

Tracking of Moving Objects from a Moving Vehicle Using a Scanning Laser Rangefinder

Robert A. MacLachlan, *Member, IEEE*, Christoph Mertz

Abstract— The capability to use a moving sensor to detect moving objects and predict their future path enables both collision warning systems and autonomous navigation. This paper describes a system that combines linear feature extraction, tracking and a motion evaluator to accurately estimate motion of vehicles and pedestrians with a low rate of false motion reports. The tracker was used in a prototype collision warning system that was tested on two transit buses during 7000 km of regular passenger service.

I. INTRODUCTION

It is vital that any vehicle avoid collisions with other objects. To operate in real-world environments, autonomous vehicles must successfully avoid collisions with other moving objects (people, vehicles.) Conventionally driven vehicles also collide far too often, and collisions could be reduced by warning of imminent collisions.

Collision avoidance and warning both require prediction of the future positions of moving objects. Given the current position and motion, we can extrapolate that motion into the future. To do this we must accurately measure object position and motion, even when the sensor itself is moving.

We have built a tracker system that has been used in an extensive on-road collision warning experiment [1][2][3] and that has also been tested on an off-road unmanned vehicle. The tracker detects moving objects, estimating their position and motion, while largely compensating for self-motion-induced changes in the scan. The tracker processes 75 scans per second when running on a 600 MHz embedded processor, and can fuse data from multiple scanners for 360° surround sensing.

II. CHALLENGES

Tracking of moving objects from a moving vehicle is a challenging problem, demanding high sensor performance. Ranging sensors are well suited to this problem because the relative motion can be calculated by the derivative of the position and the absolute velocities are derived by a simple

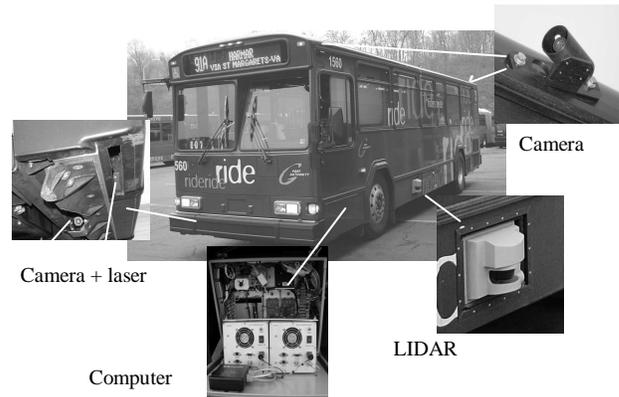


Fig. 1: Collision Warning System Configuration

self-motion coordinate transform. Scanning laser rangefinders such as the SICK LMS 200 (used in our system) reliably detect objects, have good resolution and are affordable.

Unfortunately, the object shape appears to change as different aspects of the object come into view, and this change can easily be misinterpreted as motion.

The fundamental problem is that it is necessary to choose some fixed reference point on the object in order to detect change in position. If the reference point is not truly fixed, then there is false apparent motion.

The severity of the shape change problem depends primarily on the object size, the desired speed measurement accuracy and the time available for detection. If the apparent center of the object is used as the reference, then due to angular resolution limits, the reference can shift by more than 1/2 the object size in a short time. This happens when the long side of an object suddenly becomes visible. The geometry of a severe case is illustrated in Fig. 2. If the max object size is 10m, and the detection time is 0.5 sec, then we might infer a spurious velocity of up to 10 m/sec. This far exceeds our desired threshold of 0.7 m/sec (a person walking.) We must do much better than this, since spurious velocities on fixed objects can easily cause false collision warnings.

III. RELATED WORK

Some work has been done on vehicular applications of laser scanners. The IBEO company has developed both scanner hardware and proprietary software for a prototype vehicular safety system. Reference [4] describes capabilities for pedestrian classification and correctly segmenting

Manuscript received February 27, 2006

R. A. MacLachlan is with the Carnegie Mellon University Robotics Institute, Pittsburgh, PA 15213 USA (phone 412-445-8113, e-mail: robmacl@ieee.org).

C. Mertz is with the Carnegie Mellon University Robotics Institute, Pittsburgh, PA 15213 USA (e-mail: cmertz@andrew.cmu.edu).

This work was supported in part by the U.S. Department of Transportation under Grant 250969449000, PennDOT grants PA-26-7006-02 and PA-26-7006-03 and the Robotics Collaborative Technologies Alliance under the ARMY Prime Contract Award Number - DAAD19-01-2-0012

objects in the presence of occlusion, but gives little implementation detail. Odometry is used to estimate self-motion, removing the kinematic effects of sensor motion. A Kalman filter is used for velocity estimation. Tracks are classified as car, pedestrian, etc., using apparent shape and behaviour over time.

Reference [6] shares some of the authors of [4], and describes a similar system with additional details. The Kalman filter estimates motion based on the position change of an estimated track center-point. The object classification is used to fit a class-specific prior rectangular model to the points. Although not explained, this appears to be an approach to reducing shape-change motion artifacts. The success of this would depend on the correctness of the classification and the prior model. Each object class also has distinct fixed Kalman filter parameters. A multi-hypothesis approach is used to mitigate the effect of classification error.

Reference [5] generalizes Simultaneous Localization and Mapping (SLAM) to allow detection of moving objects, relying primarily on the scanner itself to measure self-motion. An extended Kalman filter with a single constant velocity model is used in a multi-hypothesis tracker. The emphasis appears to be on mapping in the presence of moving objects, rather than the real-time detection of moving objects when no map is needed.

Building a map (or occupancy grid) appears to offer an accurate way to detect moving objects: if a location is seen to be newly occupied something must have moved into that location. One problem is that maintaining an occupancy grid is expensive; reference [7] addresses this problem with a sparse representation of open space. Yet the grid does not solve the shape-change problem because we cannot rule out the possibility that the object was there all along but we did not see it due to occlusion or range limits. The effect of range limits is particularly intractable because it depends on the unknown target reflectivity.

Several papers describe indoor people tracking systems that use laser scanners. Shape-change effects are mild when tracking people because people are compact compared to typical sensing ranges and lack the flat surfaces that cause abrupt apparent motion when viewed from different directions. Although a moving scanner will see shape change in large objects such as desks, large objects can simply be discarded because they are clearly not people.

In [8] motion is measured by registering old and new scans using chamfer fitting. A constant velocity, constant angular velocity Kalman filter is used. Because the scanner is placed above the leg level, a rigid body model is satisfactory. Although this paper does not use moving scanners, it is noteworthy because of its attempt to quantitatively evaluate performance without ground truth by measuring the position noise of stationary tracks, the measurement residue of moving tracks, and the occurrence of false positive and false negative errors in moving object detection.

There is a large literature on tracking techniques developed for long-range radar which can be applied to robotic applications [9]. However, the low resolution and long range means that all objects can be treated as point-like. Reference [10] adapts the idea of a Joint Probabilistic Data Association Filter (JPDAF) to use a particle filter approach. In comparison to a Kalman filter, at the cost of more computation, a particle filter can accurately model the non-elliptical probability distributions that arise when objects move long distances in the presence of heading uncertainty. The warning component of our collision warning system does use a particle filter, since it must predict object positions seconds into the future using only the current motion estimate. However, at 75 scans/second, objects typically move by much less than their size, so the effects of shape change is much larger the accumulated motion uncertainty from the dynamic model.

IV. LINEAR FEATURE TRACKING

To handle the shape change problem, we have developed a system that tracks linear features (lines and corners) in the scan data. The main advantages of this feature tracking are:

- Corners are stable features not affected by shape change,
- It is efficient because the amount of data is greatly reduced by condensing it to a few features, and
- A good rectangle corner fit is evidence that an object is a vehicle, not a bush or ground return.

Though this approach is named “Linear feature tracking”, the exact nature of the stable features is not crucial. The true unifying theme is rejection of false motion, and success lies in attention to many details. The main steps in tracking are: segmentation, feature extraction, data association, Kalman filtering and track validation.

V. SEGMENTATION

Segmentation divides the scanner data points into distinct objects. Distance threshold segmentation clusters points that fall within a fixed distance threshold. This simple top-down segmentation scheme works quite well, and was used in the experimental evaluation below.

Segmentation also determines which object boundaries are apparently due to occlusion by a foreground object, and thus spurious. After segmentation, the points are transformed into a non-moving coordinate system using a pose estimate derived from odometry and inertial yaw rate.

VI. FEATURE EXTRACTION

Stable feature extraction is crucial to minimizing shape-change artifacts. We fit a rectangular model to the points, and use the corners of the rectangle as features. The scanner can see at most two sides of the rectangle, so we

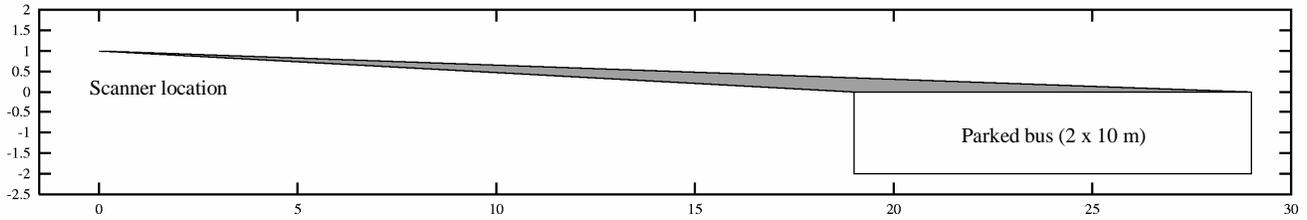


Fig. 2: Effect of 1° angular resolution with shallow incidence angle (scale in meters.)

trial fit both a line and a right-angle corner, only choosing the corner fit when it is significantly better than the line fit.

To increase robustness of the fit against violations of the rectangle model (such as rounded corners and indentations) we use a robust fitting algorithm (Fig. 3). The basic algorithm is a standard least-squares fit with some points discarded and point variable point weighting.

After a trial fit, the worst fitting 20% of the points are discarded. We also de-weight points in areas that are densely sampled (due to being closer to the scanner.) This reduces the tendency of deviations in the shape of nearer parts of the object to overwhelm sparser points on more distant parts of the object which would otherwise have more leverage on the line orientation.

All objects are given a linear fit, no matter how poor. Generally, we require a reasonably good linear fit before we will consider an object to be moving, but we make an exception for small objects (such as people.) Requiring a good linear fit helps to reject apparent motion on clutter objects such as bushes.

VII. DATA ASSOCIATION

In order for the tracker to measure the change in object position over time, we must determine which new segment corresponds to which existing track. This is done primarily by detecting spatial overlap between the predicted track rectangle model and the segment rectangle.

A. Ambiguous Overlap

When objects split, merge or are just close together,

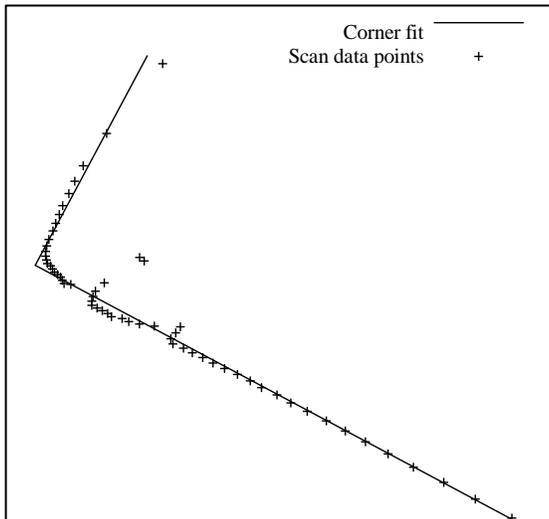


Fig. 3: Robust corner matching with outliers and round corner

overlap may not create a unique association. In this case, we choose among the overlapping options according to the *closeness* of potentially corresponding features. The closeness between a track and a segment is the sum of $1/\text{distance}$ for the closest potential correspondence of each feature. Unlike a direct distance measure, this emphasizes the best matching features.

One important case of ambiguous overlap is when tracks split or merge. Though actual splitting and merging does happen, apparent split/merge is more often due to inconsistent segmentation. Whatever the cause, it is useful to know if track creation or destruction is due to split/merge, so we annotate tracks accordingly.

B. Track Creation and Death

If we fail to associate a measurement segment with any existing track, and the segment has at least three points, then we create a new track.

If we fail to associate an existing track with any measurement segment, then we may delete the track. Tracks are deleted if they have not been associated for 0.2 seconds (or 4 seconds if the track is totally occluded by a foreground object.)

C. Multiple Scanners

The tracker can process input from multiple scanners, allowing 360° coverage of the vehicle surroundings. This is done by segmenting and associating each individual scan separately. If the same object appears in more than one scanner, it is the responsibility of data association to associate the same track with both measurements.

VIII. DYNAMIC MODEL AND KALMAN FILTER

Once we know the correspondence between an existing track and a new measurement segment, we use a Kalman filter to estimate the position, velocity, acceleration and turn rate of the object.

The dynamic model used is constant linear acceleration, constant turn rate. Fixed process noise is chosen to give the best trade-off between smoothing and response to maneuvering. The turn rate is estimated as the derivative of the heading. Normally the linear feature orientation is used as the heading. If a segment has a poor linear fit or is rounded enough that it is not clear whether to fit as a line or corner, then we use the direction of the linear velocity.

A. Measurement Noise

We have found that detailed modeling of the feature position uncertainty is crucial for minimizing the velocity estimation error caused by motion-induced shape change.

The two dominant sources of feature position uncertainty are: angular resolution limits and instability in segmentation or feature extraction. We address these two classes of uncertainty by combining two noise estimates: *static* and *adaptive*.

We decompose the feature position uncertainty into *longitudinal* (along the line) and *lateral* (normal to the line) uncertainties. When measured with the SICK laser scanner (at the ranges of interest), angular resolution contributes more position uncertainty than range resolution or range noise, so the longitudinal uncertainty normally exceeds the lateral uncertainty (Fig. 2).

B. Static Measurement Noise

The static noise is computed from the data in a single scan, and accounts for geometric effects due to resolution limits, incidence angle, and also for target dependent properties such as line fit quality and missing returns.

The RMS line fit error is used as the feature lateral position noise and the longitudinal noise is a constant times the max inter-point spacing along the line of the 7 points nearest the feature.

At long ranges and shallow incidence angles, the inter-point spacing can exceed the segmentation threshold, causing the line to end at that point even though the object extends on; the longitudinal uncertainty is then effectively infinite. In this case, the line end should not be interpreted as a stable corner feature.

Such features are annotated as *vague*, and processed with a modified Kalman filter update that zeros the longitudinal component of the position innovation. This same mechanism is used to suppress the apparent motion of endpoints caused by the moving sensor shadows of occluding foreground objects.

C. Adaptive Measurement Noise

Measurement noise is adaptive, de-weighting features that have unstable positions. After a feature has been associated 15 times, the filtered covariance of the position measurement residue (the difference between prediction and measurement) is used as the measurement error. The static noise covariance times 0.3 is added to the adaptive noise to insure a lower bound.

If the mean of the residue is not nearly zero, we are not tracking properly. When the residue mean exceeds 10 cm, we reset the feature position from the current measurement and clear the residue mean to zero. The need for this resetting comes from a sort of instability that the noise-adaptive tracker exhibits. If a feature does not track accurately (perhaps due to data fusion correctly rejecting spurious motion), then the measurement covariance for that feature becomes inflated, and this further degrades the tracking performance for that feature. By resetting the position when the residue mean becomes too large we keep

the position approximately correct, but allow fusion to continue largely ignoring the feature.

D. Data Fusion

Because we have up to three features for each track, we are faced with the question of how to combine the motion estimates from the separate features into one track motion estimate. The Kalman filter provides a natural framework for data fusion. Each feature is an independent measurement of the position. Because each feature has its own error model, the Kalman filter weights the contribution of each feature appropriately.

Each feature has an independent position, but its incremental motion is treated as an independent measurement of the track's fused incremental motion. The Kalman gain for each feature measurement is computed using the feature's position noise and the track state covariance. The position part of the track state change is applied to both the feature position and the track incremental motion.

The effect of fusing the feature incremental motion is that we do not use any single fixed reference point on the object. Due to the changing perspective of the scanner, no stable single reference point exists because no point is always visible. Attempting to use the object center as a reference point proves particularly futile because the center can rarely be directly measured.

E. Information Increment Test

Based on the measurement noise, each measurement contributes a certain amount of information to the fused track estimate. If the information increment is low, then the influence on the state is small — in this case, the tracker is not actually tracking anything, just speculating based on past data.

A common problem situation is that a track may change to a line with both ends *vague* (i.e. due to resolution limits or occlusion, both ends effectively extend to the horizon.) In this case, the track is localized in one direction only, and the longitudinal velocity is unmeasured, which can lead to unacceptable erroneous velocities.

The information increment is computed from the ratio of the covariance eigenvalues before and after the update. If either eigenvalue fails to decrease significantly, then we fail the association. Tracks that are consistently not associated are eventually deleted; see "Track Creation and Death".

F. Dynamic Limits

Tracker performance is improved in the presence of outlier measurements by either discarding physically impossible measurements or by limiting the innovation to a physically plausible value.

If the change in track state due to a measurement is too incredible, then we discard the measurement. This is done when the Mahalanobis distance of the state change exceeds 6. Before discarding the measurement, we see if we can get a reasonable association by resetting a subset of the features.

We reset any features that contribute a state change with Mahalanobis > 4 .

The time rate of change (d/dt) of velocity, acceleration and angular velocity are limited to physically plausible values. This prevents unreasonable jumps in position from causing big velocity jumps.

IX. TRACK VALIDATION

While robust feature extraction, data fusion, noise modeling and dynamic limits are all important for minimizing velocity errors, there remains an unacceptably high level of false velocities (see Fig. 5.) We apply a separate *track validation* procedure that determines:

- Whether recent observations are consistent with rigid body motion under the dynamic model, and
- Whether there is sufficient evidence to conclude the object is definitely moving.

We have found that rejecting tracks with inconsistent or ambiguous motion greatly reduces false motion detection in the presence of ground returns (scan plane strikes the ground), missing returns (poor reflectivity) and feature extraction errors.

The validation procedure can be understood as a batch smoother. Given the last 35 segments (measurements) associated with each track, we check how well the track path matches up with the measurements if we project the current position backward in time (Fig. 4). If our dynamic model adequately describes the motion and the estimated parameters are close, then the paths should agree well. If the match is poor, then there is either un-modeled disturbance (rapid change in acceleration or turn rate), or there is a tracking failure due to problems in feature extraction, etc.

We compare this matching error to the matching error of the hypothesis that we are not moving at all. A track is classified as moving if the matching error of the moving hypothesis is at least 4 times less than that of the fixed hypothesis. If the track is determined to be moving, then we apply a correction to the linear velocity which minimizes the history matching error.

We insure that the history contains sufficient information to determine motion by requiring both eigenvalues of the sum of the position information for all of the features in the history to exceed a threshold.

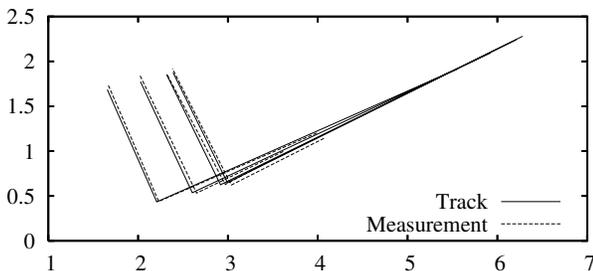


Fig. 4: Back-prediction of track history

Because the history-based validation is fairly computationally expensive (about 250 microseconds per track), we only do validation on tracks that have a significant estimated velocity.

X. EXPERIMENTAL RESULTS

One of the barriers to progress in tracking is the difficulty of quantitative performance evaluation. There are two main problems:

- It is very hard to get ground truth for complex real-world environments (especially moving objects), and
- Tracking algorithms and tuning parameters embody assumptions about the world; a tracker that works well for one use may work badly for another.

We have used three different evaluation approaches:

A. Velocity Estimation Error

To evaluate velocity estimation error without requiring instrumented moving targets, we have evaluated the tracker performance on real urban scanner data where only the scanner was in motion. Since the sensor actually measures position of objects relative to the scanner, the result is equivalent to the objects being in motion.

To characterize the performance of the Kalman filter only (without track validation), Fig. 5 shows the distribution of velocity error normal to the direction of scanner motion (the most critical component for collision prediction.) We can see that though there is a strong central tendency with a Gaussian character (σ 0.13 meters/second), there are also fat tails and extreme outliers. We developed the track validation test to reject these velocity outliers.

B. Track Validation Performance

To evaluate the performance of track validation and the moving/non-moving track classification, we tested it at several sensitivity levels. This both evaluates the sensitivity of the parameter settings and also provides a way to evaluate classification errors without manually examining every track. First we ran the classifier at a high sensitivity on 60 seconds of busy urban data (moving vehicles and pedestrians as well as lots of clutter.) This identified a large but still manageable number of possibly moving objects. We then manually classified the objects by visual examination of video records and the raw laser data. This same data was then run at successively lower sensitivity levels.

In Fig. 6 we see that only at the lowest sensitivity level do we misidentify as fixed tracks that were previously identified as moving. Setting 1 corresponds to complete disabling of track validation, with a simple velocity threshold as the moving test. The large decrease in falsely moving tracks from setting 1 compared to 2 shows that even at the highest sensitivity a large number of false velocities are being rejected.

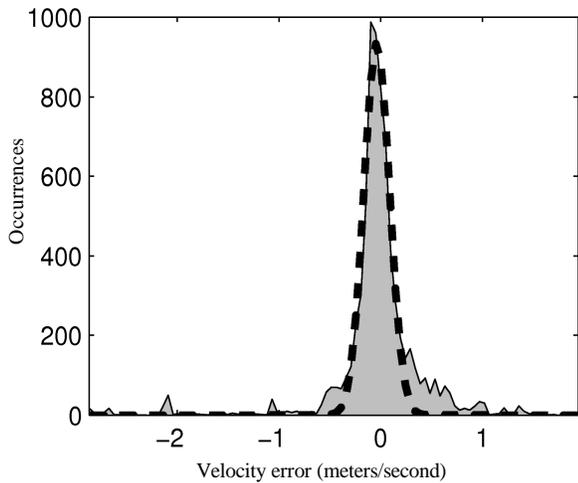


Fig. 5: Velocity error distribution

C. Collision Warning System Performance

Compelling indirect evidence for tracking performance comes from the performance of the collision warning system based on it. The warning system is very sensitive to errors in velocity estimation because these errors often cause a collision to be predicted. If large velocity errors are common, an unacceptable system false alarm rate results.

The warning system was tested on two transit buses operating in two different cities over 6989 km of normal passenger service, with 263 hours of system operation. The tracker identified 334,460 tracks, 4,527 of which were classified as moving.

To evaluate the accuracy of warnings, we selected a sample of 5 hours of data with warning rates similar to the whole dataset and classified the 302 warnings as correct or incorrect based on visual examination of video and scanner data.

Depending on likelihood of collision, the system gives two kinds of warnings: yellow and red. There were 30 false red warnings (less than 5 per hour.) The causes of the false red warnings were: tracker velocity error (70%), ground returns (15%), other (5%). Velocity error caused one false red warning for every 11,000 total tracks, and one per every 150 moving tracks.

Collision warning is a very demanding application because true dangerous situations are rare. There were 21 correct red warnings, corresponding to a false alarm rate of 60%. The warning system performance would benefit from further reductions in tracker velocity error. If the velocity-related false alarms could be reduced below one in 51000 tracks, then ground returns would become the dominant error source.

XI. CONCLUSION

Reliable tracking of moving objects from a moving vehicle is challenging, primarily because apparent shape change due to changing perspective can be interpreted as

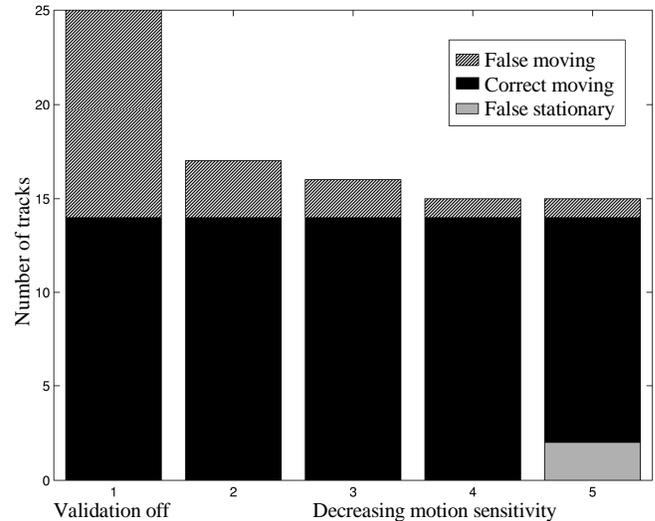


Fig. 6: Track validation performance

motion. By extracting stable linear features, characterizing feature position uncertainty and validating the consistency of recent observations with the estimated motion, we have developed a tracker that has acceptable performance for collision warning, and that is efficient enough to operate in real time on embedded processors.

REFERENCES

- [1] C. Mertz, D. Duggins, J. Gowdy, J. Kozar, R. MacLachlan, A.M. Steinfeld, A. Suppe, C. Thorpe, and C. Wang, "Collision Warning and Sensor Data Processing in Urban Areas", *5th international conference on ITS telecommunications*, 2005, pp. 73-78.
- [2] Integrated Collision Warning System, Final Technical Report (December 2004), FTA-PA-26-7006-04.1, joint publication with Carnegie Mellon University Robotics Institute and California PATH program
- [3] Integrated Collision Warning System, Final Evaluation Report (May 2006) FTA-PA-26-7006-06.1, joint publication with Carnegie Mellon University Robotics Institute and California PATH program.
- [4] K. Fuerstenberg, K. Dietmayer, V. Willhoeft, "Pedestrian recognition in urban traffic using a vehicle based multilayer laserscanner," *Intelligent Vehicle Symposium*, 2002, pp. 31-35.
- [5] C. Wang, C. Thorpe, S. Thrun, "Online Simultaneous Localization and mapping with detection and tracking of moving objects: theory and results from a ground vehicle in crowded urban areas", *International Conference on Robotics and Automation*, IEEE, 2003, pp. 842-849.
- [6] D. Steller, K. Furstenberg, K. Dietmayer, "Vehicle and object models for robust tracking in traffic scenes using laser range images", *Intelligent Transportation Systems*, 2002, pp. 118-123.
- [7] M. Lindstrom, J Eklundh, "Detecting and tracking moving objects from a mobile platform using a laser range scanner", *International Conference on Intelligent Robots and Systems*, 2001, IEEE, pp. 1364-1369.
- [8] A. Fod, A. Howard, M Mataric, "A laser-based people tracker", *International Conference on Robotics and Automation*, IEEE, 2002, pp. 3024-3029.
- [9] Y. Bar-Shalom, T. E. Fortmann, Tracking and Data Association, Mathematics in Science and Engineering, Academic Press, 1988.
- [10] D. Schulz, W. Burgard, D. Fox, A. B. Cremers, "Tracking multiple moving targets with a mobile robot using particle filters and statistical data association", *Robotics and Automation*, 2001, pp. 1665-1670, vol.2.