In IEEE Intnl Conf on Robotics and Automation (ICRA), Pages 3024-3029, Washington DC, May 2002

Laser-Based People Tracking

Ajo Fod Andrew Howard Maja J Matarić
Computer Science Dept.
University of Southern California
Los Angeles, CA 90089-0781, USA
fod@fnord.usc.edu,{ahoward|mataric}@cs.usc.edu

Abstract—
In this paper, we describe a method for real-time tracking of objects with multiple laser range-finders covering a workspace in a parallel and distributed fashion. Tracking people is a popular problem in machine vision. Here we adapt the methods used in vision to planar laser scanners. We group range measurements into entities like blobs and objects. With the help of a Kalman Filter (KF), the tracker smooths object paths and estimates its path even when the underlying object is occluded from all the lasers. We finally evaluate the tracker's performance through four experiments.

Keywords—**Kalman filter, Laser, Human Tracking, Occlusion Reasoning, Map Building, Human Interactions.**

I. Introduction

Tracking people, for surveillance, crowd control, activity recognition and characterization is a well studied problem in machine vision e.g., [22]. Our attempt to track people is inspired by the need in humanoid robotics to perceive, imitate and react to human beings. We present a method of tracking objects using scanning laser range finders - a type of sensor rapidly gaining popularity in mobile robotics. Earlier work on motion tracking for crowded environments using scanning laser range finders by Prassler et.al. [13] use occupancy grids and linear extrapolation of occupancy maps to estimate trajectories. Our methods involve the use of multiple lasers distributed in the environment supported and the use of advanced trajectory estimation algorithms.

In this paper, we discuss a real-time method of laser-based tracking of objects as a step towards the larger goal of identifying people and their activities. We demonstrate the use of a standard off-the-shelf laser range-finder as an efficient sensor for tracking multiple objects in a workspace. Existing commercial laser-based trackers servo on the location of the object being tracked using a single laser beam [1]. They are usually expensive, cannot handle occlusions, and impose limitations on the acceleration and velocity of the object being tracked.

Laser sensing differs significantly from vision in ways that can be exploited for tracking. In vision, variables like color, intensity, depth (if binocular) are available. In contrast, lasers are restricted to a 1D projection of events in one plane of the observable space. Most of the useful information for tracking is in just one parameter - the range to the nearest obstacle on small intervals over an arc. The range measurements are, however, of high accuracy, especially in comparison with other range sensors, such as ultrasound or infra red. Thus, lasers are rapidly gaining popularity for mobile robotic applications such as collision avoidance, navigation and localization[9], [24], and map building[28], [14] where precise measurements of distance have to be made quickly.

In this paper we show that lasers are highly effective in a problem domain where vision has so far dominated. Compared to vision, laser data are sparse but more accurate and efficient, as much less processing is required. Since lasers ranging is based on measurements performed on laser light reflected from target objects, they are not very sensitive to noise from natural sources like ambient light as in the case of vision and require almost no calibration. Furthermore, they have significant range, capable of covering entire rooms or corridors, and are thus ideal candidates to address the tracking problem. As with other line-of-sight devices, lasers are sensitive to occlusions.

We study the problem of tracking multiple moving objects in a workspace covered by multiple lasers. We draw from vision-based approaches to estimate motion recursively from subsequent scans of the laser. We use a version of the constraint formulation for rigid bodies [23], [26] and an abstract representation of objects for tracking. We associate a Kalman Filter (KF) with each object to alleviate the consequences of occlusions and to smooth the effects of sensor noise.

We perform four experiments to measure the accuracy in positioning of the tracker. The first experiment compares the scatter in position estimates of large objects in comparison with a beacon of known diameter. The second shows that the tracker is stable and that no errors develop over time. To quantize the dynamic performance of the tracker, we measure the residual of the KF in estimating the position of objects. A fourth experiment exhibits the capability of the tracker to accurately estimate the object count in the room.

II. Approach

A. Overview

Figure 1 illustrates the overall structure of the algorithm. The laser returns range measurements to the nearest obstacle in polar coordinates. From this we construct a model of

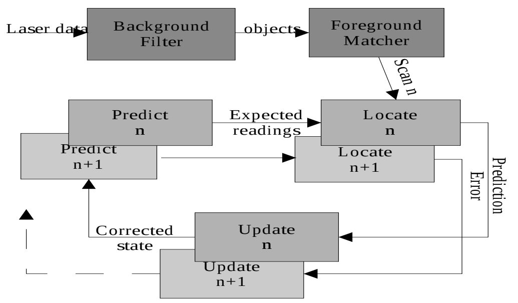

Fig. 1. **Background Filter:** Laser data is first filtered to remove background readings. **Foreground Match and Locate:** blobs in each scan are matched with blobs from the previous scan. **Update:** The error in match is converted to a correction in state estimate. **Predict:** Position of the blob in the next scan is predicted. **Note:** There are as many instances of the Predict-Locate-Update loop as there are blobs, but there is only one instance of the Background Filter and the Foreground Matcher

the process that generates the range readings. The model consists of two parts: the *Background Model* and the *Foreground Model*. The Background Model is used to filter background information that is not of immediate relevance to tracking. The Foreground Model is used to predict range measurements that are not explained by the Background Model. The foreground maintains an estimate of the velocity of each object it tracks. The calculations in the Foreground model involve the perdition of range readings using the velocity and information from the previous scan, as well as re-estimation of the velocity. Repeating this process over a sequence of scans enables us to estimate trajectories of moving objects. Next, we verbally describe the approach, then go into the formal details.

We aggregate measurements that could potentially form a continuous surface in the foreground into “blobs.” At the end of each scan, the Foreground Model consists of two sets of blobs - one from the previous scan and the other from the current scan. Each blob in the current scan is matched or associated with one or more blobs from the previous scan. Blobs are modeled as rigid bodies moving in a plane. Their velocities are stored in a state vector. The state vector associated with a blob in a scan is a linear combination of the states from all matching blobs from the previous scan. After the completion of a scan, the error in prediction of the Foreground readings is used to correct the state of the blob.

Every object manifests itself as a set of blobs. Groups of blobs representing the same object (such as the arms and the torso of a moving human) tend to stay together. Such groups are parsed into a higher level entity called an “object” which is tracked by an *ObjectTracker*. ObjectTrackers have a long term existence and their purpose is twofold. They smooth estimated trajectories of the objects and predict the path of the underlying object through temporary complete occlusions - where no laser can see the object.

B. *Background Model*

Background models used in most vision applications utilize pixel-level statistics [18], [22]. Each new pixel value is classified as the background or a moving object. We modify this scheme in classification using lasers. Since Laser range measurement gives us the range to an object, we can use a modified rule for the classification of foreground and background. We assume that the farthest known stationary object is part of the background. In order to be robust to small errors, we also maintain the mean and the variance of the measurements that are classified as background and use the information to test the hypothesis that any reading is part of the background

Any measurements that cannot be classified as the background are treated as follows:

- If the range measurement is greater than the current background but below the error threshold, the measurement is considered part of a new background.
- Objects that are in front of the current background are processed as part of the foreground.
- Any reading greater than 8m is assumed to be an error (see Section [II-D]); readings above this threshold are ignored.

C. *Foreground Model*

Many methods exist for determining structure of objects independent of motion [21], structure from motion [16], [20], and motion of a known structure [27], [10], [11]. Koller [17] has described a method of tracking objects using a template of the object. We use a method of recursive estimation of motion using a linear model. Effectively, our template of the object for any given scan is given by the previous scan.

The tracking process is divided into two subtasks: 1) tracking blobs and 2) grouping surfaces into objects. A *BlobTracker* tracks the blobs, and an *ObjectTracker* groups blobs into objects and estimates paths of objects in the workspace. The ObjectTracker is discussed further in SectionII-C.2.

C.1 Blobs

Formally, we define a *Blob* as a grouping of adjacent foreground readings from the laser that appear to be on a continuous surface. The set of measurements that form this continuous surface constitute the *signature* of the blob. Since the laser is oblivious to the component of motion perpendicular to the plane of scan of the laser, we do not attempt to model motion or changes along this direction i.e. we only consider planar motion. Within reasonable margins, human beings tend to stay in a plane parallel to the ground in a planar workspace at about the height of the

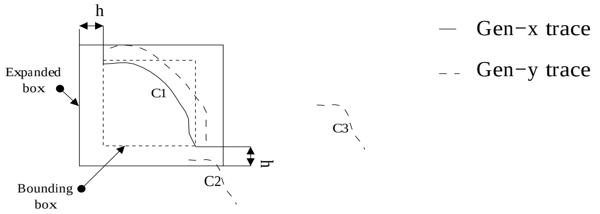

Fig. 2. Consider a single blob belonging to gen-x. The bounding box is first expanded to correct for possible errors. Blobs C1, C2, and C3 are gen-y blobs. Only blobs C1 and C2 are considered for a match. Of these, C1 is strongly linked to the gen-x blob, and C2 is weakly liked to it.

waist - this is the plane we chose to observe activity in. In our experiments, we assume that measurements that are spatially separated by less than 10cm belong to the same blob.

The Foreground model contains two generations of blobs at any given instant: 1) **gen-x**, the old blobs and 2) **gen-y**, the new blobs. To correct the state estimate of every blob, we need to match blobs in gen-x with those in gen-y. To reduce the search space in matching blobs, we draw a rectangular bounding box around any blob. Based on a bound on average acceleration of $10m/s^2$ (about 1g) and the time interval between scans of about 0.2s, we estimate the maximum error in the prediction of the state to be about 50mm. We expand the bounding box in all directions by this amount and search for matches between the signatures of a blob in gen-x and a blob in gen-y only if an overlap exists between the expanded bounding box of gen-x and the simple bounding box of gen-y. Within a blob, a match is said to exist between two points if the distance between them is less than 50mm. We attempt to pair every point in gen-y with two points in gen-x that match and have the minimum distance to the new point. We use these two points for a linear approximation of the surface of the blob. This concept is along the lines of the chamfer method suggested in [4] and is illustrated in Figure2. We use a linear model to predict the state of the blob. Newly discovered gen-y blobs inherit their state vector as a linear combination of the predicted state vectors from the matched gen-x blobs. The weighting of each of the gen-x blobs in determining the state of the gen-y blobs is given by the number of point matches between the blobs:

$$q_y = \frac{\sum_x m_{x,y} q_x}{\sum_x m_{x,y}} \tag{1}$$

where $m_{x,y}$ is the number of point matches between blob x of gen-x and blob y of gen-y. Blobs that do not have any parents are initiated with a state vector of 0. The gen-x blobs are removed and the current gen-y blobs become the gen-x blobs for the next scan.

We use a rigid body model with the signature from the previous scan and a velocity state estimated from earlier scans to predict the measurements from the next scan. The matched blob's signature is also used to update its rotational and rectilinear velocities. For a blob with N points, the space which represents the blob would have $2N$ dimensions. However, a rigid body moving in a plane has only 3 degrees of freedom corresponding to the state vector.

The method of subspace filtering as presented in [23], [**?**] requires one-to-one correspondences between features. It is not easy to find correspondences in the measurements from the laser. In the laser sensor space, our strength is in the ability to determine the range reading accurately - so we try to fit depth maps . We approximate the surface of objects to a sequence of lines connecting predictions of the signature of a blob. Each new j^{th} measurement $\mathbf{x_{j,t+1|t+1}}=(x_j, y_j)$ is paired with the two closest adjacent i and $i+1$ predictions $\mathbf{x_{i,t+1|t}}$ and $\mathbf{x_{i+1,t+1|t}}$ based on the previous observations. We measure the distance from a line joining $\mathbf{x_{i,t+1|t}}$ and $\mathbf{x_{i+1,t+1|t}}$ and the point $\mathbf{x_{j,t+1|t+1}}$ as a measure of the error in our prediction of the point. We thus attempt to fit the new point to a linear approximation to the previous surface. This error is summed over all points in the new surface to obtain a metric that measures the performance of the prediction.

As mentioned earlier, each new point (x_i, y_i) is paired with exactly two predicted points - here they are represented by $\mathbf{x_{i,t+1|t}}=(a_x, a_y)$ and $\mathbf{x_{i+1,t+1|t}}=(b_x, b_y)$. For small errors, the distance E_i of a generic point (x_i, y_i) from the line passing through and $\mathbf{a}$ and $\mathbf{b}$.

$$\Delta x_i = a_x - b_x \tag{2}$$

$$\Delta y_i = a_y - b_y \tag{3}$$

$$\tan\theta_i = \frac{\Delta y_i}{\Delta x_i} \tag{4}$$

$$\bar{x_i} = \frac{a_x + b_x}{2} \tag{5}$$

$$\bar{y_i} = \frac{a_y + b_y}{2} \tag{6}$$

$$E_i = (y_i - \bar{y_i})\cos\theta_i - (x_i - \bar{x_i})\sin\theta_i \tag{7}$$

To determine the impact of errors in the estimate of the velocity of the blob, we determine this distance as a function of these small errors in the velocities Δv_x, Δv_y and $\Delta\omega$. The predicted surface is shown here as a function of the original surface and the parameters.

$$\bar{x_i} \Rightarrow \bar{x_i} + \Delta v_x t - y\Delta\omega t \tag{8}$$

$$\bar{y_i} \Rightarrow \bar{y_i} + \Delta v_y t + x\Delta\omega t \tag{9}$$

$$\theta_i \Rightarrow \theta_i + \Delta\omega t \tag{10}$$

Substituting these into equation 7, we get

$$\begin{aligned} E_i &= t\sin\theta_i(\Delta v_x) - t\cos\theta_i(\Delta v_y) \\ &\quad - t(x_i\cos\theta_i + y_i\sin\theta_i)(\Delta\omega) \\ &\quad + (y_i - \bar{y_i})\cos\theta_i - (x_i - \bar{x_i})\sin\theta_i \end{aligned} \tag{11}$$

This equation is in the form:

$$E_i = \frac{\partial E_i}{\partial v_x}\Delta v_x + \frac{\partial E_i}{\partial v_y}\Delta v_y + \frac{\partial E_i}{\partial \omega}\Delta\omega + c_i \tag{12}$$

$$= \sum_j \frac{\partial E_i}{\partial q_j}\Delta q_j + c_i \tag{13}$$

We now formulate a sum of squares error metric $M = \sum_i E_i^2$. We determine the best correction that can be applied to our estimate of the state. Since the error is quadratic, the exact solution Δq_j^* is given by:

$$\mathbf{\Delta q^*} = -\mathbf{A^{-1}d} \tag{14}$$

where:

$$\mathbf{A_{jk}} = \sum_i \frac{\partial E_i}{\partial q_j}\frac{\partial E_i}{\partial q_k} \tag{15}$$

$$d_j = -\sum_i \frac{\partial E_i}{\partial q_j}c_i \tag{16}$$

C.2 Objects

Objects map to multiple blobs in our scheme. For example, the hands and the torso may form different blobs. These are unified into one percept of an object by the ObjecTraker. Intille et al [25] describe heuristic methods to explain the scenes described by blob trackers in a nine-step procedure which includes the idea of objects. Although our algorithm is based on some of these principles, the accuracy of range measurements changes our approach. For example, we do not need to distinguish between blobs that occupy the same space, since we can identify them easily by using our 2D positioning accuracy.

Each blob inherits its ObjectTracker from its parent in the previous generation. When two blobs form the parent of a blob, the closest ObjectTracker is associated with the new blob. Blobs that appear spontaneously within a prescribed radius of any ObjectTracker are associated with it. Whenever no known ObjectTracker is sufficiently close to a new blob, a new ObjectTracker is constructed for the blob. Once instantiated, ObjectTrackers are allowed to persist until supporting blobs are not visible for over 5 seconds - an empirically determined threshold.

When objects move out of the plane of the laser, the ObjectTracker is left with no blobs to support it. The ObjectTracker continues scanning for objects that can be described by it with less error than by others. The time update steps are performed on the state during the occlusion as suggested in [12], [22]. So, when the object reappears, the ObjectTracker becomes active again as if the object had never disappeared. In this way, we can continue to track objects through temporary occlusions.

The tracker we implemented was designed to work in real time. To do this we made modifications to reduce the processing time of the algorithm by eliminating processing steps that does not contribute to the tracking task. To extract the rigid body motion component of the movement, most of the state information can be obtained from a few laser readings from the blob. We use at most five readings from each blob. Whenever the delay in processing exceeded a certain threshold, we skipped processing frames to speed up the processing. With these modifications, we could track up to 20 objects simultaneously without significant degradation of performance.

D. *Sensor Errors*

While lasers are very accurate in general, they do return erroneous readings occasionally. These errors can be better understood by considering the physical principles involved. The laser scanner uses the time between the transmission and the reception of the reflection of the pulse to determine the range to the observed object. Some surfaces, including very dark objects, do not reflect enough laser light back to the receiver to be visible. Sometimes, the angle of incidence of the laser on a surface is high enough for the beam to be reflected away in a specular fashion. Finally, the laser may not perceive anything within its range. In all these situations, the laser reports a range above 8m and consequently we can not distinguish between them. We handle such errors by ignoring measurements above 8m.

Other sources of error arise from our modeling of the process. The laser scans provide an estimate of the surface of the object being tracked. Since the object is not rigid (people are not), we can expect the predictions of the signature of the object to be inaccurate. For simplicity, we threshold the distance from a predicted scan to the actual scan to match objects in subsequent scans. Increasing this threshold would result in more frequent matches which might be inaccurate. On the other hand, decreasing it might lead to lost objects.

III. Experimental Procedure

Our experiments were based on real laser scan sequences. In this Section we report our observations from experiments conducted on data obtained in our lab. We used Player [8], a server and protocol that connects robots, sensors and control programs across the network to communicate with our laser range-finders. The latest version of this software can be found at [19].

We used standard SICK planar scanning laser rangefinders placed $0.75 - 0.80m$ above ground level, targeting the waist height of an average walking person. This is the cross section which moves least. Moreover, rarely do people move out of this plane. For example, a person who sits would still be visible albeit at shoulder height.

We performed four sets of experiments to measure the performance of the tracker. Three experiments were performed on the tracker to measure the accuracy with which the tracker estimates the position of objects in the workspace and one experiment was used to measure the accuracy in estimating the number of objects in the workspace.

A. Apparatus and Methods

We made two sets of measurements in each experiment set. The first measurement - the ground truth - was recorded using a camera and post analyzed by a human operator. The second was the output from the tracker. The experiments were performed with multiple lasers in a single room. Our choice of a single room as opposed to multiple rooms was made to make the measurement of ground truth easier.

B. Experiment 1: Positioning Accuracy

The tracker maintained the position estimates of objects being tracked. We performed the following two tests to determine the accuracy of this estimate. The first set used laser beacons which are well behaved i.e stationary and of small diameter and estimate a point closely. This was done to measure the performance of the tracker in the absence of any noise arising out of factors like the shape, size and movement of human beings. The second set was performed with human subjects. The goal here is to measure the performance of the tracker in the presence of such noise.

B.1 Methods

We obtained two measurements of position for each sample point: one in the laser frame of reference an the other in the visual frame of reference (henceforth frame of reference is abbreviated to frame). Figure 3 shows the room used in the experiment. In order to get measurements in the visual frame, we positioned beacons at corners of 2’ x 2’ floor tiles. A second set of measurements of the positions of the beacon was made in the reference frame of the laser.

We determined the transformation T from the laser frame to the visual frame which minimized the mean squared distance between laser beacons sensed in the visual frame and the transformed laser frame. This transformation is used subsequently to map the position of any object in the laser frame to the corresponding point in the visual frame.

The beacon positions are shown in Figure 4. The error or positional inaccuracy can now be plotted as a set of 2D displacements from the position measured in the visual frame to the position measured in the transformed laser frame. In the ideal case of zero error, these vectors would all be located at the origin. This plot is shown in Figure 6.

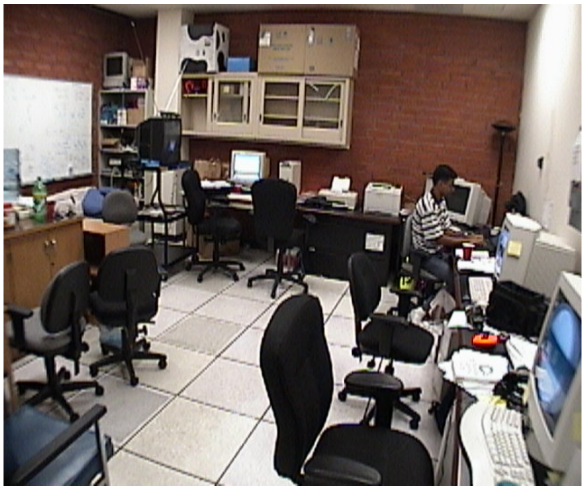

Fig. 3. **Exp 3:** An image of the room used in the experiments.

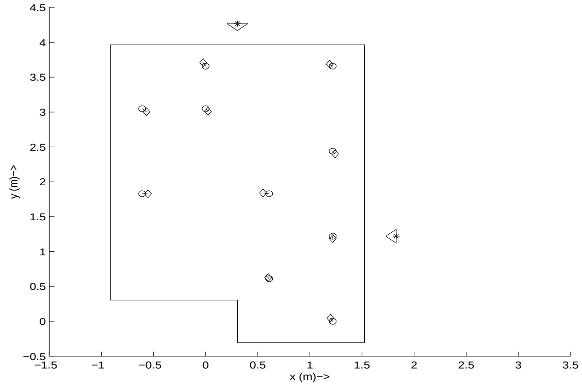

Fig. 4. **Experiment 1:** The positions of the beacons in the room. The diamonds show the location of the beacon in the laser frame and the circles in the visual frame. Correspondences are connected with a line. The two lasers are marked with a “*” and the arrow shows the direction the laser scanners face.

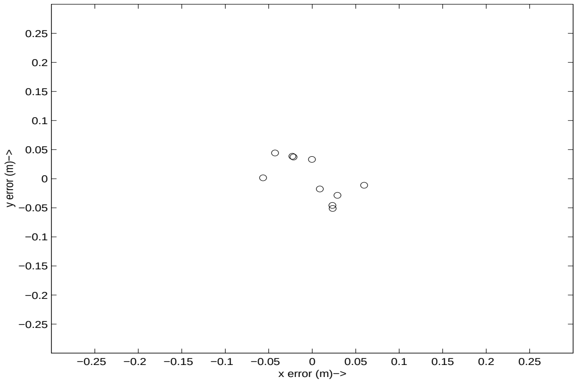

Fig. 5. **Experiment 1:** Each point represents the displacement required to map the measurement of the beacon in the laser frame to the visual frame. This effectively shows the error vector for position estimates made by the tracker.

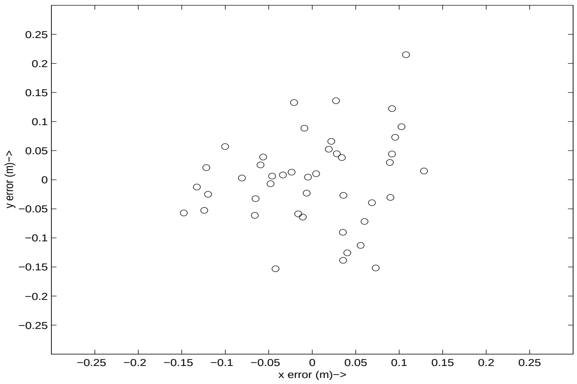

Fig. 6. **Experiment 1:** The plot shown in Figure 6 is repeated with people in this figure to show the change in the scatter distribution. Notice that it is now of the order of the cross section of a human torso.

We performed a second set of measurements of people standing on tile corners. For this set, the subjects moved between tile corners and waited for about a second at each position. This was done to make it easy to find correspondences between positions in the visual frame and the laser frame.

We used the transformation T determined using the beacons to measure the error in position estimate measured by the laser for the human movement data. The scatter plot for this set of measurements, shown in Figure **??**, shows that the error is now of the order of the cross section of a human torso: about 0.3m.

C. *Drift in Tracking*

We performed the following experiment to determine if any systematic error developed over time in the position measurements.

C.1 Methods

We performed a set of measurements along the lines of the second test in Experiment 1 for a duration of 20 minutes within the specified workspace. At least 40 correspondences between the laser and the visual frames were recorded starting at every 5 minute interval. We superposed the paths followed by a person in the visual frame with that in the laser frame Figure 7 and plotted the positional error as a function of time in Figure8. The figures show that there is no noticeable drift or systematic increase RMS error over this period.

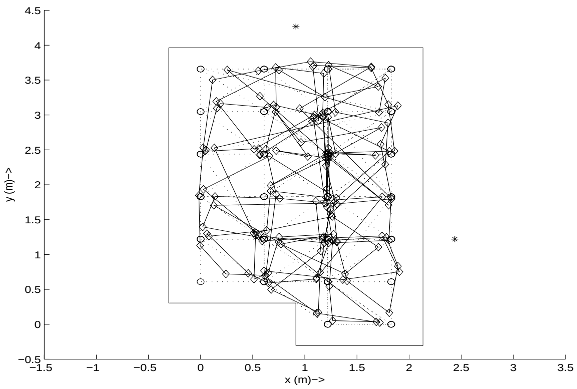

Fig. 7. **Experiment 2:** Paths in the laser frame (dotted lines) superposed with paths in the visual frame (solid lines).

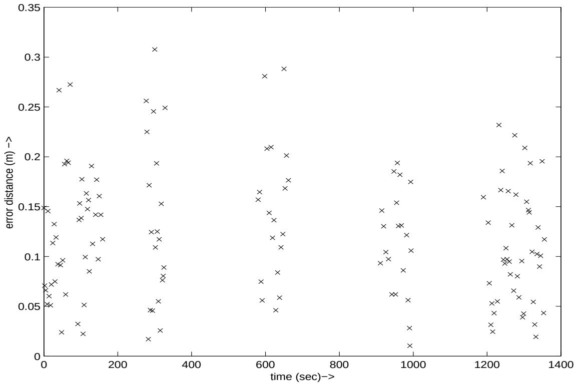

Fig. 8. **Experiment 2:** The magnitude of the error is shown here as a function of time for individual measurements. These measurements were made at 40 points separated by 5 minutes. No points were plotted where measurements were not made. The figure shows that the drift does not increase significantly over a 20 minute period.

D. *Experiment 3: Dynamic Performance*

To measure the performance of the KF in estimating the state of the object, we developed an error metric which measures the error in predicting the trajectory of the object. We defined two quantities for every object - $\Delta\bar{\mathbf{c}}_\mathbf{t}$: the observed change in the position and $\Delta\hat{\mathbf{c}}_\mathbf{t}$: the predicted change in the position at time t. We now define the error metric:

$$\epsilon_1 = \frac{1}{T}\sum_{t=1}^{T}|\Delta\bar{\mathbf{c}}_\mathbf{t} - \Delta\hat{\mathbf{c}}_\mathbf{t}|^\mathbf{2} \qquad (17)$$

ϵ_1 is called the residual and is a measure of the instantaneous uncertainty or variance in the error of the position estimate of the blob. The variance was in the range $1 - 5cm^2/s$. As expected, it was larger for objects that move around like people and smaller for objects that are stationary.

E. *Experiment 4: Tracked number of objects*

We performed the following experiment to determine the accuracy of the laser in determining the number of people in the room.

E.1 Methods

We ran the tracker on a scripted play 9 which involved normally observed activities in the room such as walking,

time	*action*
0.00	Start of program - with A at *Station A*
1:33	A moves fully (initialization)
1:40	B walks into the room, goes to *Station B*, and sits down
1:56	A gets up and walks over to *Station B* to talk to B
2:15	C walks in for a cup of coffee to the *coffee table*
2:25	C walks back out with cup
2:40	D walks in and shakes hands with A
2:49	D pulls up a chair next to *Station A* and sits down
2:58	A goes to *Station A* and sits down to talk
...	...
7.03	A walks to one of the moved chairs
7.10	B walks in towards the *printer*
7.12	A and B cross paths
7.15	A bends below the laser's range
7.19	A gets back in the range
7.27	A walks back to *Station A*

Fig. 9. **Experiment 4:** Excerpts from the script used in the test.

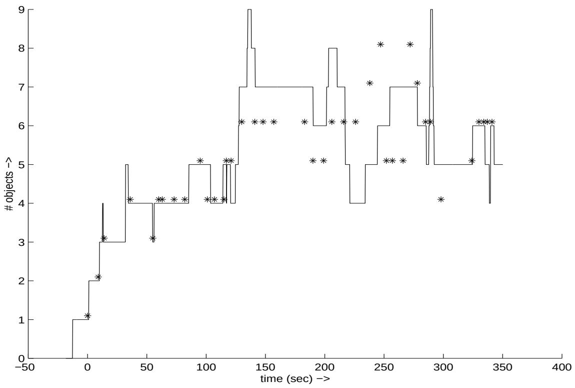

Fig. 10. **Experiment 4:** The test shows the accuracy in tracking the number of objects in the room. The stars represent the number of objects as measured by the camera and the solid line the number tracked by the tracker.

talking, discussions, and path crossings.

The tracker can only see or track objects that have moved and it can not differentiate between people and objects. So we use the visual ground truth measurement to find the number of objects that have moved. Figure 10 shows the number of objects that have moved as measured by the tracker compared with the visual measurements.

E.2 Results

The errors in estimating the number of objects in the room were analyzed with the aid of the script and the video. ObjectTrackers are designed to track objects that have moved once. Consequently chairs appear spontaneously near previously existing objects (people). While the tracker retains separate identities for people and chairs, when people are sitting on chairs, it becomes difficult at times as when the chair is temporarily occluded, to continue to distinguish between the chair and the person. In these cases, the tracker underestimates the number of objects in the room. This phenomenon occurred between 220 and 270 secs in the Figure 10. On the other hand, the number of objects in the room is overestimated when they move very fast or when they jump and the KF is unable to keep up. This sometimes results in temporary objects and is manifested in spikes in Figure 10 at about 150s, 220s, and 290s. Occasionally, the newly spurious object becomes the parent of a stable blob. When this happens the number of objects is overestimated for an extended period of time like between 170s and 200s.

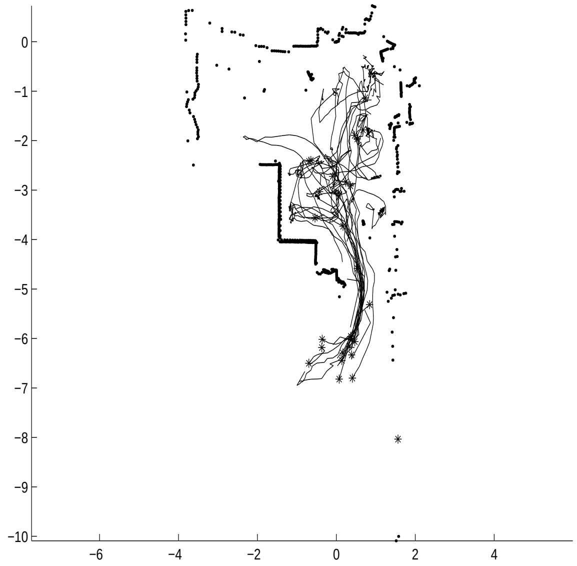

Fig. 11. **Scripted Test:** Shows the paths of people in a room in the scripted test for Experiment 4.

IV. Discussion

In this paper we have introduced a novel laser-based tracking approach. Our method exploits the advantages of lasers while overcoming some of the shortcomings of vision like processing requirements. Lasers provide a "gist" of the scene in a plane very accurately. Unlike vision, where multiple cameras are required to cover a scene, a single laser can effectively cover a reasonably large workspace. Owing to the sparse and accurate nature of laser data, they are easier and computationally cheaper to handle. However, it is difficult to obtain a set of features that uniquely distinguish one object from another in laser data. Occlusions are the greatest problem; while it is possible estimate the identities of objects that have been occluded, based on the velocity and position at the beginning of occlusion, it is difficult to distinguish between them positively. Figure 11 shows the paths of people for the scripted test used in experiment 4. The corresponding image of the workspace is shown in Figure 3.

Our tracking algorithm is computationally light-weight; it works in real time on an IBM 600e Think-pad with 64MB RAM and a 300MHz P-II processor with 2 lasers and up to 10 objects. We have also collected extensive data with a 900MHz P-III with 128MB RAM using up to 6 lasers covering two rooms.

Using lasers has its drawbacks. Lasers are oblivious to objects outside their plane of scan, thus making it difficult to perceive details of activity. It is possible to use multiple lasers in different planes to build up a 3-D representation of the scene. Multiple lasers can also be used in large workspaces with a vast numbers of people, where occlusions prohibit effective tracking. Unfortunately, as a practical matter, lasers are currently expensive relative to cameras.

A. Applications

Gait recognition and action modeling are among the interesting applications of trackers that can store and process entity-specific information. Bregler and Brand [7], [5], [6] use motion capture to record joint orientation data for movement analysis. One could modify these to use laser data to determine patterns in the movement in a plane possibly to identify the task being performed or even to identify characteristics of the person.

Microsoft's research on EasyLiving Technologies [15], [3] is a commercialization of this technology which requires locating and tracking individuals in a closed workspace. They use this information to guess the intent of the user and help in automating some of the mundane tasks like switching appliances on in the presence of a user, setting parameters like the temperature in a workspace automatically or even logging on automatically in the presence of a user.

Robotic security devices [2] need to identify moving objects and possibly determine any helpful information about them. One feature would be to find out the number of objects, if the object is an intruder, and if so to determine his physical characteristics. These features can potentially be determined by using a laser tracker.

In this tracking application, we have filtered out the background. However, we could instead filter out the foreground. Moving the laser would then provide us with an estimate of the velocity of the sensor. We have used the KF to estimate a differential state. We could instead use the KF to estimate the displacement directly. This would be applicable to laser-based map building, navigation and localization as suggested in [14], [9], [24].

V. Conclusion

We have argued that lasers are an effective sensor for object and people tracking. We have evaluated the performance of our KF to show its effectiveness in tracking objects behind occlusions. We have also developed a computationally efficient algorithm for multiple object tracking and validated it in a laboratory workspace.

In our continuing work, we are scaling up the presented approach to track large groups of interacting humans in complex environments with multiple lasers. We have used attributes that can be measured from laser data to qualify objects and shown how this can be useful.

Acknowledgements

The research described here was supported in part by the NSF Career Grant IRI-9624237 to M. Matarić, and in part by ONR DURIP Grant N0014-00-1-0140 for equipment support to M. Matarić and G. Sukhatme.

References

[1] Laser tracking system. http: //mehttp: //members.aol.com /apisensor /lastrack.htm. Last accessed March 2001.

[2] Security robots. http:// www.nosc.mil/ robots/ land/ robart/ history.html. Last accessed March 2001.

[3] J. Krumm B. Brumitt, B. Meyers. Easyliving: Technologies for intelligent environments. In *Handheld and Ubiquitous Computing*, September 2000.

[4] H. G. Barrow. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proc. IJCAI*, volume 2, pages 650–663, 1977.

[5] M. Brand. Understanding manipulation in video. In 2^{nd} *International Conference on Face and Gesture Recognition*, Killington, VT, 1996.

[6] M. Brand. Style machines. In *SIGGRAPH 2000*, New Orleans, Louisiana, USA, July 23-28 2000.

[7] C. Bregler. Learning and recognizing human dynamics in video sequences. In *CVPR97*, 1997.

[8] Richard T. Vaughan. Brian Gerkey, Kasper Stoy. "player robot server". Institute for Robotics and Intelligent Systems Technical Report IRIS-00-391, University of Southern California, 2000.

[9] S. Thrun D. Fox, W. Burgard. Markov localization for mobile robots in dynamic environments. *Artificial Intelligence*, 11:391–427, 1999.

[10] D.Gennery. Tracking known 3-dimensional objects. In *Proc. AAAI 2nd Natl. Conf. Artif. Intell.*, pages 13–17, Pittsburg, PA, 1982.

[11] D.Gennery. Visual tracking of known 3-dimensional object. In *Int. J. of Computer Vision*, volume 7, pages 243–270, 1992.

[12] J.Malik D.Koller, J.Weber. Robust multiple car tracking with occlusion reasoning. In *Third European Conference on Computer Vision*, LNCS 800. Springer-Verlag, 1994.

[13] A. Elfes E. Prassler, J. Scholz. Tracking people in a railway station during rush hour. In H.I.Christensen, editor, *Proc. Computer Vision Systems, First International Conference, ICVS'99*, pages 162–179, Las Palmas, Spain, 13-15 Jan 1999. Springer-Verlag.

[14] A. Reina J. Gonzalez, A. Ollero. Map building for a mobile robot equipped with a laser range scanner. In *IEEE Int Conference on Robotics and Automation, San Diego, CA, USA*, San Deigo, CA, USA, May 1994.

[15] B.Meyers J.Krumm, S.Harris. Multi-camera multi-person tracking for easyliving. In *IEEE Workshop on Visual Surveillance*, 2000.

[16] Thomas J.Oliensis, J.Inigo. Recursive multi-frame structure from motion incorporating motion error. In *DARPA Image Understanding Workshop*, 1992.

[17] D. Koller. Moving object recognition and classification based on recursive shape parameter estimation. In 12^{th} *Israel Conference on Artificial Intelligenc, Computer Vision*, December 27-28 1993.

[18] B. Brumitt K.Toyama, J.Krumm. Wallflower principles and practice of background maintenance. In *International Conference on Computer Vision*, pages 255–261, Corfu, Greece, Sep 1999.

[19] Robotics Lab. Stage home page. http:// robotics.usc.edu /player/.

[20] T.Kanade L.Matthies, R.Szeliski. Kalman filter based algorithms for estimating depth from image sequences. *Int. J. of computer vision*, 1989.

[21] G.Medioni M.Lee. Structure and motion from a sparse set of views. In *Proceedings of the IEEE Conference on Computer Vision*, Coral Gable, Florida, Nov 1995.

[22] S.Sclaroff R.Rosales. Improved tracking of multiple humans with trajectory prediction and occlusion modeling. In *IEEE Conf. on Computer Vision and Pattern Recognition*, Workshop on the interpretation of Visual Motion, Santa Barbara, CA, 1998.

[23] P.Perona S. Soatto, R.Frezza. Motion estimation via dynamic vision. In *Proc. of the 33rd IEEE Conf. on Decision and Control, CDC 94*, 1994.

[24] W. Burgard S. Thrun, D. Fox. Robust monte carlo localization for mobile robots. *To appear in Artificial Intelligence*, 2000.

[25] A.Bobick S.Intille, J.Davis. Real-time closed world tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 697–703, Puerto Rico, June 1997.

[26] P.Perona S.Soatto. 3d visual mtion estimation using subspace constraints. In *Proc. of the 1st IEEE Conf. on Image Processing*, 1994.

[27] R.Chellappa T.Broida. Estimation of object motion parameters from noisy images. In *IEEE trans. PAMI*, Jan 1986.

[28] S. Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 1:21–71, 1999.