Leader Tracking for a Walking Logistics Robot

Michal Perdoch†∗, David M. Bradley†, Jonathan K. Chang†, Herman Herman†, Peter Rander†, Anthony Stentz†
†National Robotics Engineering Center
∗Computer Vision Laboratory
Carnegie Mellon University

Abstract—A key challenge of developing robots that work closely with people is creating a user interface that allows a user to communicate complex instructions to a robot quickly and easily. We consider a walking logistics support robot, which is designed to carry heavy loads to locations that are too difficult to reach with a wheeled or tracked vehicle. In this application the robot is carrying equipment and supplies for a group of pedestrians, and the primary task for the user interface is to keep the robot traveling with the overall group in the right formation. This paper presents a marker tracking system that uses near infrared cameras, retro-reflective markers, and LIDAR to allow a particular user to designate himself as the robot’s leader, and guide the robot along a desired path. We provide an extensive quantitative evaluation to show that the proposed system is able to detect and track a leader through unconstrained and cluttered off-road environments under a wide variety of illumination and motion conditions.

I. INTRODUCTION

Off-road environments can pose serious logistical challenges for transporting supplies and equipment. Typical wheeled and tracked platforms can access less than half the world’s landmass [1]. Here we consider the Defense Advanced Research Project Agency (DARPA)’s Legged Squad Support System (LS3), a high-mobility legged platform designed to accompany pedestrians and transport supplies in the other half. Guiding a robot through a complex off-road environment using joystick control is difficult and consumes the full attention of an operator. As a result, most logistics robots have developed “leader-follow” modes of operation where the robot attempts to follow in the path chosen by a designated leader, typically using LIDAR data, GPS or Ultra-Wide-Band (UWB) radar transponder tags on the leader [2], [3], [4]. However, each of these approaches has limitations: GPS often has poor accuracy in forests, LIDAR-only tracking is difficult in cluttered environments, and active transponders require batteries that must be kept charged.

This paper presents a system for tracking pedestrians designated with passive markings through cluttered off-road environments. The robot’s intended leader is marked with two horizontal retro-reflective stripes. A camera on the robot rapidly captures pairs of images with and without a Near Infra-Red (NIR) LED flash in under 1/2000th of a second. The image with only ambient illumination, $I_A$, is subtracted from the image with both ambient illumination and the LED flash, $I_F$ to create a difference image $I_D = I_F - I_A$ that cancels most illumination effects and highlights the retro-reflective strips as shown in Fig. 2. Potential leader markings are detected in the difference image and confirmed using the 3D point cloud returned by the LIDAR for that area. Confirmed detections are then grouped into tracks and a motion trajectory for each track over time is estimated with an extended Kalman filter.

II. RELATED WORK

LIDAR and multimodal systems for obstacle and pedestrians detection and tracking in urban environments were proposed recently in automotive applications [5], [6]. They use a rotating multiple beam LIDAR sensor that produces over 1M points per second. In [5], the point cloud data are classified directly to produce a 2.5D occupancy grid. The LIDAR based object detection system proposed in [6] generates spin images and virtual orthographic camera images from the LIDAR data, computes histograms of gradients of those virtual images, and classifies both individual observations and a global descriptor of the track with boosted classifiers. In contrast, due to mechanical reliability and packaging constraints our off-road system uses a single actuated LIDAR plane sensor that provides much fewer (40×) points per second. This combined with heavily cluttered off-road scenes makes detecting of leaders from LIDAR alone very challenging. For this reason, our work is similar to the early fusion approach used in [7], where LIDAR detections are used to limit the search space in the image domain, and image detections based on implicit shape models [8] are fused with multiple motion models represented by Kalman Filters. We reverse this fusion ordering and use our image based detector to hypothesize leader locations, and the point cloud for precise distance estimation and verification.

Much work has focused on detecting and tracking unmarked pedestrians in monocular images, and an excellent overview of the state of the art is available in [9]. While unmarked pedestrian detection in monocular images has
shown significant progress in recent years, a leader-follow user interface requires high-reliability with an extremely low false positive rate in complex off-road environments where occlusions are common.

Recently significant progress has also been made in image domain tracking [10], [11], and a thorough overview of the state of the art is provided in [12]. Early experiments with the CSK tracker [11] for this application were quite promising – however, an initial detection is still required to initialize tracking. The image-based marker detector proposed in Section III proved to be efficient enough to run on every frame.

To make the leader tracking problem more tractable we place unobtrusive passive markers on the leader. Passive markings have been used in other work to improve pedestrian safety in industrial environments. In [13] construction workers wearing fluorescent safety vests are detected in images by using color histograms. Retro-reflective strips on the vests were detected in [14], [15] by adding active illumination to the camera system and capturing pairs of near infra-red (NIR) images with and without a NIR flash. Range to the pedestrians was estimated using a stereo pair of IR cameras, or from a learned classifier on monocular images. Their work was concurrent with and independent from ours and they show promising performance on a small test set (7 minutes total) despite a significant 20ms gap between flash and non-flash images. By contrast this work uses a dramatically shorter (100×) time difference between flash and non-flash images. This enables lower false positive rates and longer range detection (30m) even on a walking platform where the sensors are constantly subjected to significant angular motion. Additionally we present results on a large 67 hour dataset consisting mainly of natural off-road terrain, providing a useful complement to the industrial sites discussed in previous work.

III. IMAGE BASED MARKER DETECTOR

An image-based marker detector is used to provide real-time detection of the retro-reflective markings on the leader with high angular accuracy. A camera on the robot rapidly captures a pair of images with \((I_F)\), and without \((I_A)\) a Near Infra-Red (NIR) LED flash in under 0.5 ms. The “difference” image, \(I_D = I_F - I_A\), removes most effects of ambient illumination. The retro-reflective strips of the leader marker are particularly bright in \(I_D\) due to their strong specular reflections (see Fig. 2). An efficient sliding window classifier is used to find hypotheses of the angular position and approximate range of the retro-reflective leader marker in the difference image. This section describes the image processing pipeline (after an image denoising procedure specific to this type of camera) and continues to a description of the cascade classifier.

A. Bayesian Prefilter

Most windows in the difference image can be eliminated using computationally inexpensive features – the window...
mean and variance – computed efficiently over a set of scales \( S \) using integral images. As a fast prefiltering stage we train a simple classifier to predict whether a marker will be detected by the full marker detector for a window of size \( h \in S \) using only the mean \( \mu_h \) and variance \( \sigma^2_h \) of intensity \( I_D \) in that window. This classifier is strictly less powerful than the full marker detector, and by training it only to predict the output of the full marker detector we can utilize an extensive unlabeled dataset. The unlabeled dataset was used to collect 3D histograms of marker detections, \( H^+(h, \mu_h, \sigma_h) \) and non-marker classifications \( H^-(h, \mu_h, \sigma_h) \) for the set of parameters \( h \in S, \mu_h \in M = \{0 \ldots 255\} \), and \( \sigma_h \in \Sigma = \{0 \ldots 128\} \).

The pseudocount parameters \( \alpha, \beta, \alpha_m, \beta_m \) add a weak Dirichlet prior that ensures the marker detector is evaluated in uncertain or unobserved parts of the parameter space.

\[
P(M|h, \mu) = \frac{\alpha_m + \sum_{i \in \Sigma} H^+(h, \mu, \sigma_i)}{\beta_m + \sum_{i \in \Sigma} H^+(h, \mu, \sigma_i) + H^-(h, \mu, \sigma_i)}
\]

\[
P(M|h, \mu, \sigma) = \frac{\alpha + H^+(h, \mu, \sigma)}{\beta + H^+(h, \mu, \sigma) + H^-(h, \mu, \sigma)}
\]

Marginal \( P(M|h, \mu_h) > \tau_1 \) and joint \( P(M|h, \mu_h, \sigma_h) > \tau_2 \) Bayesian classifiers were then defined. The two classifiers were chained into a cascade that stops evaluation once a window is rejected as negative. The classifier thresholds \( \tau_1, \tau_2 \) were set to an average false negative rate of 1%. Between 95% and 97% of the true negative windows were rejected by this stage (depending on the environment).

### B. Marker Detector

The leader marking pattern of two horizontal strips was chosen for long-range observability (which necessitates large retro-reflective and non-retro-reflective regions) while still retaining a distinctive appearance and allowing monocural range estimation. This marking pattern is also well-suited to Haar-like features which can be computed efficiently from integral images. For every candidate window location \( (r, c) \), scale \( h \), and orientation \( \theta \in \{-\pi, \pi\} \), we construct a feature vector from five rectangular regions (see Fig. 3). This feature vector includes: 1) the means of the five rectangular regions.

2) As marker brightness can change quickly with range, and range is linearly related to height in the image, size-normalized means \( \mu_{en} \) are also computed to provide some invariance to range. 3) The minimum of the means of regions 2 and 4 which should correspond to the marker strips. 4) The maximum of regions 1, 3, and 5 which should not be retro-reflective. 5) Absolute differences between all four pairs of regions which should have the same intensity. 6) Window parameters: \( h \), the average intensity \( \mu \) and variance \( \sigma^2 \) of the window.

All windows \( (r, c) \) in each image of the training dataset in a close vicinity \( (r, c) \in \{(r, c) : |(r, c) - (\hat{r}, \hat{c})| \leq t/h\} \) of the ground truth marker location \( (\hat{r}, \hat{c}) \) were labeled as positive examples and random non-overlapping windows were labeled as negative examples. Three decision tree classifiers were trained, one for each orientation, on an extensive manually labelled dataset. The maximum orientation response is then thresholded for the final classification decision.

1) **Clustering**: The sliding window marker classifier usually responds around the correct location of the marker both in \( (r, c) \) coordinates and neighboring window heights \( h \). To find cluster centers in the \( (r, c, h) \) space we use a fast mean shift algorithm, alternative quick shift [16], which can be efficiently evaluated on a sparse point cloud. We estimate density using a Gaussian window function and employ spatial hashing to reduce computation.

Each cluster center is used as a marker position hypothesis. After clustering the hypotheses, the scale space position of each cluster center is computed as the weighted mean of hypothesis \( (r, c, h) \) coordinates weighted by detection confidence. Similarly, the weighted average orientation of the hypotheses in a cluster is used as an initial estimate of the orientation. Finally, the sum of the confidence of the hypotheses is used as an overall score for the cluster in further processing.

2) **Clustered Marker Classifier**: After clustering the number of remaining marker hypotheses is much smaller than the original number of windows, allowing the use of a more computationally intensive classifier, which takes advantage of the estimate of marker location, size and orientation provided by the clustering stage to fit a model of the marker into the image evidence around the proposed location.

First, an image patch is extracted for each hypothesis and normalized using the estimated marker orientation and scale. Starting from the estimated center of the marker, each image column in the normalized patch is searched to find intersections with the retro-reflective strips of the marker. Dynamic programming is employed to segment successive columns and extract the most probable outline of the strips.
Fig. 4. LIDAR returns from inside the detection frustum projected into the camera image. Top row from the left to right: leader at about 7m, leader at 24m, leader accompanied by another person, and two cluttered forest scenes. 2D detections are denoted by white squares, \( g \) is denoted by a blue vector, and \( x \) is displayed as a green vector. The intensity of the points is proportional to the difference from the estimated 2D depth. Points inside the “marker” region are shown in green, points inside the “person” region are shown in yellow, and points in the “context” region are shown in red. Bottom row: corresponding depth histograms for the frustum. Note that the time incidence threshold \( t_e \) was increased to 0.5 sec for visualization, resulting in a 2.5 fold increase in the number of LIDAR returns.

Once the strips have been precisely located, a decision tree and random forest classifiers [17] were trained to predict the probability that the image patch contains the leader marker given the following features: 1) Average intensities and standard deviations of the retro-reflective strips and the dark space between them, before and after image denoising, as well as the absolute differences of those quantities between strips. 2) Average widths and positions of the strips in the normalized patch coordinate system. 3) Each of the statistics is further accumulated over three regions top, center and bottom part of the strip. 4) Coefficients of a second order polynomial to capture the curvature of the strips. 5) Aspect ratio of each strip. 6) Number of clustered hypotheses, estimated size of the marker computed from the strip model and average size of the clustered hypotheses.

After detection, the orientation of the marker is refined by finding a dominant orientation over a small spatial neighborhood. Finally, the refined and horizontally oriented marker patch is split into lower and upper halves, and distance between the centroids of the white pixels in each half are used to estimate the range to the marker.

IV. LIDAR BASED MARKER DETECTOR

The image based marker detector discussed previously can be confused by patterns that are similar to the real leader marking, but at the wrong scale. For instance, as can be seen in Fig. 2, grass is very reflective in NIR and two parallel blades of grass close to the camera can approximate the appearance of the leader marker at long range. Additionally, as cameras are angular sensors, the image based marker detector is very accurate in determining the direction to the leader, but less accurate in determining the distance to the leader (marker size is a noisy indicator when the leader orientation is unknown). The actuated line-scan LIDAR on the robot provides a complimentary source of information with constant 3 cm range accuracy, but reduced directional and temporal accuracy. Hence, a separate detector based on the 3D point cloud of LIDAR returns is used to reject false positives and refine the accuracy of the 3D position of the leader.

The LIDAR based marker detector starts from 2D hypotheses provided by the image-based detector. For each 2D hypothesis, the LIDAR point cloud around the ray from the camera center to the detected marker is used to compute a feature vector for each quantized range along the ray. A simple classifier is then used to detect if the 3D point distribution around each quantized range is consistent with a pedestrian.

A. 3D Features

As 3D points are received from the LIDAR they are projected into world-frame coordinates using a pose estimate of the robot movement computed by a separate visual odometry system [18]. Each image marker hypothesis provides a ray \( d = l - c \) pointing from the camera optical center in world coordinates at the time the image was captured, \( c \in \mathbb{R}^3 \), to the approximate location of the leader estimated using the detected marker location in the image \( l \in \mathbb{R}^3 \). Pedestrians naturally orient themselves with respect to gravity, so for each camera image we construct a non-orthogonal coordinate system using the gravity vector \( g \) and the ray from the camera optical center to the image detection \( d \) as \( y = g / \| g \|_2, z = d / \| d \|_2 \), and \( x = y \times z \).

Variations in the posture of the leader can cause foreshortening of the target. This leads to incorrect monocular range estimates with the image detections. Consequently, in this image-specific coordinate system, 3D features are computed as histograms of weighted sums of points in a set of 3D boxes covering every 0.25m slice orthogonal to the \( z \) (range) axis from \( \| d \|_2 / 2 \) to \( 2 \| d \|_2 \) as shown in Fig. 4. The set of boxes for each range slice consists of a small box around the expected position of the marker, a larger box that
should encompasses the pedestrian acting as the leader, and a context box that should not contain any points from the leader.

The weighted sum for each box uses a weight \( w = w_tw_r \) for each 3D point \( p \) consisting of a temporal weight \( w_t \) and a range-normalization weight \( w_r \). The temporal weight \( w_t \) is computed from the difference between the image timestamp \( t_i \) and the 3D point timestamp \( t_p \) as \( w_t = e^{-|t_p-t_i|} \) if \( |t_p-t_i| < t_e \) and \( w_t = 0 \) otherwise. The range-normalization weight \( w_r = 1/n_r \) is designed to compensate for the 0.5° angular spacing between successive measurements in each scanline from the LIDAR sensor, where \( n_r \) is the expected number of returns from a fully-occupied box at range \( r \).

**B. Mode Classifier**

Modes of the 3D point distribution with respect to range are extracted by first convolving the box weighted sums along the range axis with a windowing function to increase robustness to quantization induced noise, and then extracting the modes. A subset \( S_m \) of all local modes \( m_i \) with values \( v_{m_i} \), is computed for the person’s histogram:

\[
S_m = \{ m_i \mid v_{m_i} > 0.8 \max v_{m_i} \}
\]

The final classification score for each mode is computed as \( f(m_i) = w^T x_i + b \), where \( w \) is a hand-tuned vector of weights, \( x_i \) a vector concatenating the box sums near the range of mode \( m_i \), and \( b \) is a learned threshold.

**V. EKF Tracker**

A tracking module fuses the marker detections into 4D \((x,y,z,t)\) world-frame tracks and outputs the most likely leader position, if any, around the robot. Each marker hypothesis from the image and LIDAR based marker detectors is either associated to an existing leader track, used to start a new leader track, or discarded. For each track, an Extended Kalman Filter (EKF) is used to estimate the world-frame 3D position and horizontal velocity at each detection. For more details of the Kalman Filter parameterization please refer to the technical report version of this paper.

To filter out erroneous observations, we maintain multiple potential leader tracks and use a simple classifier to predict which track, if any, is most likely to be a leader. For each new marker observation \( z_k \), an affinity function \( A(z_k, T) \) is computed for every existing track \( T \in \mathcal{T} \) in the current set of tracks \( \mathcal{T} \).

To define the affinity function \( A(z_k, T) \) between observation \( z_k \) and track \( T \), we denote the hidden dynamic state \( \hat{x}_{m|n,T} \) estimated by the EKF for track \( T \) at time \( n \), given the observations associated to track \( T \) up to and including time \( n \). We will also denote as \( l(x, \mu, \Sigma) \) the log likelihood of \( x \) given the measurement \( \mu \) with covariance \( \Sigma \):

\[
l(x, \mu, \Sigma) = -\frac{1}{2} (\log |\Sigma| + (x-\mu)^T \Sigma^{-1} (x-\mu) + C)
\]

The affinity function \( A(z_k, T) \) of observation \( z_k \) to track \( T \) of observations up to and including \( k-1 \) is defined as:

\[
A(z_k, T) = l(\hat{x}_{k|k-1,T}, z_k, R_k) + l(\hat{x}_{k|T}, \hat{x}_{k|k-1,T}, P_{k|k-1,T})
\]

where:

\[
\hat{x}_{k|k-1,T} = H_k \hat{x}_{k|k-1,T} + v_k
\]

i.e. a sum of log likelihood of the predicted observation \( \hat{x}_{k|k-1} \) given the observation \( z_k \) and log likelihood of posterior state \( \hat{x}_{k|T} \) given the predicted state \( \hat{x}_{k|k-1,T} \).

Track \( T^* \) with maximal affinity to the new observation is then defined as:

\[
T^* = \arg\max_{T \in \mathcal{T}} A(z_k, T).
\]

If \( A(z_k, T^*) \) is greater than an association threshold \( \tau_a \), the observation is associated with \( T^* \). If \( z_k \) is sufficiently far from any existing tracks (i.e. \( A(z_k, T^*) < \tau_a \) where \( \tau_n < \tau_a \)), it is assigned to a new track \( T_n \). Otherwise, \( z_k \) is assigned to the erroneous observations set \( \emptyset \):

\[
z_k \rightarrow \begin{cases} T^* & \tau_a < A^*(z_k, T^*) \\ \emptyset & \tau_n \leq A^*(z_k, T^*) \leq \tau_a \\ T_n & A^*(z_k, T^*) < \tau_n, \ T_n \rightarrow \mathcal{T} \end{cases}
\]

The affinity function (3) is based only on position with respect to current tracks and does not reflect the confidence of the leader observation \( z_k \). Confidence is incorporated through two thresholds \( \tau_{low}, \tau_{high} \) and \( \tau_{low}, \tau_{high} \) for image and LIDAR detections respectively. Detections below the \( \tau_{low} \) thresholds are discarded before computation of the association score. Detections above \( \tau_{high} \) are considered for starting a new track, when the affinity score is lower than \( \tau_n \).

**A. Construction of Leader Path**

Potential marker tracks are scored using a simple function \( L(T) \) of the most recent observations in the track, which uses the number and score of image and LIDAR observations and their affinity scores \( A^*(z_k, T) \) over the last few associations. The most probable leader track \( T_{leader} \) is chosen as:

\[
T_{leader} = \arg\max_{T \in \mathcal{T}} L(T).
\]

During the observation association step, the affinity score of the most likely leader track, \( A(\hat{x}_{k|T_{leader}}) \), is increased by a factor \( \alpha_{leader} \) to encourage associations to the leader track over any nearby competing tracks. Gaussian smoothing is used to create a final leader marker path from the observations in the tracks selected as \( T_{leader} \). This merged and smoothed path is then output for use in robot navigation. The final leader marker path is also used to actively pan the leader tracking camera to the predicted position of the leader marker, allowing longer range operation with a steerable cameras with narrow horizontal field of view.
VI. COMPONENT EVALUATION AND PARAMETER SETTING

The leader tracking system was extensively evaluated during weeks of field testing in a variety of test sites with greatly varying terrain, seasonal conditions, vegetation cover, and lighting conditions. From this testing, over 67 hours of data was manually labeled for parameter tuning, as well as for qualitative and quantitative evaluation of the leader tracking system.

A. Datasets

Sensor data was collected and manually labeled from a variety of terrains including urban, desert, forest, meadows, and rain forest areas. Additionally the datasets captured seasonal variations with data collections spread out over a period of two years. The LOE dataset was collected with the final version of the system during the evaluation of the system and was not used for any system training or tuning. All sensor data was accurately calibrated, time-stamped and logged for all robot operation, creating an extensive dataset for offline algorithm development. The logs contain raw sensor inputs including color, NIR, monochrome stereo, and differencing camera images, as well as LIDAR scans, encoder, IMU and GPS readings. The position of the leader was manually labeled in this data to provide datasets for training and quantitative evaluation.

Data-labeling for a high-reliability system providing a novel capability is ill-posed: until a high-reliability system exists it can be very expensive to obtain the massive amount of labeled data required to quantify high-reliability. For this application no alternative existing system was available to provide labels for the correct marker position. Consequently, a dataset for training the image based marker detector was constructed by an iterative process of projecting tracked marker trajectories from the world frame into the difference images using the known sensor to world transforms, and selecting the most confident nearby image based marker detection. Negative examples were extracted from confident marker detections far from a marker trajectory. The labeled difference images were then manually inspected for accuracy and corrected as needed. This process was repeated several times as the marker detector and overall tracking accuracy improved. Ultimately an extensive dataset with more than 600k images was obtained from a selected subset of logs. This dataset was used for the evaluation and testing of the image based marker detector classifiers.

The leader trajectories generated by the leader tracking system were manually inspected and labeled to generate a set of confirmed leader tracks, false positive tracks, true negative time periods (when the leader was not visible) and false negative time periods when the leader was visible but not detected. These were then used for the overall system quantitative evaluation, as well as component-level evaluations for the LIDAR based marker detector, and the EKF tracker.

B. Image Based Marker Detector

The image based marker detector is composed of several components that can be tested and tuned individually.

1) Marker Detector Bayesian Prefilter: The image based marker detector Bayesian prefilter stage described in Section III-A provides a computational speedup by predicting if the full detector would fire on a given image window, given computationally cheap window mean \( \mu_h \) and window standard deviation \( \sigma_h \) features.

Fig. 5 shows the speedup provided by the joint probability model \( P(M|h, \mu_h, \sigma_h) \) on the logged data for a variety of threshold values \( \tau_1 \) and \( \tau_2 \), on the marginal (Eqn. 1) and joint probability (Eqn. 2) distributions respectively. Threshold \( \tau_2 \) was set to reject about 1% of positive windows and provided an average speedup of 25\( \times \) on the logged data.

2) Overall Performance of Image Based Detector: To assess the overall performance of the image based detector and LIDAR based detector we used annotated true leader tracks and labeled the output of both stages. The leader path location was back-projected into the difference images and a bounding box of the target was computed using its known physical size. Hypotheses generated by the image based marker detector were labeled as true or false using a bounding box overlap test. Finally, all marker hypotheses in the dataset were ordered by the post-filter classifier confidence and a precision recall curve was plotted, see Figure 6. This performance analysis was used to set the image detection thresholds of the tracking stage at \( \tau_{low} = 0.7 \), and \( \tau_{high} = 0.9 \).

C. LIDAR Based Marker Detector

A similar evaluation process was applied to the output of the LIDAR based detector. 3D location hypotheses incorporating an image detection labeled “true” and closer than 0.5m to the true leader range were used as true positives and the rest as incorrect hypotheses. The score \( f(m_k) \) (see IV-B) of the LIDAR verification stage multiplied by confidence of the image based hypothesis was used to compute ordering for a precision recall curve. Results for all target distances
and results separated into 10m range bins are shown in Fig. 7. Marker hypotheses produced by the LIDAR based marker detector have a large impact on the precision of the final trajectory as they provide the most reliable estimates of the distance to the leader. Consequently, the thresholds $\tau_{low} = 0.15$, and $\tau_{high} = 0.47$ used in the tracking module were set for a low false positive rate when starting new tracks and a high recall rate for existing tracks.

VII. QUANTITATIVE EVALUATION OF THE LEADER TRACKING SYSTEM PERFORMANCE

The final system level evaluation was performed using the parameter settings from the previous sections. All stages of the leader tracking system were rerun on all available datasets. The output data were split into segments with a reported leader track, and segments without a leader track. All segments were manually inspected and labeled to create a dataset for assessing the overall system performance. Segments were excluded from evaluation if they met any of the following criteria: duration shorter than 2 seconds, sensor acquisition or calibration errors, the marker was visible but not on a pedestrian, the marker was within 2m from the robot, the marker was beyond 40m from the robot, the marker was rotated by more than 80 degrees, or multiple markers were visible. Results of the quantitative evaluation are summarized in Table I. Please note, that the performance of the system improved over time, which in turn improved the quality of the data collected by the actively-controlled panner module holding the differencing cameras. Hence the performance of the system when measured by reprocessing old logs is expected to be strictly worse that the live performance of the final system. Also later datasets were generally collected from more challenging routes, as the data quality improved over time, which in turn improved the quality of the data collected by the actively-controlled panner module holding the differencing cameras. Consequently the leader tracking system was tuned to minimize false positives. As shown in Table I, in final testing it produced only 16 seconds of false positives in almost 9 hours of leader tracking. False negatives were more common caused by sparse vegetation, even in cases where the image based detector is able to tolerate partial occlusions (ordered by occurrence): - The robot to move towards non-leader objects or people – are potentially much more hazardous than false negatives – where the robot simply stops until the leader is reacquired. Consequently the leader tracking system was tuned to minimize false positives. As shown in Table I, in final testing it produced only 16 seconds of false positives in almost 9 hours of leader tracking. False negatives were more common and accounted for the other 3 minutes and 6 seconds of incorrect tracking in the final experiment. False negatives were generally caused by a few common failure modes (ordered by occurrence):

- **Partial occlusions can prevent LIDAR detections** - The image based detector is able to tolerate partial occlusions caused by sparse vegetation, even in cases where the vegetation prevents any LIDAR returns from the leader. However, if the leader track is lost completely, reacquiring the precise position of the leader requires detections from

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Total tracking</th>
<th>Correct Tracking</th>
<th>Incorrect Tracking</th>
</tr>
</thead>
<tbody>
<tr>
<td>LTA2</td>
<td>17h 12m 25s</td>
<td>41.237</td>
<td>57.639</td>
</tr>
<tr>
<td>LTA3</td>
<td>23h 9m 34s</td>
<td>57.639</td>
<td>78.281</td>
</tr>
<tr>
<td>LTA4</td>
<td>10h 29m 59s</td>
<td>41.624</td>
<td>57.639</td>
</tr>
<tr>
<td>LTA5</td>
<td>7h 39m 17s</td>
<td>52.673</td>
<td>78.281</td>
</tr>
<tr>
<td>Total</td>
<td>58h 31m 16s</td>
<td>98.747%</td>
<td>1.252%</td>
</tr>
<tr>
<td>LOE</td>
<td>8h 53m 07s</td>
<td>52.828</td>
<td>46.540</td>
</tr>
</tbody>
</table>

TABLE I

SUMMARY OF THE QUANTITATIVE EVALUATION.
the LIDAR-based marker detector, and consequently it may not be possible to re-establish tracking until a clear line of sight is available.

- **Frequent occlusions** - To improve resolution for long-range tracking the differencing cameras have narrow field of view lenses, and their field of view is steered in the horizontal direction by a panning mechanism. If a leader track is available, the panning mechanism compensates for latency in the leader tracking and panner control systems by forward-predicting the leader position. Otherwise the panner searches the whole horizontal FOV. In challenging environments with frequent occlusions however, if the leader is lost there may be very few chances for reacquisition. This can lead to an extended false negative period.

- **Difficulty reacquiring leader at long range** - The long range tracking was found to be reliable to about 30 m. There are three main factors contributing to failure at longer distances: i) the density of the LIDAR point cloud decreases with range, reducing the frequency of LIDAR returns from the marker. ii) The irradiance of the NIR flash reflected from the leader marker decreases with range, lowering the signal to noise ratio of the retro-reflective markings in the difference image. iii) The differencing camera does not correct for robot motion or panner module actuation before computing the difference image $I_D$. Hence these motions can cause thin bright regions in $I_D$ along strong gradients in the ambient illumination image.

- **Limited vertical FOV** - The panner module contains two differencing cameras with slightly overlapping fields of view to cover a combined vertical field of view of -10 degrees to +40 degrees from the horizontal axis of the robot. In rugged terrain the leader marker can drop out of the bottom of the field of view as the robot is cresting a hill, causing a loss of tracking.

VIII. CONCLUSIONS

We have presented a multistage system for guiding a high-mobility walking platform through off-road environments by tracking a human leader. Specifically the system presented here reconstructs highly reliable human leader trajectories in world coordinates, for use by the robot’s planning system. It is one of the key components of the robot’s user interface, and it allows a designated leader to guide the robot through challenging off-road environments while imposing a minimal cognitive load. An extensive quantitative evaluation showed that the final system performed correctly for 98.8% of the total 67 hour manually-labeled dataset collected from five diverse environments, and 99.4% of the final test dataset.

ACKNOWLEDGMENT

The authors would like to thank the many people at Boston Dynamics, JPL, NREC, DARPA and MCWL who contributed to the LS3 program and made this work possible. In particular Clark Haynes, Jean-Sebastian Valois, Chad Rockey, Matthew Powers, Joseph Russino, Benjamin Wasserman and James Bagnell made early contributions to the leader-tracking software. Michael Licitra, Jaime Bourne, Justin Scheiffle, Jeffrey McMahan and Vladimir Altman were instrumental in creating the integrated sensor head used in this work. Max Bajracharya and Jeremy Ma from JPL created the pose estimation system. Many people from Boston Dynamics were involved in creating the uniquely-capable walking platform used in this work, as well as providing logistical support during testing, and fruitful technical discussions. In particular, the authors would like to thank Matthew Malchano, Adam Komoroski, Kevin Blakenspoo, Alfred Rizzi, Joseph Bondaryk and Marc Raibert for their efforts to make this work possible.

This material is based upon work supported by DARPA under Contract No. 1011050. The views, opinions, and/or findings contained in this article/presentation are those of the author(s)/presenter(s) and should not be interpreted as representing the official views or policies of the Department of Defense or the U.S. Government.

REFERENCES