2006

Deployment strategies for hybrid free-space optic/ radio frequency mobile networks

Christopher Ryan Mansley
Lehigh University

Mansley, Christopher Ryan

Deployment Strategies for Hybrid Free-Space Optic/Radio...

May 2006

DEPLOYMENT STRATEGIES FOR HYBRID FREE-SPACE OPTIC/RADIO FREQUENCY MOBILE NETWORKS

by
Christopher Ryan Mansley

A Thesis
Presented to the Graduate Committee
of Lehigh University
in Candidacy for the Degree of
Master of Science
in
Computer Science and Engineering

Lehigh University
2006

This thesis is accepted in partial
fulfillment of the requirements for the degree of
Master of Science.

28 Apr 06
(Date)

John Spletzer

Hank Korth

Acknowledgment

First, I would like to thank my advisor, John Spletzer, whose patience and experience I could not have done without. I would also like to thank Jason Derenick, who made me appreciate the simplicity and beauty of mathematics and would always answer my questions, no matter how quaint. If it were not for Chris Thorne, I would have never been involved in this project at all. In general, I would like to thank anyone who has worked in the VADER Laboratory (Humberto Sermeo-Villalta, Mike Kowalski, Chao Gao, Matt Casey, etc...) for providing a fun and simulating work environment.

Lastly, I would like to thank my family and friends for being so supportive of my long hours, late nights and general absenteeism. They have always been there for me, always.

Contents

List of Figures

Abstract

In this paper, we present an extension to existing work on hybrid free-space optics/radio frequency (FSO/RF) networks for mobile robot teams. Previous work [3], raised technical challenges for the deployment of mobile robot teams utilizing hybrid FSO/RF networks. This paper attempts to address these concerns by demonstrating a feasible implementation for autonomous deployment of a free-space optical link in an outdoor environment.

Specifically, we address the problem of navigation and link acquisition in an outdoor environment. The solution uses an extended Kalman filter (EKF) to fuse local sensor measurements, and a proportional-differential controller for navigation. Our link acquisition is a vision-based system, which we defined so as to establish high-throughput carrier grade patches. Unlike a naive approach, our method avoids correlation-based template matching and instead implements a scale invariant feature transform (SIFT)[9]. This method allows us to establish optical links using only a single template that can be tracked independent of affine transforms in the image scene.

Chapter 1

Introduction

In a scenario where communications infrastructure in an urban area is damaged or destroyed, often the repair work is hazardous or impractical and would benefit from an automated, temporary repair technique. Using traditional radio frequency (RF) technologies to patch a severed fiber link is not practical given the rapid decline in throughput at moderate ranges. Instead, we envision a scenario where a team of automated robots could be used to create carrier-grade patches to the damaged network. By using free-space optic transceivers (FSO), the distance and bandwidth limitations are resolved, but deployment becomes slightly more complex. Our work combines RF and FSO network technologies to create a hybrid network that can be deployed automatically using robots and a simple link acquisition system[2].

Free-space optics are not a direct replacement for all RF network solutions, but they do provide advantages that, in combination with RF technologies, make a hybrid solution desirable. RF networking typically does not require line-of-sight for deployment, where as FSO technologies do. This shortcoming does come with the advantage of high throughput. While most all widespread wireless technologies (*e.g.* UWB, 802.11x), rapidly lose bandwidth at short distances ($r \geq 15m$) [13], commercial FSO technologies can provide several Gbps over distances greater than a kilometer. Hybrid solutions could provide the best of both worlds, giving long distance, high throughput backbones to small clusters of RF networks that do not have line-of-sight restrictions.

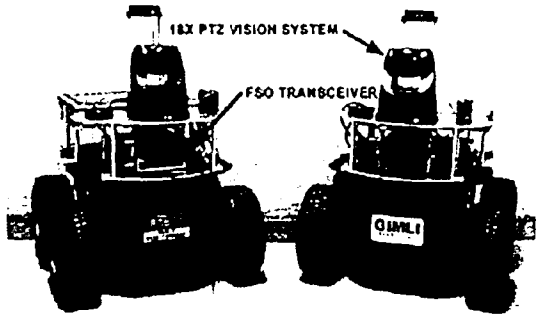
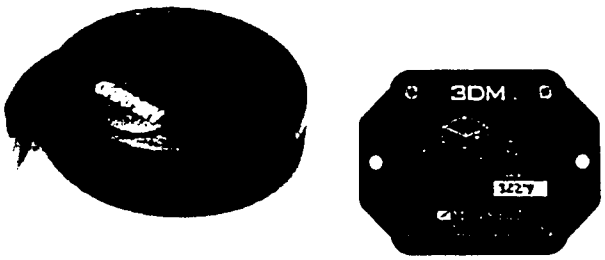


Figure 1.1:

The approach taken here is combine a RF/FSO hybrid solution with a small group of autonomous robots (*e.g.* see Figure 1.1). In the past, the setup of a FSO link required tedious alignment of both networking heads to provide line-of-sight communication. This involves a skilled human technician, possibly one at each site setting up the FSO link, often only in a static configuration. Our approach uses a hierarchical link acquisition system that utilizes high zoom cameras, pan/tilt mechanisms and prior knowledge to locate, identify and connect to link partners. All of our experiments were done in an autonomous fashion using two robotic research platforms in an outdoor environment.

1.1 System Overview

For all of the trials conducted, we used a pair of modified Pioneer P3-ATs. These robots are equipped with four major systems that were used for the navigation and link acquisition: vision system, pan/tilt FSO head, GPS receiver and three axis orientation sensor. The base platform is provided by MobileRobots (formally ActivMedia) and is equiped with odometry information, which we use to measure the linear velocity of the robot. Mounted to the platform is a pan-tilt-zoom camera from Sony using a 1/4 inch CCD, with a optical focal length from 4.1 to 73.8 mm. The pan tilt FSO mechanism is a custom built assembly used to orient the FSO,



(a) Garmin GPS18 (b) MicroStrain 3DMG

Figure 1.2: These sensors are used to measure position and orientation of the robot platform

a 100 Mbps *LaserWire*, with two degrees-of-freedom (DOF). The GPS receiver is a Garmin GPS 18 USB, which is able to track 12 satellites and is enabled with Wide Area Augmentation System (WAAS). Finally, mounted on an extention above the robot, is the three axis orientation sensor or inclinometer. Our robots are equipped with a MicroStrain 3DMG capable of measuring roll, pitch and yaw in absolute, velocity and acceleration values.

Chapter 2

Link Acquisition

2.1 Introduction

In any of the proposed scenarios involving hybrid RF/FSO nodes, the largest technical challenge involves establishing and possibly maintaining the optical link. In a commercial scenario, a skilled technician would use an optical sight to carefully align the two FSO heads. This process must now be replicated by the hybrid node, so as to effectively and efficiently establish the connection. To overcome these issues, a hierarchical link acquisition system was proposed in [3]. A review of that system and our improvements to it are outlined here. An overview of the entire link acquisition system can be seen in Figure 2.1.

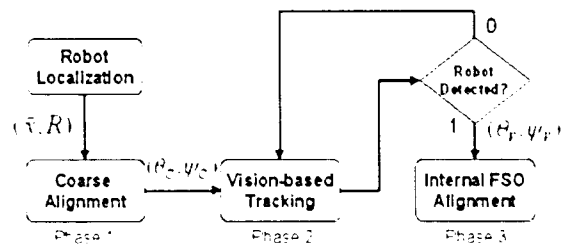


Figure 2.1: The three phase diagram of the link acquisition system

2.2 Coarse Alignment

In the proposed scenario, we can imagine that both robots are given *a priori* designated waypoint targets. This implies that prior to deployment each robot would know its link partner's objective pose. We assume there is a common reference frame, W, to which both robots can compute their own pose (position and orientation). This assumption is not restrictive, as we implemented a solution using a global positioning system (GPS) and an orientation sensor, which provided a common frame of reference for all of the robots.

We will typically denote a position or objective waypoint as, $x \in \Re^3$, in our world frame, W. We will assume there are two objective positions, $\{x_1, x_2\}$, in the world frame. Each robot will be able to navigate to the objective positions with some small pose error, which assumes that the robots can measure their position and orientation in the world frame. After each robot reaches the objective waypoint, it will adjust its relative orientation to coarsely align the FSO transceivers. Without loss of generality, only one side, x_1, will be shown in the calculations, but they are equally valid from each side of the link. Assuming that the optical transceiver frame, F_1, and W are equivalent up to a translation, then the link direction from x_1 to x_2 would be

$$\vec{x} = \frac{x_2 - x_1}{\|x_2 - x_1\|} \tag{2.1}$$

This does not completely define the direction from F_1 to x_2, because of the uncertainty in current orientation. Using the local rotation matrix, $R_1 \in SO(3)$, as measured by the on-board sensors, we can compute the target direction in F_1 by $\dot{x} = R_1^T \vec{x}$. Converting this target direction to spherical coordinates

$$\begin{bmatrix} \theta \\ \upsilon \end{bmatrix} = \begin{bmatrix} \arctan\left[\frac{\dot{x}(3)}{\sqrt{\dot{x}(1)^2 + \dot{x}(2)^2}}\right] \\ \arctan\left[\frac{\dot{x}(2)}{\dot{x}(1)}\right] \end{bmatrix} \tag{2.2}$$

where $\dot{x}(k)$ denotes the k^{th} element of the vector. The result, $\{\theta, \upsilon\}$, defines the azimuth and elevation for the FSO head to be aligned with the link partner's location.

As previously stated, each robot may have both position and orientation errors, which makes the direct application of (2.2) difficult. However, we can use these

results to seed a second alignment stage that will refine the previous orientation estimates given above. The use of the optical link implies a line-of-sight to the partner robot, which can be exploited in refining our link estimate. This allows the robots to utilize a high zoom camera to assist in the second phase of the link establishment.

2.3 Vision

After migrating to their respective objective waypoints, the robots have orientate themselves, so as to be approximately the optical transceivers in the right place facing each other. In reality, this migration process inherently has an error associated with it, making it unlikely that the partner robot will be in the theoretically ideal location. As the distance between the robots grows, so does the possibility that the partner robot will be outside the initial field-of-view (FOV). To combat this problem, in the second phase, the robot will search in azimuth and elevation with its pan-tilt-zoom camera for the partner robot. Using this information, the orientation estimate can be recalculated to provide the correct angle information for aligning the FSO head.

One of the main advantages of using a vision-based approach is that each robot can actively search for its link partner independently of the link partner's own search. Since each robot has a FSO transceiver, we could search using the internal link acquisition system built into FSO heads to test alignment. But, this approach would expand the search space exponentially, because both heads would need to be aligned to verify a positive match. In our vision approach, each robot can locate the link partner independently of any other robot, thereby reducing the search space from $\Re^4$ to $\Re^2$.

2.3.1 Previous Work

In the initial proposal [3], a two-dimensional pattern matching approach was used. The authors discuss the limitations associated with using this type of pattern matching scheme. The first limitation is the number of templates needed to capture the variety of orientations and positions, which would become computationally intractable very quickly. The authors reduce the number of templates needed by exploiting the *a priori* knowledge of the location and orientation of the link partner. The paper goes on to suggest that this would reduce the number of templates (≈ 20) as long as a robust tracking algorithm is used. This type of template matching approach also quickly breaks down under large ($> 30\%$) changes in scale or orientation.

The approach taken here is to replace the template based pattern matching with another algorithm, which would perform more robustly in an outdoor environment. Specifically, a scale invariant feature transform (SIFT) approach addresses these concerns directly by being resistant to scale, affine distortion, changes in 3D viewpoint and illumination. In addition, an object typically only requires one template or set of keys to be detected robustly.

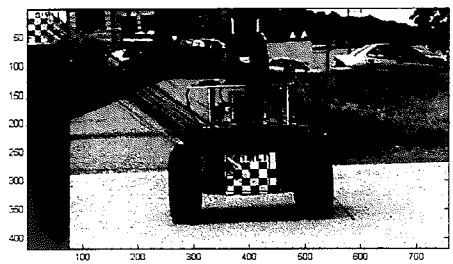
2.3.2 SIFT Approach

Scale invariant feature transforms (SIFT) were first outlined in Lowe's paper [8] and then later expanded and improved on by Lowe and others [9][6]. The basic idea is to generate features that are invariant to affine transforms and illumination changes. These features (or keypoints) are extracted from a reference image of the object being tracked and stored in a database. An object is detected by generating the SIFT features of the scene image and finding a match in the database corresponding to a reference object. Each object can be thought of as having a feature vector associated with it and then trying to generalize the new vectors from the scene to find the closest match in feature space. Many of the same approaches for speeding up feature matching, that apply in pattern recognition, apply in this context as well.[4]

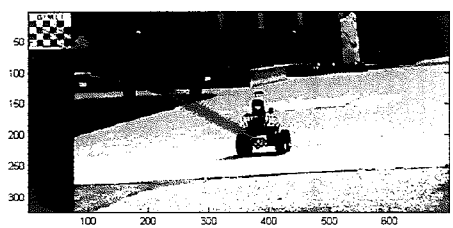
As no modifications were made to the original algorithm defined by Lowe [9], only

a brief overview of the technique will be given here. The algorithm has four major stages of computation to generate the keypoints. The first is the scale-space extrema detection. In this stage, the difference-of-Gaussians is computed to be used to detect areas of interest in the image from which to select keypoints. In this scale-space the local maxima and minima are located by comparing the point of interest with its 8 neighbors and 18 neighbors in the scale above and below the current scale. The second stage is the keypoint localization. Initial work on this stage simply placed the keypoint at the location and scale of the central sample point, but newer methods use a data fitting of the data in the surrounding area to give better results overall. Part of this stage is a filtering step that removes any keypoints that are not strong or stable enough for keypoint selection. The third stage orients the keypoints relative to the local image information. By using the local image information, a keypoint can be assigned an orientation that is rotationally independent of the scene. The fourth stage generates the keypoint descriptor, which is a representation of the location and orientation of the keypoint that removes the remaining variants, namely illumination and 3D viewpoint. This final step is based on some research done on mammalian vision systems, generating a key that is relatively invariant to affine transforms and illumination.

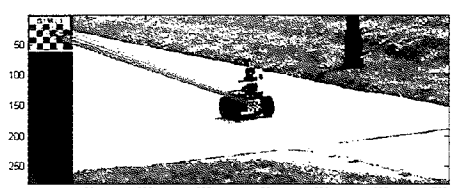
The interesting results from the paper are in terms of number of keys and the stability of detection under heavy rotation. As each object that is being tracked can generate hundreds of keys, the uniqueness of the keys could be a concern. But, the experiments in Lowe's paper demonstrate an 80% repeatability in a database of over 100000 keys. Under rotations greater than 30 degrees, the algorithm can select the correct keys $> 75\%$. Although, this may sound like a low percentage of correct matches, each template can contain hundreds of keypoints, making the probability high that a large number of keypoints will be matched against the database. In our own experiments, shown in Figure 2.2, the algorithm performed better than expected under large changes in both scale and rotation.



(a) Minor Transform



(b) Major Scale, Minor Rotation



(c) Major Scale, Major Rotation

Figure 2.2: Examples of SIFT features being generated and matched in a variety of scenarios

Chapter 3

Navigation

3.1 Introduction

In previous work, the focus was on generating a proof-of-concept system that worked in a controlled indoor environment [3]. In this type of environment, simple navigation techniques can be used effectively [11]. However, extending some of these techniques to an outdoor environment can be imposing or unreliable. The lack of any landmarks, precludes the possibility of using an approach that would utilize reference positions to compute its pose. Instead, different approaches to navigation must be used to ensure the reliability and accuracy needed for the link acquisition system outdoors. In the following sections, a Kalman filter based localization system is described, as well as a proportional-derivative (PD) controller, which replaces a simple reactive controller.

3.2 Kalman Filter

One of the most popular robot localization tools is the Kalman filter. This algorithm allows for a computationally efficient way of estimating the current probability density, while at the same time minimizing the mean squared error. This filter allows us to estimate the past, present and future of the state, even with a poorly defined

model of the system. A brief review of the filter along with our derivations follows. [12]

The filter addresses the general problem of estimating the state of a process given a set of sensor measurements, their respective probability distributions and a dynamic (or motion) model. The basic Kalman filter requires the state and measurement equations be governed by linear stochastic difference equations. This means that the state, $x \in \Re^n$, can be represented by a linear equation of the form

$$x_k = Ax_{k-1} + Bu_{k-1} + w_{k-1} \tag{3.1}$$

with a measurement, $z \in \Re^m$, of the form

$$z_k = Hx_k + v_k \tag{3.2}$$

In this formulation, the random variables w_{k-1} and v_k represent the motion model (or process) noise and measurement noise respectively. The variables are represented by zero mean Gaussians and are considered independent of each other. The normal probability distributions can be represent as follows

$$p(w) \sim N(0, Q) \tag{3.3}$$

$$p(v) \sim N(0, R) \tag{3.4}$$

where Q and R are the process noise and measurement noise covariance matrices, respectively.

The $n \times n$ matrix A in (3.1) describes the relationship between the state at time $k-1$ and the current time k, where as the $n \times 1$ matrix B relates the optional control input to the current state. The matrix H relates the state to the current measurement. All of the matrices could be dynamic from one time step to the next and, in fact, in our final implementation, our noise covariances changed at each measurement. For this linear process, the Kalman filter will estimate the state to make the error covariance of the state estimate as minimal as theoretically possible. However, the state and measurement are defined in terms of linear equations, thus limiting the power of the general Kalman filter to linear process relationships.

Often in navigation or localization problems, the process model is not linear with respect to the robot or sensors. This violates the constraints required by the regular Kalman filter, making it useless for this application. Instead, a Kalman filter that linearizes about the current mean and covariance can be used, which is referred to as an extended Kalman filter (EKF). The original process definitions given above have to be modified to allow for a non-linear equation to define both the state and measurement data as follows

$$x_k = f(x_{k-1}, u_{k-1}, w_{k-1}) \tag{3.5}$$

$$z_k = h(x_k, v_k) \tag{3.6}$$

as in (3.1) and (3.2), the random variables w_{k-1} and v_k represent the noise distributions of the process and measurement noise. Often in practice, these noise values are unknown at each time step and therefore set to zero. Instead of the linear equations relating the state, measurements and optional control, it is now the non-linear functions f and h that relate the previous time step to the current one.

Both the regular Kalman filter and the extended Kalman filter estimate the process by using a form of feedback control. This feedback can also be described in the two steps of the algorithm, time update and measurement update. The time update equations are responsible for projecting the state forward in time based solely on the dynamic model and control inputs, A and Bu_{k-1} respectively from (3.1). During the measurement update phase, the state estimate is corrected by the measurements taken reflecting (3.2). This two cycle approach is often called a predictor-corrector algorithm, with the time update predicting the next state estimate and then allowing the measurement update to correct the estimate. The derivation of the EKF equations results in the final result of a set of time update equations

$$\hat{x}_k^- = f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{3.7}$$

$$P_k^- = A_k P_{k-1} A_k^T + W_k Q_{k-1} W_k^T \tag{3.8}$$

For a complete definition of notational reference, see Welch and Bishop [12]. During the time update phase, the state, $\hat{x}_k^-$, and the covariance, P_k^-, estimates are

projected forward based on the current (or past) state and the control input. The set of measurement update equations are derived as

$$K_k = P_k^- H_k^T (H_k P_k^- H_k^T + V_k R_k V_k^T)^{-1} \tag{3.9}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^-, 0)) \tag{3.10}$$

$$P_k = (I - K_k H_k) P_k^- \tag{3.11}$$

where the matrices above are defined as follows

$$A = \frac{\partial f}{\partial x} f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{3.12}$$

$$W = \frac{\partial f}{\partial w} f(\hat{x}_{k-1}, u_{k-1}, 0) \tag{3.13}$$

$$H = \frac{\partial h}{\partial x} h(\hat{x}_k, 0) \tag{3.14}$$

$$V = \frac{\partial h}{\partial v} h(\hat{x}_k, 0) \tag{3.15}$$

and Q and R are as defined in (3.3) and (3.4). This fully defines the time and measurement update steps of the EKF.

The EKF does lose some power because of the linearizion done in the derivation. It only approximates the optimal error, making it a good estimate, but not provably the best estimate. In addition, the distributions can no longer be considered normal.

3.3 Localization

During the link acquisition phase, each robot requires prior knowledge of the location of the partner robot within some error tolerance. Each robot needs to be able to localize itself in an outdoor environment within some radius of error to allow for the partner robot to search for it. To accomplish this goal, a Kalman filter localization approach was used, allowing the robot to navigate to the final waypoint within a small error tolerance.[7][1]

In this case, each robot had a global positioning system (GPS) receiver and a three axis orientation sensor, which allowed for the complete recovery of pose in $\Re^3$. Due to the absolute nature of both the GPS and orientation sensor, the kidnaped

robot problem [5], which Kalman filter approaches cannot handle robustly, is not an issue as the robot will always have an indication to its current location within some error.

We can begin by describing the robot in terms of the global frame, which for experiments was the north-east-down (NED) frame. For our navigation trials, we assumed a locally level area, so that the Kalman filter derivation is only in two dimensions, the north-east frame. From an arbitrary origin, chosen at run time, the robot's position and orientation are described by

$$\vec{x} = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix} \tag{3.16}$$

The motion model of the robot can be describe by the velocity estimate, V, provided by the odometry and the angular velocity, ω, measured by the MicroStrain 3DMG. Both of these measurements are assumed to be corrupted by Gaussian noise. Using this information, we can write the function, $f(\vec{x})$, which describes the time update given in (3.7) as

$$f(\hat{x}_{k-1}, u_{k-1}, 0) = \vec{x}_{k-1} + \begin{bmatrix} V\cos\theta\Delta t \\ V\sin\theta\Delta t \\ \omega\Delta t \end{bmatrix} \tag{3.17}$$

which can be described as taking the last state and adding the translational and rotational components to get the current state. Notice that this is the time update, so if this was used without the measurement update, the error present in the odometry model would cause the state error to grow without bound. This is where the extended Kalman filter is required because of the non-linear nature of the time update equations. To calculate the actual time update, the values for equations (3.12), (3.13), (3.14) and (3.15) must be defined as

$$A = \frac{\partial f}{\partial x} = \begin{bmatrix} 1 & 0 & -V\sin\theta\Delta t \\ 0 & 1 & V\cos\theta\Delta t \\ 0 & 0 & 1 \end{bmatrix} \tag{3.18}$$

which is the amount of change with respect to the state variables $\{x, y, \theta\}$. For the sensor measurements, the Jacobian W is

$$W = \frac{\partial f}{\partial w} = \begin{bmatrix} \Delta t \cos\theta & 0 \\ \Delta t \sin\theta & 0 \\ 0 & \Delta t \end{bmatrix} \tag{3.19}$$

which is simply the amount of change with respect to the process variables $\{V, \omega\}$. These are not the same as the measurement variables, which are dealt with later.

One of the interesting side effects of directly measuring the state using sensors (instead of measuring a different entity and then deriving the state) is that the measurement function becomes trivially

$$z_k = h(x_k, 0) = x_k \tag{3.20}$$

From this result, it is easy to see that the Jacobians for both H and V are both the identity matrix, because both the state, x_k and the measurement variables, z_k are equal. The only matrices left to define are the Q and R matrices. These represent, as in (3.3) and (3.4), the covariance matrices of the process and measurement noise, respectively. As pointed out earlier, all of these matrices do not have to remain constant for the life of the filter, in fact, they may change from time step to time step. This is the case for our R matrix, which we define using both empirical results and direct measurements. The Q matrix is the angular velocity, ω, and linear velocity, V, covariance, which can be measured directly using the three axis orientation sensor and the on-board odometer. The manufacturer of the orientation sensor specifies a standard deviation of 1 degree per second or 0.0174 radians per second for angular velocity. The robot platform does not specific a tolerance for the linear velocity measurement, so this was fixed empirically at 0.08 m/s. This results in a Q matrix defined by

$$Q = \begin{bmatrix} 0.08^2 & 0 \\ 0 & 0.0174^2 \end{bmatrix} \tag{3.21}$$

which was left fixed for the entire trial. The R matrix was slightly different, in that the GPS receiver used could measure the standard deviation directly, using

a proprietary algorithm built into the unit. The orientation sensor was used to measure θ with respect to the NE frame, but it does not provide the tolerance live as did the GPS receiver. The manufacturer's specification defined 5 degrees or 0.087 radians of accuracy over the full range of the unit giving a R matrix defined by

$$R = \begin{bmatrix} GPS_v & 0 & 0 \\ 0 & GPS_h & 0 \\ 0 & 0 & 0.087^2 \end{bmatrix} \tag{3.22}$$

where GPS_v and GPS_h is the measured variance in the x and y world frame at run-time, as measured by the Garmin 18USB unit.

3.4 Controller

Providing a reliable way of migrating the robot to its objective waypoint is an important step in providing some predictable error tolerance at the final waypoint. The prior work had used a simple reactive controller that allowed the robot to follow walls in an indoor environment. For our outdoor environment, a controller had to be added that provided better reliability than a reactive controller could provide. This work was based heavily on previous work in the lab on controllers [10].

Since, our robots are of the differential drive variety, we assume the kinematic model is of the form

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \end{bmatrix} = \begin{bmatrix} v(t)\cos\theta \\ v(t)\sin\theta \\ \omega \end{bmatrix} \tag{3.23}$$

If we assume that $v(t)$ is constant, over the small interval between control inputs, we can derive

$$\ddot{y} = v\cos\theta\dot{\theta} = \omega v\cos\theta \equiv u \tag{3.24}$$

Since, we can place the origin at any point, we assume the origin is on the target waypoint. Instead of creating a smooth path from the robot's starting location to the destination, we choose a straight line as the desired path. Assuming that both the starting and objective points lie on the x axis, we can generate an error term

as the distance off the x axis, $e = -y$. From which we can design a proportional-differential (PD) controller as

$$u = -k_d\dot{y} - k_p y \tag{3.25}$$

where k_d and k_p are the differential and proportional gain constants, respectively. The parameter that we want to adjust is the angular velocity ω. By modeling the controller as a second-order system, we need to regular the acceleration term, $\ddot{y}$. Only by choosing the second-derivative as the control input, u, can we get an expression directly for ω. Substituting (3.23) in (3.25) then solving for ω with (3.24), we can obtain

$$\omega = -k_d \tan\theta - \frac{k_p y}{v\cos\theta} \tag{3.26}$$

Instead of varying both gains independently, the performance of the controller was only measured under critically damped behavior, *i.e.*, $k_d = 2\sqrt{k_p}$. Using [10] as a starting point, k_p was initially set to 1.2. This value did not perform on the mobile robot as well as on the wheelchair platform, due to obvious differences. Via experiments on the robot, a better gain ($k_p = 0.2$) was selected that drove the robot in a "predictable" manner as defined by the human observers.

One of the earlier assumptions required the initial and objective waypoints to lay along the x axis. This would force all of the waypoints to lie along the east-west axis in the world frame, W. Instead, during the initial navigation phase, the vector between the initial and final positions is computed and used as the new x axis. We can then rotate the current state as estimated by the Kalman filter in the W frame into the controller frame, C, where the origin is located at the objective waypoint and the x axis intersects the initial position of the robot. We can compute the rotation matrix as follows

$$R_1 = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.27}$$

where θ is computed by taking the initial and objective waypoints, $\{x_1, x_2 \in \Re^2\}$

$$\theta = \tan^{-1}\left[\frac{x_1(2) - x_2(2)}{x_1(1) - x_2(1)}\right] \tag{3.28}$$

then, by transforming the measured state, $\vec{x}$ from the world frame, W, to the control frame, C by

$$x_c = R_1 \vec{x} \tag{3.29}$$

$$x_c(3) = \vec{x}(3) - \theta \tag{3.30}$$

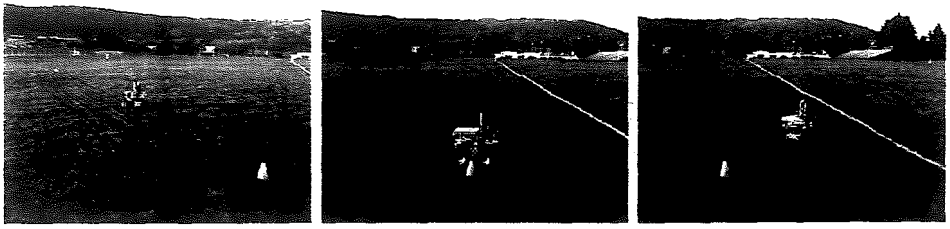
which results in the robot initially facing down the x axis toward the origin.

3.5 Waypoint Trials

Our trials include a course made up of GPS waypoints for the robots to follow, identified (for humans) by orange cones. A robot would traverse a loop attempting to stop on the objective waypoints at each leg of the course. The error would be determined by the distance from the orange cone that the robot stopped and then rotated to the next leg of the course.

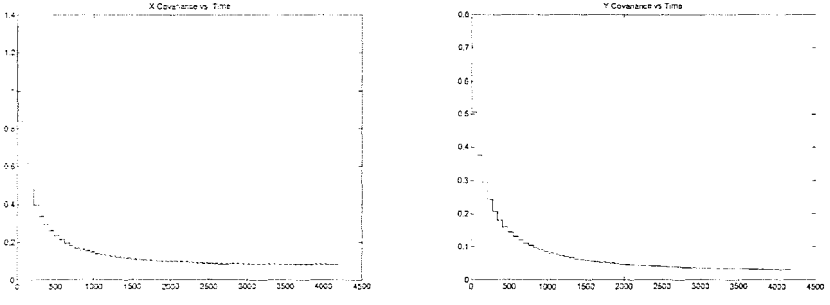
In experimental trials of the Kalman filter, we were able to get the robots to get within 2 meters of a waypoint. In Figure 3.1, the robot can be seen approaching a waypoint (orange cone) from the left, traveling approximately 30-40 meters. In the trial pictured, the robot stops within a meter of the target, rotates to a new objective waypoint and travels toward it on the right.

As shown in Figure 3.2, the covariance matrix decreases over time, estimating the state while minimizing the mean squared error. This data is from a trial conducted outside traveling from one GPS waypoint to another approximately 38 meters away. It is interesting to note that during the time update portion of the Kalman filter, the covariance for both the x and y increase, only to be corrected by the measurement update phase.



(a) Incoming (b) Turning (c) Departing

Figure 3.1: The robot travels toward the objective waypoint, stops, rotates and then continues toward a second waypoint



(a) (b)

Figure 3.2: The covariance of both the x and y pose decrease over time. The "sawtooth" effect demonstrates periods where only the time update is being used.

Chapter 4

Discussion and Future Work

The objective of this work was to investigate some deployment strategies for hybrid FSO/RF networks in an outdoor environment. By using a high zoom camera coupled with a robust tracking algorithm, we were able to locate a link partner using only one template over a wide variety of poses and illuminations. By fusing sensor data from multiple sources, we were able to use a Kalman filter to navigate our robot platforms to target waypoints with a reasonable error tolerance. By demonstrating these technologies, we can clearly see that an outdoor deployment of a hybrid network of mobile robot teams is feasible. To combined these technologies in a production platform, significant future work is needed.

In our navigation framework, we assumed a locally flat area making the Kalman filter derivation two dimensional. This restriction is slightly unrealistic, as most terrain has some elevation associated with it, and the vision/FSO system is capable of connecting at a large variety of angles. This could be extended to three dimensions quite easily, as most of the derivation would be the same. This would allow deployments that connected valleys by creating a link over the ridge.

Although the SIFT approach claims to be illumination invariant, separate tests in our lab have shown that it is quite sensitive to large changes in illumination or contrast. Further research into methods for combating illumination disparity should be investigated.

The work laid out in this paper provides all of the pieces for a strong outdoor

implementation of a hybrid network. Utilizing the waypoint navigation to position the robots at the objective location, then searching for the link partner with SIFT and finally aligning the FSO heads is a daunting integration task. The integration of these pieces and production trials is the source of ongoing research in the coming months.

Bibliography

[1] M. Bruch, G. Gilbreath, and J. Muelhauser. Accurate waypoint navigation using non-differential gps. 2002.

[2] J. Derenick, C. Thorne, and J. Spletzer. *Multi-Robot Systems: From Swarms to Intelligent Automata*, chapter Hybrid Free-space Optics/Radio Frequency (FSO/RF) Networks for Mobile Robot Teams. Kluwer, 2005.

[3] J. Derenick, C. Thorne, and J. Spletzer. On the Deployment of a Hybrid FSO/RF Mobile Ad-hoc Network. In *Proceedings of IROS-05, The IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '05)*. Edmonton, Alberta, Canada, August 2005.

[4] R. Duda, P. Hart, and D. Stork. *Pattern Classification*, chapter Nonparametric Technqiues. Wiley, 2000.

[5] D. Fox, W. Burgard, and S. Thrun. Markov localization for mobile robots in dynamic environments. *Journal of Artificial Intelligence Research*, 11, 1999.

[6] Y. Ke and R. Sukthankar. PCA-SIFT: A More Distinctive Representation for Local Image Descriptors. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2004.

[7] A. Kelly. A 3d space formulation of a navigation kalman filter for autonomous vehicles. Technical Report CMU-RI-TR-94-19, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1994.

[8] D. Lowe. Object recognition from local scale-invariant features. In *International Conference of Computer Vision*, Corfu, Greece, September 1999.

[9] D. Lowe. Distinctive image feature from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004.

[10] H. Sermeno-Villalta and J. Spletzer. Vision-based Control of a Smart Wheelchair for the Automated Transport and Retrieval System. In *accepted to the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*, Orlando, FL, USA, May 2006.

[11] S. Thrun, D. Fox, W. Burgard, and F. Dellaert. Robust monte carlo localization for mobile robots. *Artificial Intelligence*, 128(1-2):99 141, 2000.

[12] G. Welch and G. Bishop. An Introduction to the Kalman Filter. Technical report, University of North Carolina at Chapel Hill, 2004.

[13] J. Wilson. Ultra-wideband / a disruptive rf technology? Technical report, Intel Research and Development, 2002.

Vita

Christopher Ryan Mansley was born on October 12, 1981 in Doylestown, Pennsylvania, USA, the first of two sons of Arlene and Ron Mansley. He graduated from Palisades High School in June 1999 and enrolled in the Computer Engineering program at Lehigh University in August 1999. After graduating with a Bachelor of Science in Computer Engineering in May 2003, he became a full-time employee at Moritz Aerospace. In August 2004, he left his position as lead software engineer to pursue an advanced degree in Computer Science at Lehigh University. He received his Master of Science in Computer Science in May 2006.

END OF TITLE