necessary to give the estimator sufficient information to initialize. All the user-specified variables are listed in Table 1. The results of these experiments are discussed in the next section.

5. RESULTS

The results of several different experiments using various parameter values are plotted in Figure 4 through Figure 13. In each experiment the robot starts at the origin. The common behavior across all the runs is that the robot moves toward the target before moving to the side. Decreasing the range first enables an increased change in bearing later, yielding more information to the triangulation subsystem.

The first run, shown in Figure 4, demonstrates an eight maneuver path using the known location of the target object. In practice, this is an unreasonable situation as the estimator will only have an initial guess of the target location. However, it represents the best possible path given the geometry of the problem.

Notice, as with all the subsequent paths, the robot heads toward the object first, turning and moving to the side at the end of the path. The initial maneuvers have a high range rate and add little information to the system. However, the low information gain is traded off for high bearing rates, and even higher information gain, in the second half of the path. The bearing rate is the key component of the optimal trajectory. If the optimal path was a function of the absolute change in bearing angle, the expected path would be a straight line.

The behavior of the robot in the ideal run clearly demonstrates that the optimal path cannot be calculated without knowledge of the target location and without predicting into the future. The best move to do at a given instant is to move at an angle to the line of sight of the target. However, the best path for a given total time is not the sum of the best moves at every instant.

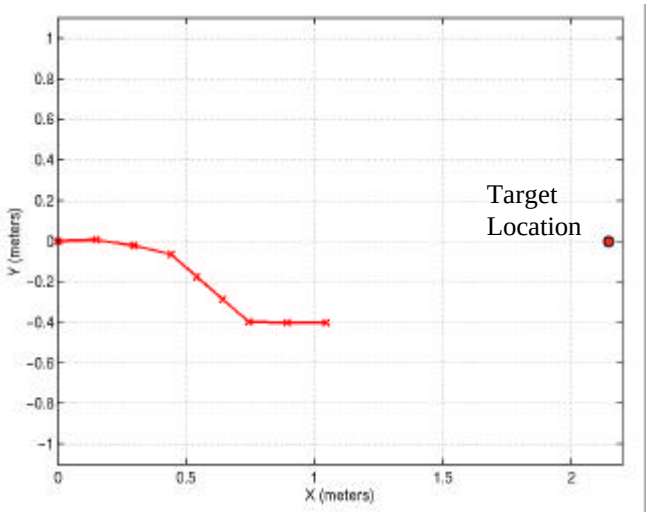

Figure 4. Eight maneuver path using known target location

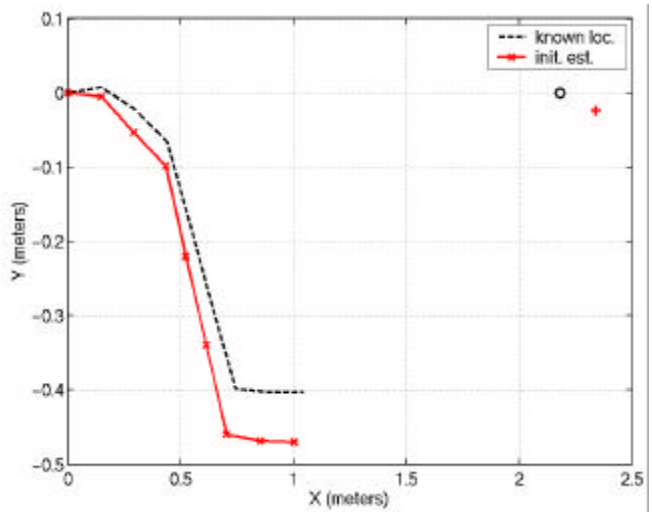

Figure 5. Eight maneuver path using initial target estimate with no replanning (note axes scales not equal)

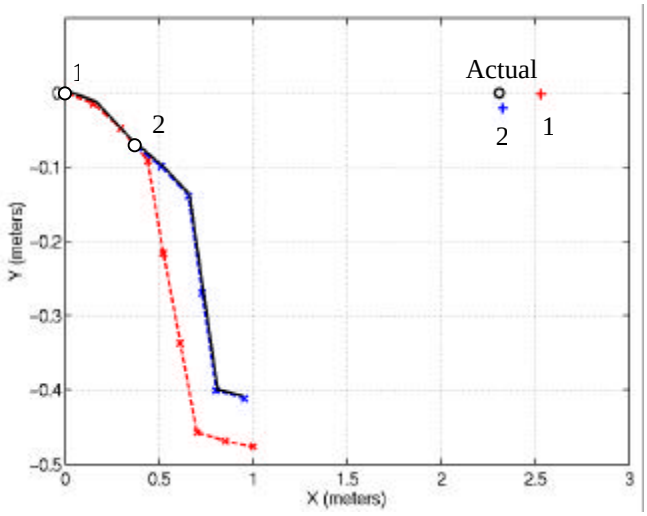

Figure 6. Eight maneuver path replanning when target location changes by 20 centimeters.

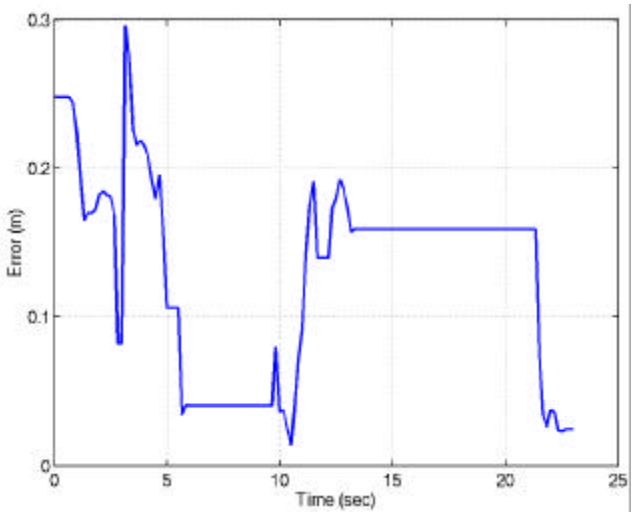

Figure 7. Magnitude of target location error vs. time

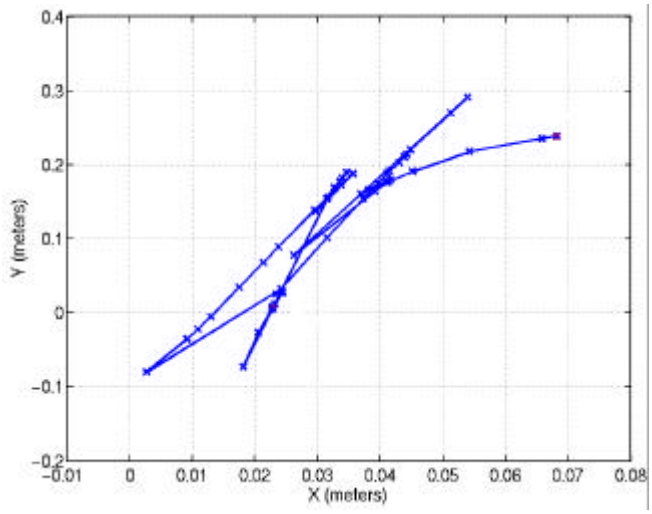

Figure 8. Target location error

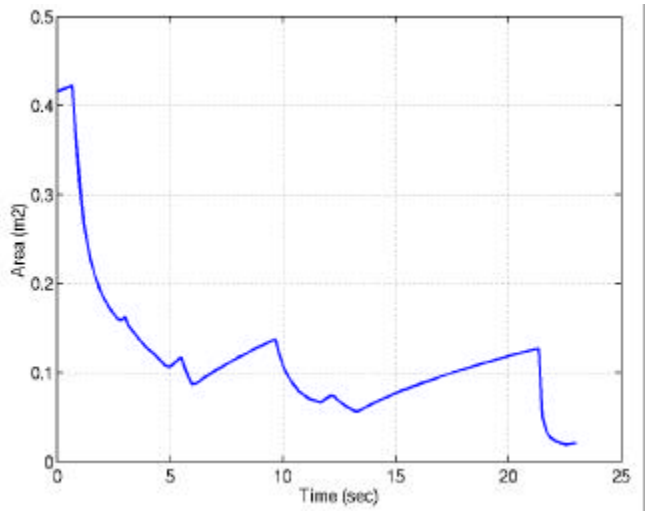

Figure 9. Area 0f 1-sigma target error ellipse

The next run, shown in Figure 5, is also an eight maneuver path. As opposed to the first run, the trajectory generator uses the initial estimate of the target location (marked with the red cross) in order to plan the path. Compared to the first experiment, the trajectory turns sooner and therefore terminates farther below the first path.

During the next experiment, the planner replans whenever the estimate changes by more than 20 centimeters. In the example shown in Figure 6, the planner has to replan twice. The numbered circles mark where the vehicle replanned and the crosses mark the estimate location used. The dashed colored lines indicate the calculated paths while the solid black line marks the actual path followed by the vehicle. Because the total path length is fixed, each subsequent replan uses fewer maneuvers, reducing the depth of the search space and decreasing the computation time.

Figure 7 through Figure 9 show the behavior of the target estimate during the experiment of Figure 6. As expected, the estimate error is significantly reduced as the vehicle moves. Errors in the vision tracking due to noise, calibration errors, and a short time lag, cause the estimate to jump about and not converge smoothly. This behavior is best demonstrated in Figure 7.

In contrast to the estimate, the $1-s$ error ellipse area, as defined by the estimate covariance matrix, decreases in a steady manner. Figure 9 shows the error area after the propagation step of the Kalman filter. The few upward trends occur when no measurements are being taken. This situation takes place in one of two ways. The first case transpires when the vehicle stops to replan. In order to assure proper filter behavior the EKF only innovates when the velocity is non-zero. The second case corresponds to instances when the vehicle's camera is turned away from the object. The large rise at time t = 6 seconds is an instance of the former case while the larger one at t = 13 seconds is the latter.

The final experiment, shown in Figure 10 through Figure 13, demonstrates the minimum length path needed to achieve a desired error ellipse area of 0.01 meters squared. Because the planner is ideal and the actual estimator is not, the system must iterate and continue moving several times before the desired area is reached. The numbered circles mark the locations where new plans were computed.

The reasons for the multiple additional plans are twofold. First, the estimator is not efficient and therefore does not reach the ideal. This is to be expected because the estimator is a linearization of the nonlinear problem. Second, and more significant, the estimator model includes non-zero process noise in order to improve the convergence properties.

Figure 13 shows the area of the $1-s$ error ellipse for this run. There are four distinct regions where the area grows. The first large section occurs when the vehicle turns away from the target. The other three occur while the vehicle is stopped. The plot shows that the reduction in the error bound during the short additional maneuver is almost counterbalanced by the increase that occurs while replanning.

The results of the final experiment demonstrate the inappropriateness of the Fisher Information Matrix as an optimization metric when a specified error bound is desired or when the vehicle is expected to turn away from the target. In both cases a better metric will be the predicted error covariance which accounts for the effects of process noise in the system. Although the stationary model should have no process noise, it is included in order to give the EKF better performance. Furthermore, the algorithm will need to incorporate process noise if more complex, dynamic

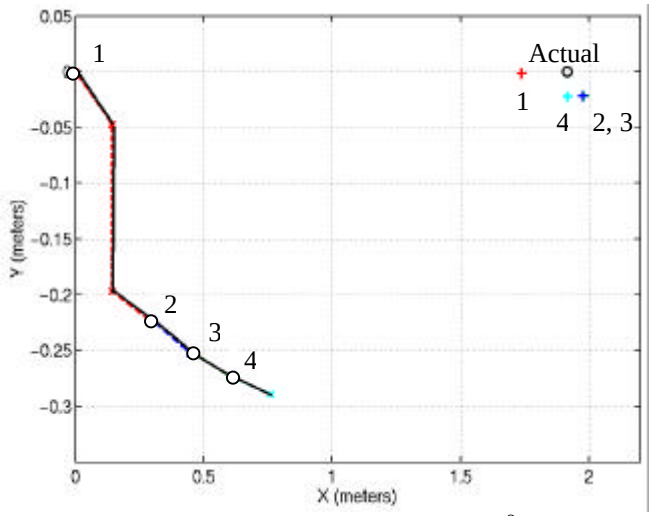

Figure 10. Desired error ellipse area of 0.01 m^2, replanning if initial path does not yield desired result

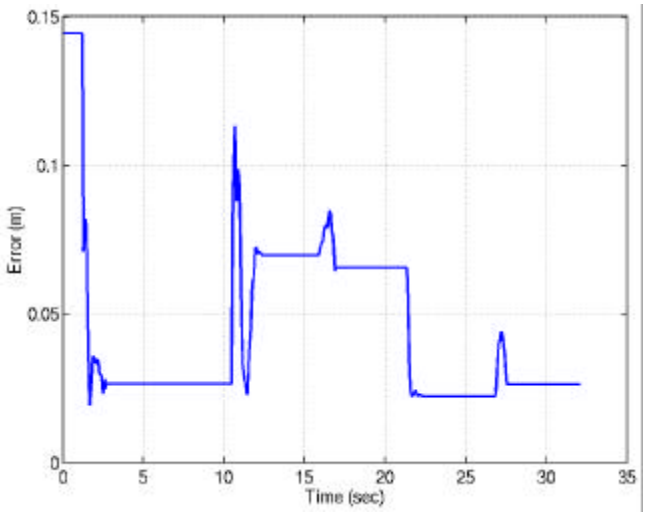

Figure 11. Magnitude of target location error vs. time

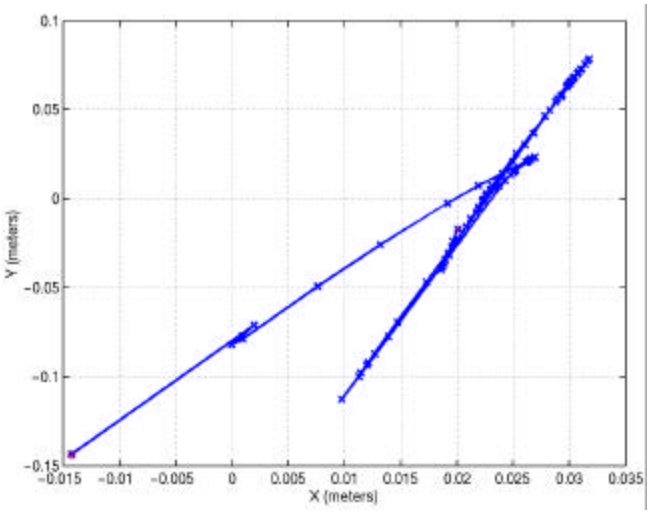

Figure 12. Target location error

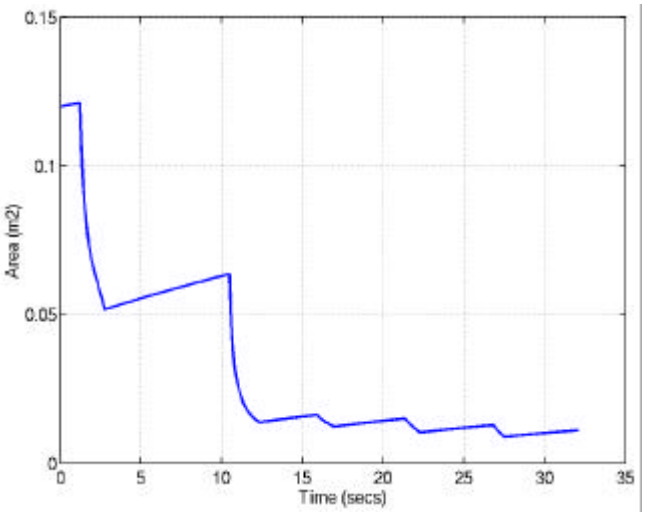

Figure 13. Area 0f 1-sigma target error ellipse

motion models are to be considered in the future.

6. CONCLUSIONS

The nonlinear nature of monocular vision-based target localization requires the generation of trajectories that insure proper observability. Furthermore, these trajectories can be generated to maximize the information, via the Fisher Information Matrix, provided to the estimation subsystem.

The total system, estimator and path planner, contain a paradox. The path planner needs to know where the target is located in order to plan the optimal path. However, the point of planning the path is to provide information to the estimator in order to determine the location of the target. Any trajectory generation solution must account for this fact. The algorithm presented here uses a simple iterative approach.

The generated paths begin with a relatively large range rate in exchange for an increased bearing rate, and more information, towards the end of the path. In all of the cases shown here, the general pattern of behavior was the same.

Current work is being done to characterize the impact of the different design parameters, such as the replan distance, and to investigate new formulations of the optimal planning problem that explicitly incorporate the initial estimate uncertainty bounds.

Finally, this paper showed that the algorithms described above can be used successfully on a real robot, using a simple vision tracking system.

7. REFERENCES

[1] D. Murray and J. Little, "Using real-time stereo vision for mobile robot navigation", *Proceedings of the IEEE Workshop on Perception for Mobile Agents, Santa Barbara, CA, 161-171,* June 1998.

[2] T. Kanade, R. T. Collins, A. J. Lipton, P. Burt, and L. Wixson, "Advances in Cooperative Multi-Sensor Video Surveillance," *DARPA Image Understanding Workshop (IUW), Monterey, CA, 115-122,* Nov. 1998.

[3] M. W. M Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, "A Solution to the Simultaneous Localization and Map Building (SLAM) Problem", *IEEE Transactions on Robotics and Automation* ***17*** *No. 3, 229-241,* June 2001.

[4] J. Leonard, "Large-Scale Concurrent Mapping and Localization", *Proceedings of SPIE* ***4196****, 370-376,* 2000.

[5] M. Deans, "Robust and Efficient Simultaneous Localization and Mapping from Bearings Only Sensing" Thesis Proposal, Carnegie Mellon University, 1999.

[6] C. Tomasi and T. Kanade, "Shape and Motion from Image Streams: a Factorization Method.", Technical Report CMU-CS-92-104, Carnegie Mellon University, 1992.

[7] C. J. Taylor, D. J. Kriegman, and P. Anandan, "Structure from Motion in Two Dimensions from Multiple Images: a Least Squares Approach", *Proceedings of the IEEE Workshop on Visual Motion, 242-248,* 1992.

[8] V. J. Aidala, "Kalman Filter Behavior in Bearings-Only Tracking Applications", *IEEE Transactions on Aerospace and Electronic Systems* ***AES 15****, No. 1, 29-39,* Jan. 1979.

[9] A. Logothetis, A. Isaksson, and R. J. Evans, "Comparison of Suboptimal Strategies for Optimal Own-Ship Maneuvers in Bearings-Only Tracking", *Proceedings of the American Control Conference, 3334-3338,* June 1998.

[10] S. Nardone, A. Lindgren, and K. F. Gong, "Fundamental Properties and Performance of Conventional Bearings-Only Target Motion Analysis", *IEEE Transactions on Automatic Control* ***AC 29****, No. 9, 775-787,* Sept. 1984.

[11] O. Guanghui, S. Jixiang, L. Hong, and W. Wenhui, "Estimating 3D Motion and Position of a Point Target", *Proceedings of SPIE* ***3173****, 386-394,* 1997.

[12] S. E. Hammel, P. T. Liu, E. J. Hilliard, and K. F. Gong, "Optimal Observer Motion for Localization with Bearing Measurements", *Computers Math. Applic.* ***18****, No. 1-3, 171-199,* 1989.

[13] J. M. Passerieux and D. Van Cappel, "Optimal Observer Maneuver for Bearings-Only Tracking", *IEEE Transactions on Aerospace and Electronic Systems* ***34****, No. 3, 777-788,* July 1998.

[14] Y. Oshman and P. Davidson, "Optimization of Observer Trajectories for Bearings-Only Target Localization", *IEEE Transactions on Aerospace and Electronic Systems* ***35****, No. 3, 892-902,* July 1999.

[15] J. C. Spall, "The Information Matrix in Control: Computations and Some Applications", *Proceedings of the 38th Conference on Decision and Control, 2367-2372,* Dec. 1999.

[16] P. Chaudhuri and P. A. Mykland, "On Efficient Designing of Nonlinear Experiments", *Statistica Sinica* ***5****, 421-440,* 1995.

[17] C M. Clark and S M. Rock, "Randomized Motion Planning for Groups of Nonholonomic Robots", *Proceedings of the 6th International Symposium on Artificial Intelligence, Robotics, and Automation in Space,* Montreal Canada, June 2001.

[18] R. F. Stengel, *Optimal Control and Estimation,* New York: Dover Publications, Inc, 1994.

[19] S. Morris and M. Holden, "Design of Micro Air Vehicles and Flight Test Validation", http://spyplanes.com/Background/index.html.

[20] I. Ford, D.M. Titterington, and C. Kritsos, "Recent Advances in Nonlinear Experiment Design", *Technometrics* ***31,*** *No. 1, 49-60,* Feb. 1989.

8. BIOGRAPHY

Eric Frew *is a Ph.D. candidate at the Stanford University Aerospace Robotics Laboratory. He received his M.S. from the Aeronautics and Astronautics Department at Stanford University in 1996 and his B.S. from the Mechanical Engineering Department of Cornell University in 1995. His research interests include computer vision and autonomous vehicle control, with an emphasis on aerial and space robotics. He was the project manager of the Stanford HUMMINGBIRD Autonomous Helicopter.*

Dr. Stephen Rock *received his S.B. and S.M. degrees in Mechanical Engineering from MIT in 1972. He received his Ph.D. in Applied Mechanics from Stanford University in 1978, and joined the faculty of the Aeronautics and Astronautics department of Stanford in 1988. He is the Director of the Aerospace Robotics Laboratory where his research focus is to extend the state-of-the-art in robotic vehicle control. His interests include the application of advanced control techniques for robotics and vehicle systems. Areas of emphasis include remotely operated vehicles for both space and underwater applications. Dr. Rock teaches several courses in dynamics and control at Stanford. Prior to joining the Stanford faculty, Dr. Rock managed the Controls and Instrumentation Department of Systems Control Technology, Inc.*